Journal of Privacy and Confidentiality
Vol. 8 (1) 2018

Issue in honor of S. Fienberg

Submitted April 2018
Published December 2018

A PRIVACY PRESERVING ALGORITHM TO RELEASE SPARSE HIGH-DIMENSIONAL HISTOGRAMS

BAI LI, VISHESH KARWA, ALEKSANDRA SLAVKOVIC, AND REBECCA C. STEORTS

PhD Student, Department of Statistical Science, Duke University, Durham, NC
e-mail address: bai.li@duke.edu

Assistant Professor, Department of Statistical Science, Temple University, Philadelphia, PA
e-mail address: vishesh@temple.edu

Professor, Department of Statistics; Associate Dean for Graduate Education, Eberly College of Science, The Pennsylvania State University, University Park, PA
e-mail address: sesa@psu.edu

Assistant Professor, Department of Statistical Science, affiliated faculty in Computer Science, Biostatistics and Bioinformatics, the information initiative at Duke (iiD) and the Social Science Research Institute (SSRI), Duke University, Durham, NC
e-mail address: beka@stat.duke.edu

Abstract. Differential privacy has emerged as a popular model to provably limit privacy risks associated with a given data release. However releasing high dimensional synthetic data under differential privacy remains a challenging problem. In this paper, we study the problem of releasing synthetic data in the form of a high dimensional histogram under the constraint of differential privacy. We develop an (ϵ, δ)-differentially private categorical data synthesizer called *Stability Based Hashed Gibbs Sampler* (SBHG). SBHG works by combining a stability based sparse histogram estimation algorithm with Gibbs sampling and feature selection to approximate the empirical joint distribution of a discrete dataset. SBHG offers a competitive alternative to state-of-the art synthetic data generators while preserving the sparsity structure of the original dataset, which leads to improved statistical utility as illustrated on simulated data. Finally, to study the utility of the resulting synthetic data sets generated by SBHG, we also perform logistic regression using the synthetic datasets and compare the classification accuracy with those from using the original dataset.

1. Introduction

Large amounts of data help advance scientific inquiry, but they also come with growing privacy concerns, e.g., Lane et al. [2014]. Many recent works have demonstrated that traditional methods for data protection such as anonymization, coarsening, and releasing aggregate data can be broken [Dwork et al., 2017]. To resolve this issue, differential privacy (DP) has emerged as a framework that informs the design of privacy mechanisms with a

Key words and phrases: differential privacy, high dimensional histogram, synthetic data, feature selection, feature hashing, Stability Based Algorithm (SBA), Stability Based Hashed Gibbs Sampler (SBHG).

JPrivacy Confidentiality
www.journalprivacyconfidentiality.org
DOI:10.29012/jpc.657

mathematically specified disclosure risk [Dwork et al., 2006]. However, it is often criticized for failing to maintain sufficient data utility, i.e., the accuracy loss could be large making data unusable for valid statistical inference. Moreover, DP is primarily designed to work in an interactive setting where users ask queries and receive noisy answers. This limits the applicability of DP only to those queries that can be specified beforehand and for which DP implementations exist.

An alternative approach is releasing synthetic data (via multiple imputations) initially proposed by Rubin [1993]. Synthetic datasets allow a user to move beyond the interactive interface and perform a larger class of statistical analyses, potentially arbitrary, enabling enhanced data sharing and scientific reproducibility. Synthetic data methods have shown promise of preserving statistical utility and have seen an explosion in methodological developments and applications over the past decade, especially with data from official statistics surveys, including producing both fully and partially synthetic data [Abowd and Woodcock, 2001, Drechsler and Reiter, 2010, Drechsler and Vilhuber, 2013, Kinney et al., 2011, Raghunathan et al., 2003, Reiter, 2005a]. However, the formal privacy protections offered by synthetic datasets are not well understood. The key caveat is the following: The utility of synthetic data is primarily determined by imputation models [Reiter, 2005b] and models that are too accurate often leak sensitive information [Abowd and Vilhuber, 2008]. Moreover, it is difficult to quantify the risk of multiple synthetic data releases, whereas in a DP framework, the risk composes.

In 2015, Steve and I traveled together to Ithaca from Pittsburgh for a workshop on differential privacy (DP). This is one of my fondest memories with Steve for many reasons. The trip involved many of my favorite things, which Steve and I both shared — statistics, hockey, great food, and good conversation. For me, this is one of my most memorable trips with Steve, as I learned so much from our conversations together.
The trip started with Steve picking me up at my apartment in Shadyside. We started our long journey in the afternoon. As the car pulled out of the driveway, you could hear the announcers yelling in the background, hockey sticks banging back and forth, and if you knew Steve well, then you knew of course the Penns were playing. So, we pulled off and mostly in silence for the first part of our journey, both hoping that the Penns would pull out a victory. After a bit, we stopped for a quick bite to eat and then resumed our drive.
For the second part of the trip, we agreed that I would drive this stretch. It was during this part of the drive that Steve and I had this very long discussion about record linkage and differential privacy. We had a long discussion regarding the history of DP, the challenges of working in a DP framework, and open problems in the field. This was quite useful for me as it enabled me to understand the entire landscape of DP without having to worry about the minute details. It's been even more helpful as I have worked on trying to make DP methods work in practice for sparse contingency tables with former MS student Bai Li and my collaborators Vishesh Karwa and Sesa Slavković.
After our research conversation, we listened to the news and Steve read his newspaper. It reminded me of earlier days when I would take road trips with my father, who would

(cont.)

Our approach combines the *statistical* methods of synthetic data generation and DP, similar to [Charest, 2011, Karwa et al., 2017, Park and Ghosh, 2013], such that DP controls the risk formally, even under multiple data releases, and the synthetic data allow for more usability, thus combining the best of both worlds. Practical applications of DP synthetic data have been demonstrated using the U.S. Census OnTheMap data that consist of approximately one million records with two variables [Machanavajjhala et al., 2008]. However, generation of usable DP high-dimensional and sparse synthetic categorical data still remains a challenge. Sparse tables are those for which the model complexity is of the same order or even larger than the sample size [Fienberg et al., 2010]. We consider a histogram (or contingency table) to be sparse if the number of non-zero entries is small compared to the number of zero entries, and there is a large number of non-zero entries with small counts. The first issue is the choice of model used to generate the synthetic histograms, and modeling of sparse tables even without privacy constraints is non-trivial (e.g., [Dunson and Xing, 2009]). Second, it is well known that support estimation under ϵ-DP is difficult [Wasserman, 2012]. In fact, in ϵ-DP it is necessary to add noise to non-zero entries of the table, making accurate release of high dimensional data impossible.

also sit in the passenger side of the car and read the newspaper. Steve would look over the newspaper from time to time and his eyes would wander over the speedometer. He would then say "watch the speed," and then he would go back to reading. Later on, we listened to some music from a band that one of his neighbors was in. They were quite good and I remember thinking how fun the music was to drive to. We eventually made it from Pittsburgh to Ithaca and this is one of my most memorable times with Steve.
All in all, it was wonderful to see this other side of Steve. As I came to learn, he was one of the kindest people I knew, he was extremely trustworthy, and he was a true friend in addition to being one of the best researchers our field has ever seen. I feel so fortunate each day that I was able to be his collaborator, his mentee, and his friend. He's missed each and every day, but his memory will always go on.

Beka Steorts
DOI: 10.29012/jpc.709

To address the above issues, we propose an (ϵ, δ)-DP algorithm, the *Stability Based Hashed Gibbs Sampler*, for releasing high-dimensional sparse histograms, by combining the (ϵ, δ)-DP Stability Based Algorithm (SBA) [Bun et al., 2016, Vadhan, 2016] with feature selection and Gibbs sampling. We address the first issue by approximating the empirical distribution, which is a good model with high statistical utility, by a collection of conditional distributions. These are released under DP, and a Gibbs sampler is used to generate synthetic datasets from the noisy conditionals. The second issue is addressed because the SBA allows for release of high-dimensional histograms without destroying their support. Incorporating feature selection reduces the number and dimensionality of the resulting conditional histograms. As a result, we ensure that the sparsity pattern of the joint distribution of the high-dimensional histogram is partially preserved. These two techniques ensure impossible combinations in the data remain impossible after privacy, and it also ensures that the histograms have enough mass to obtain non-trivial utility. Thus, our proposed method is not just a combination of existing methods, but a carefully thought out solution (and the first of its kind) for the issue of high-dimensional histogram estimation. Finally, we use a Gibbs sampler to sample from the noisy hashed conditional distributions to generate synthetic contingency tables. The proposed framework is provably (ϵ, δ)-DP. In addition, for both simulated and real data, we illustrate that both privacy and utility can be achieved. Finally, we also perform logistic regression on the resulting output from our algorithm, comparing the classification accuracy as a downstream task to illustrate the utility of the resulting synthetic data sets.

The rest of the paper proceeds as follows. Section 2 reviews differential privacy, the Laplace mechanism, and the Stability Based Algorithm (SBA). Because it is well known that statistical utility highly depends on the values of ϵ and δ, we propose measuring the statistical loss in utility when SBA is applied as a function of both parameters (see Proposition 1). Via our proposition, one can see that many choices of ϵ and δ are unsuitable for high-dimensional histograms, which leads us to consider DP Gibbs sampling. Section 3 reviews the success and limitations of a recently proposed DP Gibbs sampler. We then propose our Stability Based Hashing Gibbs (SBHG) sampler that combines the stability based algorithm (SBA) with feature selection to develop a Gibbs sampler based synthesizer. In this framework, we use feature selection methods to increase the counts in an histogram while also reducing the dimensionality of the histograms. Sections 4.1 and 4.2 evaluate the performance of our proposed algorithm using simulated data sets and real data sets, respectively. Section 5 provides a discussion and directions toward future work.

2. Differentially Private Algorithms for Releasing Histograms

In this section, we present notation, review two DP algorithms for releasing histograms, and propose a measure of accuracy loss for the stability based algorithm (SBA).

Steve has shaped the lives and careers of many people within and outside the statistics community, over the course of his long career. I am very fortunate to be one of those people. Steve was unique in his ability to forge connections between seemingly unrelated fields. He was the "match maker" — connecting people and research areas as varied as privacy and statistical inference, algebra and statistics, copulas and log-linear models, sampling and design of experiments, networks and contingency tables and so on.

My approach towards research, in general, and in the area of privacy and statistical inference in particular, has been greatly influenced by Steve. I want to give a few examples to honor his memory. I first met Steve when I was a graduate student at Penn State University working with my PhD advisor Aleksandra Slavkovic. I gave a talk at a grant meeting explaining an idea to incorporate the additional randomness introduced due to privacy in the likelihood function for statistical modeling. I was just a graduate student, so my thoughts were not very clear. Steve immediately knew what I was trying to say, because he had been advocating a similar approach! The key goal of statistical inference is to make statements about population parameters, hence one needs to design privacy procedures with an eye towards this goal. This philosophy, which surrounds my work on privacy, is straight out of Steve's book! Another direction that I work on, that Steve often advocated, was the focus on finite sample inference as opposed to asymptotic inference. This is evident from his pioneering contributions to the work on sparse contingency tables where asymptotic tests don't always make sense. One of his pet peeves was the problem of analyzing sparse contingency tables under privacy, for which one necessarily has to take a finite sample viewpoint. Another point that always stays with me was Steve's

(cont.)

2.1. **Differential Privacy.** Let $D = (\mathbf{d}_1, \ldots, \mathbf{d}_n) \in \mathcal{D}^n$ be an input database containing n observations (records), where $\mathbf{d}_i \in \mathcal{D}$. The goal is to produce a synthetic dataset, say $Z \in \mathcal{Z}$, which satisfies DP. Let $\epsilon, \delta > 0$. As in Wasserman and Zhou [2010] and Hall et al. [2011], define $Q(\cdot \mid D)$ to be a randomized mechanism that takes D as an input data set and generates a synthetic data set Z. Let $D \sim D'$ if $D' \in \mathcal{D}^n$ and D and D' differ by one record. Q_n satisfies (ϵ, δ)-DP Dwork et al. [2006] if for all measurable $B \subset \mathcal{Z}$ and all $D \sim D' \in \mathcal{D}^n$,

$$Q(Z \in B \mid D) \leq e^{\epsilon} Q(Z \in B \mid D') + \delta.$$

For small ϵ, which is the privacy budget, the value of one individual's record has a small effect on the output. When $\delta = 0$, Wasserman and Zhou [2010] use a hypothesis testing framework to show that ϵ-DP provides protection against an adversary who knows all but one record, when the records are independent; this is the strongest form of DP. The parameter δ measures the failure probability and is generally set to be a small value, typically negligible in n.

suggestion to explore the possibility of a "Bayesian" version of privacy. While there are many such notions out there, I don't think they would have answered Steve's question. Quoting from my not-so-good memory, Steve said something to the extent that "Differential privacy requires one to reason about all the datasets that one would have collected, and not the one we actually have at hand - this is similar in spirit to what is done in a frequentist setting, where one reasons about other datasets one could have seen, as opposed to a Bayesian setting where one conditions only on the observed dataset."

There are many other instances where Steve influenced my thinking, including other problems that I work on in network modeling, causal inference and algebraic statistics, all of which were Steve's favorites. In fact, I still channel Steve in my work and my talks - What would have Steve said, or done for this problem? One thing that used to happen a lot with me when talking to Steve was the following - he would say something to me about a research problem during our meetings, and I would come out of the meeting assuming that I understood what Steve was saying. But six months later, out of nowhere, I would suddenly realize what Steve was actually trying to say! This happens even now, and in fact I frequently go back to re-read his papers and my email conversations with him to get advice and insights. I truly miss him!

Vishesh Karwa
DOI: 10.29012/jpc.704

2.2. **Differentially Private Histograms.** Assume dataset $D = (\mathbf{d}_1, \ldots, \mathbf{d}_n)$ consists of n independent and identically distributed (i.i.d.) random vectors of p categorical features. Each record $\mathbf{d}_i$ is an independent random sample of the vector $X = (X_1, \ldots, X_p)$ with the jth feature X_j taking values in the set I_j. For example, if all features are binary, then $I_j = \{0, 1\}$ for all j. Let $\mathcal{I} = \prod_j I_j$ denote the Cartesian product of $I_1, \ldots, I_p$. Let $m = |\mathcal{I}|$ denote the cardinality of $\mathcal{I}$ or the number of cells in the contingency table. Note if all features are binary, then $m = 2^p$. In general, as p increases, we quickly observe $m \gg n$. The histogram representation of D is obtained by counting the number of occurrences of each element in $\mathcal{I}$. For each element $i \in \mathcal{I}$, let c_i denote the number of times i appears in D. The set of counts $c = \{c_1, \ldots, c_m\}$ then denotes the histogram representation of D, where $\sum_{i \in \mathcal{I}} c_i = n$. Let $\mathbb{S} = \{i \in \mathcal{I} : c_i > 0\}$ denote the support of the histogram and let $s = |\mathbb{S}|$ denote the number of non-zero bins in the histograms, which is a number of non-zero counts. For high-dimensional and sparse histograms, s is much smaller than n and m.

The Laplace mechanism releases a histogram under ϵ-DP by adding Laplace noise to each cell of the histogram with the scale parameter $b = \frac{2}{\epsilon}$ (see Algorithm 3, Appendix B) [Dwork et al., 2006]. When the histogram is dense (large s relative to m), the Laplace mechanism may work well. When dealing with high-dimensional histograms such as binary contingency tables such that $m = 2^p >> n > s$, then there are many cells with small counts, and in particular, zero counts. The Laplace mechanism adds noise to every cell and in such a sparse setting this leads to a great loss in statistical utility [Fienberg et al., 2010]. In addition, the Laplace mechanism is computationally inefficient, with computational complexity $O(m)$, and the noise added to every cell is linear in the dimension m of the histogram. However, Balcer and Vadhan [2017] recently proposed a computationally fast approach for the Laplace mechanism for sparse settings.

2.3. **Stability Based Algorithm.** The idea of thresholding and adding noise to bins with non-zero counts was initially introduced by Korolova et al. [2009] and Götz et al. [2009] for the release of search logs. Vadhan [2016] and Bun et al. [2016] provide a simplified algorithm and coined the term stability based algorithm (SBA), tailored for sparse high-dimensional histograms and satisfying (ϵ, δ)-DP. But the straightforward application of the SBA may lead to significant statistical utility loss (i.e., accuracy loss) as explained below. In the SBA (Algorithm 1), Laplace noise is added to every non-zero cell of the histogram. Finally, noisy cells that are smaller than a fixed threshold $t = 1 + \frac{2}{\epsilon}\log\frac{1}{\delta}$ are set to zero.

Algorithm 1 Stability Based Algorithm (SBA)

```
Input: Non-zero counts in a histogram $\{c_i, i \in \mathcal{S}\}$, $\epsilon$, $\delta$
Output: An $(\epsilon, \delta)$ DP histogram
Let $\mathcal{S} = \{i : c_i > 0\}$ and $|\mathcal{S}| = s$.
For each $i$ in $\mathcal{S}$, let $z_i = c_i + e_i$ where $e_i$ is Laplace noise with mean 0 and scale parameter $\frac{2}{\epsilon}$.
Set all noisy counts below the threshold $t = 1 + \frac{2}{\epsilon}\log\frac{2}{\delta}$ to 0.
Output the noisy histogram
```

Statistical utility is highly dependent on the values of ϵ and δ, and even more so in the setting of high-dimensional sparse histograms. Due to this, we propose measuring the statistical loss in utility when SBA is applied (see Proposition 2.1), in terms of the expected L_1 error as a function of ϵ and δ.

Proposition 2.1. *For a fixed histogram of counts* $\{c_1, \dots, c_s\}$, *the expected* L_1 *error of Algorithm 1 is* $n - \sum_{i=1}^{s} p_i c_i + \dfrac{2}{\epsilon}\left(\sum_{i=1}^{s} p_i\right)$ *where* $p_i = \mathbb{P}\left(z_i > t\right) = \frac{1}{2}\exp\left(-\frac{2(t-c_i)}{\epsilon}\right)$.

Proof. Recall $z_i = c_i + e_i$. Let $I_i = \mathbb{I}(z_i > t)$, where $\mathbb{I}(.)$ is the indicator function and $I_i \sim Ber(p_i)$. The L_1 error can be written as

$$\begin{aligned} L_1 = \sum_{i=1}^{s} |z_i I_i - c_i| &= \sum_{i\in\mathcal{I}} |e_i| + \sum_{i\in\mathcal{I}^c} c_i \\ &= \sum_{i\in\mathcal{I}} |e_i| + \sum_{i=1}^{s} c_i(1 - I_i) \qquad (2.1) \\ &= \sum_{i\in\mathcal{I}} |e_i| + \sum_{i=1}^{s} c_i - \sum_{i=1}^{s} c_i I_i = \sum_{i\in\mathcal{I}} |e_i| + n - \sum_{i=1}^{s} c_i I_i, \end{aligned}$$

where $\mathcal{I} = \{i : z_i > t\}$. Now note that $|e_i|$ is a exponential random variable and conditional on $|\mathcal{I}|$, $\sum_{i=1}^{s} |e_i|$ is a Gamma random variable with scale $\frac{2}{\epsilon}$ and shape $K = |\mathcal{I}|$. Thus, we have

$$\mathbb{E}\left(\sum_{i\in\mathcal{I}} |e_i| \,\middle|\, K\right) = \frac{2K}{\epsilon}.$$

Also, $\mathbb{E}\left(K\right) = \sum_{i=1}^{s} \mathbb{E}\left(I_i\right) = \sum_i p_i$. Similarly, it is easy to see that $\mathbb{E}\left(c_i I_i\right) = p_i c_i$, which gives the result. □

When data sets have many attributes, there will be fewer records that can be grouped together and this typically leads to having the non-zero counts in their corresponding histograms be very small. Consider when a cell count is k. Then the probability that these counts are not thresholded to zero is

$$\mathbb{P}\left(k + e_i > 1 + \frac{2}{\epsilon}\log\frac{1}{\delta}\right) = \frac{\delta}{2}\exp\left(-\frac{\epsilon(k-1)}{2}\right).$$

For example, if $k = 1$, the probability that the count is not thresholded to zero is $\delta/2$. To guarantee privacy, we often choose small δ in practice, which means most of the one-counts are thresholded to zero, causing a loss of statistical utility that can potentially lead to the wrong statistical inference. More generally, when $k = O(2/\epsilon)$, the probability that the count is not thresholded is $O(\delta)$. In an extreme example, suppose that in a histogram, every non-zero count is exactly one. Then the SBA needs to threshold $1 - \frac{\delta}{2}$ of non-zero counts to zero to ensure (ϵ, δ)-DP, which leads to an almost empty histogram. Such resulting data are potentially unusable for any statistical analysis.

3. Differentially Private Gibbs Sampling and feature selection

To address the aforementioned drawbacks of SBA for releasing of high-dimensional, sparse histograms, we propose the *Stability Based Hashed Gibbs Sampler* (SBHG). This sampler combines the SBA with Gibbs sampling and feature selection, reduces the dimensionality of the histograms and most importantly improves statistical utility. We first provide background on our inspiration for this sampler and the components needed for its composition.

3.1. Gibbs sampling.

Gibbs sampling is a powerful way to sample from the joint distribution of a dataset. A generic Gibbs sampler works by iteratively sampling from the full conditional distributions $\mathbb{P}(X_i|X_{-i})$, where X_{-i} denotes the vector of all features except X_i; see Appendix C for a review of the Gibbs sampler. The Gibbs sampler requires the computation of the set of conditional distributions for each feature i. When all the features are categorical, the conditional distributions can be represented by a collection of histograms. Thus, to generate synthetic data using Gibbs sampling, it is sufficient to release the collection of histograms that represent these conditional distributions. Based on this idea, Park and Ghosh [2013] proposed a Perturbed Gibbs Sampler (PeGS) that satisfies DP and allows for the release of synthetic histograms of categorical data. PeGS, however, has several limitations, leading to poor performance specifically on the release of high-dimensional, sparse histograms.

3.2. Overview of the Perturbed Gibbs Sampler.

The PeGS is a three-step categorical data synthesizer that draws samples from perturbed conditional distributions to construct a synthesized data set, which satisfies ϵ-DP. First, it generates empirical full conditional distributions $\mathbb{P}(X_j|X_{-j})$ from the original histogram. Second, it perturbs the empirical full conditional distributions to obtain perturbed conditional distributions $\mathbb{P}_\alpha(X_j|X_{-j})$, where α is a privacy controlled parameter that satisfies DP (or l-diversity). More specifically, suppose the empirical conditional distribution satisfies

$$\mathbb{P}(X_j = i|X_{-j}) = \frac{n_{ij}}{N_j},$$

where n_{ij} is the count of the ith category, N_j is the total number of records that have the same X_{-j}. Then the perturbed conditional distribution is

$$\mathbb{P}_\alpha\left(X_j = i | X_{-j}\right) = (n_{ij} + \alpha)/(N_j + C_j\alpha),$$

where C_j is the number of categories. This perturbation is equivalent to adding α "virtual samples" to each category as $\hat{n}_{ij} = n_{ij} + \alpha$. Finally, a synthetic data set is generated by iteratively sampling from these perturbed conditional distributions similar to the process of a Gibbs sampler.

Feature selection. The number of possible combinations for X_{-j} can be large, i.e., there can be too many conditional histograms leading to computational inefficiency. Feature selection reduces the number of conditional distributions, improving computational efficiency. Park and Ghosh [2013] proposed replacing $\mathbb{P}\left(X_j | X_{-j}\right)$ for $j = 1, \ldots, p$ with hashed conditional distributions $\mathbb{P}\left(X_j | h(X_{-j})\right)$ for some hash function h, where the choice of h depends on the data used.

A feature hash h ranks all the elements of a feature vector

$$X_{-j} = (X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_p)$$

based on their mutual information with X_j, retaining the top R features. More precisely, $h(X_{-j}) = (X_{h_1}, X_{h_2}, \ldots, X_{h_R})$, denotes the highest mutual information of the jth feature X_j, where R is a tuning parameter. By applying the feature hash h, instead of using $\mathbb{P}\left(X_j | X_{-j}\right)$, we compute $\mathbb{P}\left(X_j | h(X_{-j})\right) = \mathbb{P}\left(X_j | X_{h_1}, \ldots, X_{h_R}\right)$. As a result, the conditional distributions with the same hash key are combined into one conditional distribution, leading to a compressed feature space.

Feature selection reduces the sparsity issue (zero or low counts in many cells relative to the total number of cells and the sample size n) as it increases the number of samples in each conditional histogram. Suppose for the ith feature, there are k different combinations $X_{-j}^1, \ldots, X_{-j}^k$ that have the same hash key $h(X_{-j}^1) = h(X_{-j}^2) = \cdots = h(X_{-j}^k) = h_0$, and suppose their histograms are $H(X_j | X_{-j}^i) = \{c_{1i}, \ldots, c_{pi}\}$, for $i = 1, \ldots, k$. The hashed conditional distribution corresponding to h_0 is given by

$$\mathbb{P}\left(X_j | h(X_{-j}) = h_0\right) = \frac{\{\sum_{i=1}^k c_{1i}, \ldots, \sum_{i=1}^k c_{pi}\}}{\sum_{j=1}^p \sum_{\ell=1}^k c_{j\ell}}.$$

Feature selection pulls all the conditional distributions with the same hash key into one and re-weights them to form a new histogram. Thus, the hashed conditional distributions contain more non-zero entries, and each cell contains more samples.

Limitations. In practice, the PeGS often performs poorly due to several limitations. Within each step of Gibbs sampler, the synthetic sample costs an ϵ-privacy budget, which means after N iterations, the samples only satisfy $(N\epsilon)$-DP, which is not practically useful. Park and Ghosh [2013] suggested the following modification: within each step of the Gibbs sampler, draw $B > 1$ times iteratively and only release the last sample. At the beginning of each iteration, reset the visited conditional distributions $\mathbb{P}_\epsilon\left(x_i | h(x_{-i})\right)$ to uniform distributions. Unfortunately, in order to satisfy ϵ-DP, this extra step requires the lower bound for α being $\alpha \geq 1/(\exp\{\epsilon B/p\} - 1)$, where p is the number of features. When ϵ and p are fixed, at least one of α and B must be large to satisfy this inequality. If either α or B is too large, the utility of the synthetic sample is reduced. More precisely, a large α implies that many "virtual samples" are added to the conditional histograms, while a large B implies that many conditional histograms are reset to uniform distributions.

Furthermore, for sparse data sets, the support of the noisy conditional histograms may not be close to the non-noisy ones, i.e., the conditional histograms may contain many sampling zeros even after applying feature selection. Since the perturbation in PeGS adds α "virtual samples" to each cell category, it will add many "virtual samples" to cell counts that do not exist in the original data set. In this case, there are no corresponding conditional histograms for these samples, and the Gibbs sampler becomes stuck, running into convergence issues. It should be noted that because of this issue in particular, we are not able to compare the performance of our proposed algorithm directly with PeGS on synthesis of high-dimensional sparse histograms.

3.3. **The Stability Based Hashed Gibbs Sampler.** Due to the aforementioned drawbacks of PeGS with high-dimensional sparse histograms, we propose the the Stability Based Hashed Gibbs Sampler (SBHG), which is described in Algorithm 2. The SBHG first generates the empirical hashed conditional distributions from the original histogram. We next apply the SBA to each hashed conditional histogram (instead of adding "virtual samples" as done in PeGS.) SBA is more practical because it requires no extra perturbations and preserves more utility for the hashed conditional histograms (see Section 2.3). Finally, we synthesize a new histogram by running a Gibbs sampler on the noisy hashed conditional histograms. We avoid the problem of adding noise at each step of the Gibbs sampler by releasing the conditional histograms just one time, instead of releasing a sample at each iteration. In addition, the use of SBA allows us to ensure that the support of the noisy conditional histograms remains close to the non-noisy conditional histograms, limiting the sampler from generating synthetic samples that typically do not occur in the original dataset. Our algorithm guarantees (ϵ, δ)-DP as we show in Theorem 3.1. By applying hashing, we assume that the compressed conditional distribution is a good approximation to the original full conditional distribution.

Before introducing our algorithm below or proving Theorem 3.1, we first provide some notation. Consider a feature $X_j \in I_j$, where without loss of generality $I_j = \{1, \dots, K_j\}$. Let the hash function $h(X_{-j}) \in \{h_1, \dots, h_M\}$ have M levels. This implies that feature X_j has M conditional hashed histograms. Then for the hashed value h_m, the histogram is given by the vector of counts

$$C_{j,m} = (c_{k_j|h_m})_{k_j=1}^{K_j}$$

where

$$c_{k_j|h_m} = \#I(X_j = k_j | h(X_{-j}) = h_m).$$

Theorem 3.1. *Algorithm 2 guarantees (ϵ, δ)-DP if the hash function h is chosen independent of the data.*

Proof. It suffices to show that Algorithm 2 is (ϵ, δ)-DP through step 6 of Algorithm 2 since the rest of the steps are post processing. To be more precise, consider a feature $X_j \in I_j$, where without loss of generality $I_j = \{1, \dots, K_j\}$. Consider a hash function $h(X_{-j}) \in \{h_1, \dots, h_M\}$, which implies that feature X_j has M conditional hashed histograms. As already mentioned, for each hashed value h_m, the histogram is denoted by a vector of counts $C_{j,m}$. Now consider the query that collects the vector of counts $C_j = (C_{j,m})_{m=1}^{m}$. This query has global sensitivity (see Dwork et al. [2006]) of 2. To see this, observe that each record $\mathbf{d}_k \in D$ appears in one and only one hashed histogram $C_{j,m}$. Thus, replacing a record can change at most two counts in collection of M counts. Therefore, we can release the query C_j using SBA with privacy budget $(\epsilon/p, \delta/p)$. By composition, the overall privacy is (ϵ, δ). First, we note that in order for improvements from advanced composition to take effect, one needs $p \approx 10$, as can be seen in our experimental analysis. Second, we note that due to stability, empty conditional histograms remain empty after privacy. □

Remark 3.2. Theorem 3.1 ensures that the SBHG is differentially private only if the hash function does not depend on the data. However, in some of the experiments that we conduct, the hash function of feature j that we use is data dependent, as we need to compute the top R features that have highest mutual information with X_j. One can make the SBHG fully DP by using the Exponential Mechanism in [Jung et al., 2014] at the feature selection step of the algorithm. By the adaptive composition property of differential privacy, if the hash function is chosen in a differentially private manner, and then used as an input to the SBHG, the overall algorithm remains differentially private (with additive loss in privacy). Note that selecting the correct features to define h is akin to choosing a model that best describes the data, e.g., [Lei et al., 2016]. In our experiments, SBHG+EM referrs to the Gibbs sampler where the hash function is chosen using the exponential mechanism and SBHG refers to the case where we use a data-dependent hash function without privacy. Even though the latter case is not fully differentially private, we consider this case **to understand the trade-off between privacy and utility as a function of the synthetic data generation method and not due to error in selecting an inferior set of features**, i.e., given the best set of features, how well can we generate synthetic data. We further use SBHG with Exponential Mechanism (SBHG+EM) to achieve full privacy, then compare it to SBA and also a recent method (PrivBayes), showing that our method is, in fact, preferred or has competitive performance in high-dimensional, sparse settings.

Remark 3.3. While SBA greatly reduces the problem that the support of the noisy conditional histograms may not be close to the non-noisy ones, it does not eliminate the issue completely. In fact, there can be samples produced by the Gibbs sampler which hash to features that do not have a conditional histogram. We address this issue by allowing the sampler to reject samples. When the sampler encounters a sample x^* whose conditional distribution is not available, i.e., we do not have the corresponding conditional distribution for $\mathbb{P}_\epsilon\left(x_i^* | h(x_{-i}^*)\right)$, the sampler rejects this sample by simply keeping the value from the last iteration. This is essentially a Metropolis-Hasting procedure as we regard x^* as a proposed sample that has zero likelihood in terms of the empirical distribution.

4. Experiments

We apply our algorithm to both simulated and real data sets and use L_1 error given by the distance between the true counts and the noisy counts to measure the loss of utility. Our choice to consider the L_1 distance is motivated by Proposition 2.1, where we derive the L_1 error of SBA. In addition, we also use the L_1 distance as half of L_1 distance between two contingency tables is proportional to the total variation distance, and hence is a natural measure to compare two joint distributions. Our goal is to measure the utility in terms of the entire joint distribution. Using a performance of a downstream task such as model building is also a useful alternative, but such a metric depends very specifically on the type of task and model that one aims to build with the synthetic data. On the other hand, the

Algorithm 2 Stability Based Hashed Gibbs Sampler

```
Input: Data $D$ with $n$ records and $p$ features and $p$ a hash function $h$. DP parameters $\epsilon$ and $\delta$.
Output: An $(\epsilon, \delta)$ differentially private synthetic data set
for $j \leftarrow 1$ TO $p$ do
    Let $X_j \in \{1, \dots K_j\}$ and $h(X_{-j}) \in \{h_1, \dots, h_M\}$.
    Let $C_{j,m} = (c_{k_j|h_m})_{k_j=1}^{K_j}$ where $c_{k_j|h_m} = \#I(X_j = k_j|h(X_{-j}) = h_m)$.
    Apply SBA with $(\epsilon/p, \delta/p)$ to counts $C_j = (C_{j,m})_{j=1}^{M}$ to get noisy counts $C_{j,m}^{\epsilon}$.
    Construct the conditional hashed distributions $\mathbb{P}_\epsilon\left(X_j|h(X_{-j}) = h_m\right) \propto C_{j,m}^{\epsilon}$.
end for
Initialize an empty data set $D_{\epsilon,\delta}$ with $n$ rows and $p$ columns.
for $i \leftarrow 1$ TO $n$ do
    At row $i$ of $D_{\epsilon,\delta}$, let $x_i^0 = \{(x_{1i}^0, \dots, x_{pi}^0)\}$ with $\mathbb{P}_\epsilon\left(x_{ij}^0|h(x_{i(-j)})\right) > 0$ for $j = 1, \dots, p$.

    for $t \leftarrow 1$ TO $S$ do
        for $j \leftarrow 1$ TO $p$ do
            Sample a point $x_{\text{prop}}$ from $\mathbb{P}_\epsilon\left(X_j|h(X_{-j}) = h(x_{i(-j)}^{t-1})\right)$.
            Let $h_{prop} = h\left(\left(x_{i1}^{t}, \dots, x_{i(j-1)}^{t}, x_{ij}^{t}, x_{i(j+1)}^{t-1}, \dots, x_{ip}^{t-1}\right)\right)$
            if $\mathbb{P}_\epsilon\left(X_j|h(X_{-j}) = h_{prop}\right)$ is empty then
                Accept the proposed value. Set $x_{ij}^{t} = x_{\text{prop}}$.
            else
                Reject the proposed value. Set $x_{ij}^{t} = x_{ij}^{t-1}$.
            end if
        end for
    end for
end for
Return $D_{\epsilon,\delta}$.
```

L_1 distance is independent of such a task, since it measures the distance between the full joint distributions. Thus, this particular distance is preferred over others when we consider the downstream task of logistic regression (see Section 4.2).

In our experimental analyses, we provide a comparison of our proposed methodology to the performance of our method to PrivBayes [Zhang et al., 2014]. Before presenting our experimental studies, we first briefly review the PrivBayes methods, providing crucial differences to our proposed approach. The PrivBayes algorithm synthesizes a data set by first constructing a Bayes network, then perturbing the network, and finally releasing a new data set sampled from the perturbed Bayes network. Specifically, in PrivBayes, first, a set of noisy marginal distributions are released. Next, these noisy marginals are used to approximate the joint distribution of the data as specified by a directed acyclic graph (DAG). In contrast, our approach works by writing the joint distribution of the data as the product of full conditional distributions. Note that there is no approximation in this step, unlike the first step of PrivBayes, where the joint distribution is approximated by the product of marginals. Thus, under our proposed method, we work directly with the joint distribution. In the next step, we approximate the full conditionals with the hashed conditionals, and release the hashed conditionals under differential privacy. The PrivBayes method also contains a feature hashing step, however, the major difference is being able to work with the joint distribution. In fact, working with the conditional distributions instead of marginal distributions allows us to approximate the joint distribution with a larger class of models than DAGs alone [Fienberg and Slavkovic, 2005, Gelman and Speed, 1999, Slavkovic, 2010]. Hence, we utilize a broader and more flexible class of models, while PrivBayes itself works with a small class of graphical models, restricting itself to DAGs that have a corresponding undirected graph for which the starting marginals are sufficient statistics. Another major difference between the two proposed methods is the fact that PrivBayes uses the Laplace mechanism to release conditional histograms. When the dimension of histograms is high, it will add too much noise that leads to great loss of utility as we pointed out in section 2.2. We illustrate this issue by running both methods on the sparse synthetic data set.

4.1. **Simulation Studies.** In this section, we consider two simulated data sets, one dense and one sparse. The dense dataset contains $n = 3,000$ records with $p = 3$ features; each feature has 10 categories. There are 1,000 possible distinct records in this data set, which occur at least once. The sparse dataset contains $n = 3,000$ records with $p = 10$ features; each feature has 10 categories. There are 10^{10} possible distinct records in this data set, noting that this is much larger than the number of records. Appendix A provides histograms of the dense data set. Note that the sparse data sets (synthetic and real) are too sparse to visualize. We use this data set to show how (i) the sparsity of data set, (ii) the privacy parameters, and (iii) different feature hashing settings affect our algorithm; see Appendix D for the settings of our simulation studies. While the L_1 error of our method is relatively large in our experiments, we show that it is not possible to obtain a small L_1 distance with a completely non-private method (due to the fixed choice of the feature hashing).

For the first set of experiments, we compare a naive application of SBA directly on the joint distribution to the SBHG+EM for the dense and the sparse datasets. We fix $\delta = 10^{-4}$ and vary ϵ from 0.1 to 10. We also compare the performance of our method to PrivBayes [Zhang et al., 2014]. Figures 1 (a) ($\epsilon = 0.1$ to 1) and (b) ($\epsilon = 1$ to 10) illustrate the theoretical L_1 errors from Proposition 2.1, and the empirical L_1 errors of SBA and SBHG+EM for the dense dataset. The plots suggest that theoretical and empirical L_1 error of SBA match, validating Proposition 2.1. The plots also show that SBHG+EM preserves greater utility than SBA on sparse histograms and that SBHG+EM offers stable utility over varying values of ϵ. Figures 1 (c) and (d) illustrate that when the data set is sparse, SBA offers no utility (L_1 error is equal to n), even for very large values of ϵ. On the other hand, SBHG+EM still preserves a significant amount of utility. Finally, when SBHG+EM preserves more utility compared to PrivBayes.

In the second set of experiments, we compare the performance of SBHG to two baseline (non-private) methods—an **empirical sampler**, drawing from the empirical distribution function and a **Gibbs sampler** (GS), drawing from the full conditional distributions. We use SBHG instead of SBHG+EM as we are interested in the trade-off between privacy and utility as a function of the data generation method. We begin by evaluating the performance of SBHG on the sparse dataset. In Figures 3 (a) and (b), we fix $\epsilon = 0.5$, and $\delta = 10^{-4}$ and vary the number of features R that are used in hashing. Figure 3 (a) shows the average L_1 error and Figure 3 (b) shows the running time, as a function of R. As R increases, the running time increases and the dimensionality reduction offered by hashing degrades. For example, $R = 9$ is equivalent to using a full conditional distribution because the data set

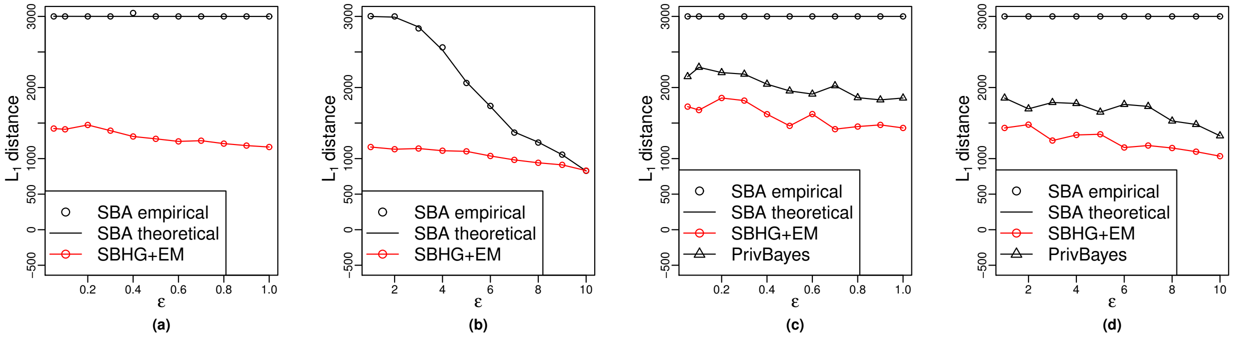

Figure 1: We apply SBA and SBHG+EM and PrivBayes on the dense (a and b) and sparse (c and d) simulated sets. The empirical L_1 errors between the original and synthetic histograms are plotted versus ϵ, where ϵ varies from 0.1 to 1 (a and c) and 1 to 10 (b and d) while $\delta = 10^{-4}$ fixed. In each plot, we also show the theoretical L_1 errors calculated from Proposition 2.1.

has 10 features. Figure 3 (a) suggests that the utility is maximized when we use only three features and the performance is closest to the non-private baseline samplers.

In Figures 3 (c) and (d), we study the trade-off between privacy and utility by evaluating how the choices of DP parameters ϵ and δ affect the L_1 error of our method on the sparse dataset. In Figure 3 (c) we vary ϵ from 0.1 to 1 while keeping δ fixed at 5×10^{-4}. In Figure 3 (d) we vary δ from 0.1 to 10^{-4}, while keeping ϵ fixed at 0.5. For each setting, we run the SBHG algorithm 40 times on the sparse data. R is optimally fixed at 3 for the sparse data set and 2 for the dense data set. We compare these results with empirical and Gibbs sampling. Figures 3 (c)–(d) show that data utility rapidly increases as we increase both parameters, and it becomes stable when they are relatively large.[1] Given the sparseness of the tested data set, these are promising results that we further validate with real data in the next section.

Finally, we compare the SBHG with an optimally chosen R to the baseline Gibbs and empirical samplers. In this simulation, we fix $\epsilon = 0.5, \delta = 10^{-4}$. We report the numerical values of the average L_1 error, (along with the standard error) for the dense and sparse datasets.

Tables 1 and 2 show that applying SBHG to a sparse data set leads to a lower utility and more unstable performance in comparison to a dense data set, as expected. The corresponding histograms for sparse data sets have small counts that are more likely to be thresholded to zero. This may introduce too many sampling zeros, and in general, SBHG always rejects the sample when it encounters sampling zeros, leading to a slowly mixing sampler. On the other hand, using an optimal R in SBHG leads to improved performance for both the dense and sparse data sets. This matches our analysis in section 3.2, where we point out that feature hashing essentially pools different conditional histograms with

[1] By the algorithm becoming stable, we mean that there is a low standard deviation of the L_1 error across multiple runs.

	L_1	SD
Empirical Sampling	978.25	81.82
Gibbs Sampling	1302.44	87.23
SBHG	1461.92	104.62
SBHG+EM (R=2)	1323.81	93.67

Table 1: The average L_1 error and its standard deviation for different data synthesizing methods applied to a dense data set. For each setting, we run 20 simulation tests.

	L_1	SD
Empirical Sampling	1101.24	101.71
Gibbs Sampling	1337.21	146.83
SBHG	1531.72	171.27
SBHG+EM (R=3)	1429.84	147.90

Table 2: The average L_1 error and its standard deviation for different data synthesizing methods applied to a sparse data set. For each setting, we run 20 simulation tests.

the same hash key together, which moderates the sparsity problem. Therefore, carefully choosing an optimal feature hashing parameter can improve the data utility.

Finally, we conclude our simulation studies with investigating how to decouple the effect of privacy on accuracy. To be more specific, implicit in the problem formulation is the constraint that the number of samples in the synthetic dataset (S) needs to be the same as the number of samples in the real dataset (n). This constraint can in fact be relaxed in many settings and we look at this small example to make it easier to interpret our paper's empirical results. First, we note that as $S \to \infty$, the error of empirical sampling will go to 0. Under SBHG, as $S \to \infty$, we find that the utility (L_1 distance) of the sampler is roughly the same as in our initial simulation studies (see Figure 2).

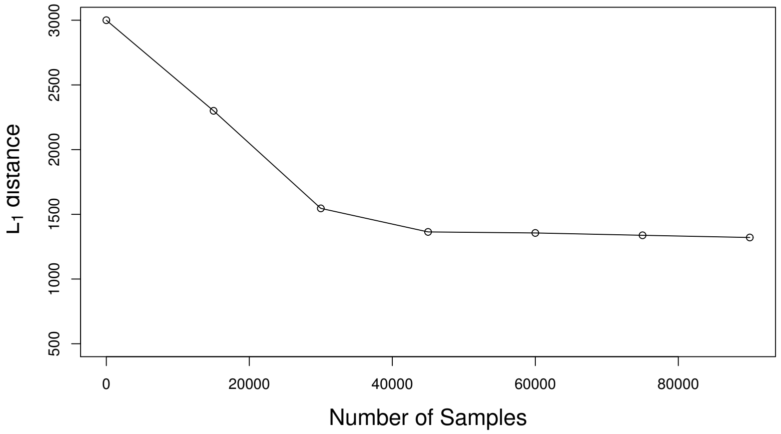

Figure 2: Under SBHG, as $m \to \infty$, we find that the utility (L_1 distance) of the sampler is roughly the same as in our initial simulation studies

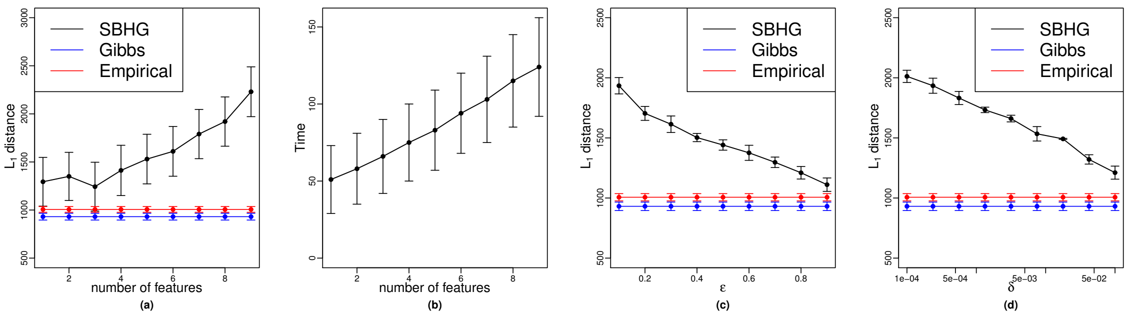

Figure 3: We apply SBHG to the simulated sparse data set with different settings. Plot (a) and (b) show the loss of utility and running time for different hashing settings. We fix $\epsilon = 0.5$ and $\delta = 5 \times 10^{-4}$ for each setting. The x-axes represent number of features kept in the feature hashing method. Plot (c) shows the loss of utility for ϵ varying from 0.1 to 1, while fixing $\delta = 5 \times 10^{-4}$. Plot (d) shows the loss of utility for δ varying from 10^{-4} to 0.1, while fixing $\epsilon = 0.1$.

4.2. **Real Data Experiments.** Next, we evaluate our algorithm on two real data sets. First, we consider the Adult data set from the UCI Machine Learning Repository [Ronny Kohavi, 1996], which contains 48,842 records regarding individual's personal information. Twelve categorical features are used such as age, work class, educational level, and others[2]. The total number of possible distinct records is $\approx 5.5 \times 10^{12}$. Second, we consider the public use microdata files from the 2012 American Community Survey (ACS) from the United States Bureau of the Census[3]. Each record in this data set represents a household in the United States. The covariates include answers to questions sent to these households for estimating housing characteristics. The public data set includes approximately 1.5 million housing units; after pre-processing we use 200,000 records from this data set. In addition, there are 35 categorical variables, resulting in 1.85×10^{25} possible distinct records. In both data sets, the number of possible distinct records is much larger than the number of records, indicating highly sparse data sets.

Since the PeGS method is not directly comparable, as discussed in Section 3, we compare PrivBayes [Zhang et al., 2014]. Since PrivBayes satisfies ϵ-DP, we evaluate the performance of our algorithm by releasing a synthetic histogram with varying DP parameter ϵ while fixing $\delta = 10^{-6}$. In order to make a fully fair comparison, we incorporate the exponential mechanism from Remark 3.2 for releasing the mutual information, which ensures that our algorithm is DP throughout. In addition, we compare our proposed algorithms with the exponential mechanism (SBHG+EM) and without the exponential mechanism (SBHG) to SBA and PrivBayes on both data sets in Figure 4.

The loss of utility, as measured by the proposed L_1 error for SBA grows very quickly when ϵ becomes small, while the others methods are relatively stable for small ϵ. Compared to PrivBayes, the SBHG algorithm performs better than PrivBayes while SBHG+EM algorithm has similar performance to PrivBayes. In fact, these results are expected since

[2] Integer features such as age are treated as categorical features.

[3] http://www2.census.gov/acs2012_1yr/pums/

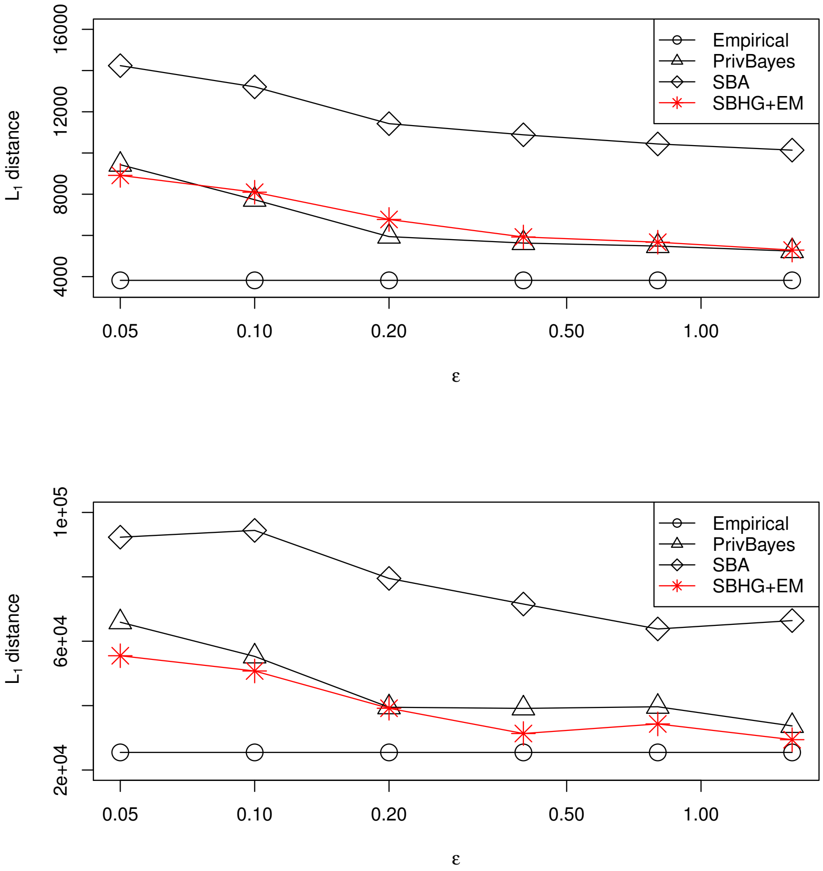

Figure 4: Comparison analysis of the Adult (upper) and ACS (lower) data sets. We set $\epsilon = 0.05, 0.1, 0.2, 0.4, 0.8, 1.6$ and test SBA, PrivBayes, and SBHG methods with optimal k. For SBA and SBGH+EM, we fix $\delta = 10^{-6}$.

both PrivBayes and SBHG utilize the conditional distributions as the basis for reconstructing histograms. That is, when the privacy risk is less restricted (larger ϵ), the two algorithms perform similarly in terms of utility. While SBHG always preserves the utility better, we cannot claim that SBHG is always better because from a privacy perspective the PrivBayes achieves ϵ-DP, while SBHG achieves (ϵ, δ)-DP. Nevertheless, in cases that are very difficult in practical data, SBHG does outperform PrivBayes much better. More specifically, when the data are significantly sparser, for example, in the case of the ACS versus the Adult data set, the SBHG performs much better from the utility perspective. In addition, it achieves the better utility risk trade-off, given that there is a low privacy cost, with $\delta = 10^{-6}$.

To study the utility of the synthetic datasets generated by SBHG for downstream tasks, we perform logistic regression on the resulting output from our algorithm and compare the classification accuracy. For the Adult dataset, we predict the income level as a binary variable $(< 50,000$ or $> 50,000)$ with all the other features. For the ACS dataset, we predict the

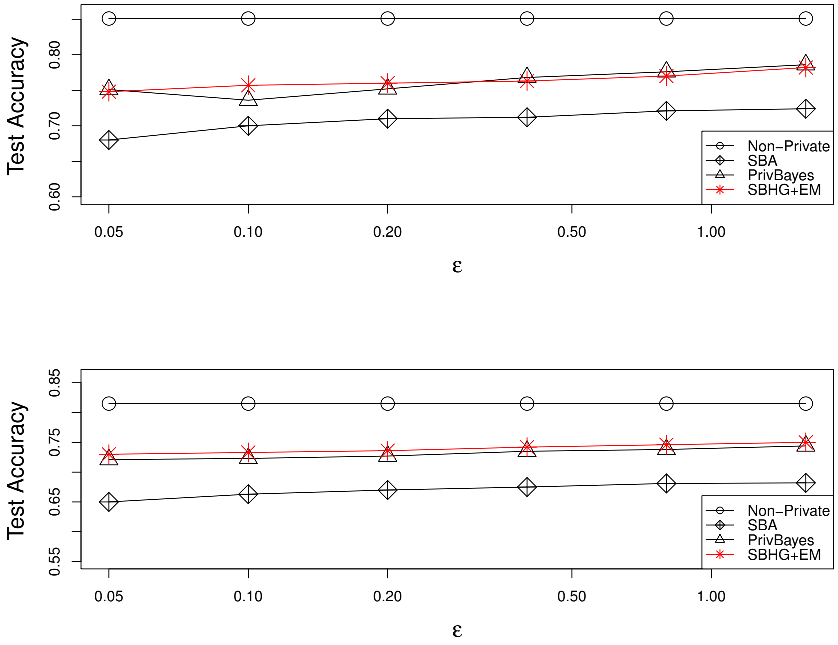

Figure 5: We perform logistic regression on the synthetic Adult (upper) and ACS (lower) data sets generated by SBA, SBHG+EM, PrivBayes, where $\epsilon = 0.05, 0.1, 0.2, 0.4, 0.8, 1.6$; $\delta = 10^{-6}$. We compare each method's test accuracy to logistic regression (non-private) applied to the original data sets.

Housing-Unit-level outcome variables FS, which is the indicator of receiving Food Stamps. Figure 5 for the Adult data set illustrates that SBHG+EM is either the same or outperform PrivBayes for certain values of ϵ, namely when ϵ decreases. For the ACS data set, we see that SBHG+EM performs roughly the same as PrivBayes. In short, as expected, we see better performance for smaller values of ϵ and when the data are sparse.

5. Discussion

In this paper, we have proposed the SBHG sampler, which releases high-dimensional sparse histograms and is provably (ϵ, δ)-DP and makes five major contributions to the literature. First, we propose an algorithm that approximates the empirical distribution, which is a good model with high statistical utility, and is made up of a collection of conditional distributions. Second, we propose an algorithm that handles support estimation under ϵ-DP, which is a well known and difficult problem [Wasserman, 2012]. We are able to do so by combining the SBA algorithm, which partially solves the support issue because the SBA allows for release of high-dimensional histograms without destroying their support. Any other issues regarding support are handled using a rejection type sampler via Gibbs sampling. Third, incorporating feature selection reduces the number and dimensionality of the resulting conditional histograms. As a result, we ensure that the sparsity pattern of the joint distribution of the high-dimensional histogram is partially preserved. Fourth, we illustrate our proposed methodology in both simulated and real experiments. Finally, we also perform logistic regression on the resulting output from our algorithm, comparing the classification accuracy as a downstream task of logistic regression to illustrate the utility of the resulting synthetic data sets.

Our paper has raised many important questions regarding future directions in releasing high dimensional sparse histograms. First, we have used a simplistic form of feature hashing in this paper. Future directions may wish to look at more optimal ways of including the feature selections step to further maximize the amount of dimension reduction. We suspect that looking at complex feature hashing would require relaxing DP, however, looking at this regime and the tradeoffs would be very useful in practice. Another point of future research would be investigation of the distribution of the hashed Gibbs sampler. In fact, it is unclear what the stationary distribution is and what the price to pay is from using such dimension reduction methods. Finally, a very important and open question in the privacy literature is the price to pay of estimating high dimensional sparse histograms without privacy. We have made some contributions to this in our empirical analysis in our paper, however, quantifying this price regarding theoretical and optimality guarantees regarding private versus non-private histogram release for sparse histograms would further solidify our proposed methodology and push forward an important area of DP [Valiant and Valiant, 2016].

Acknowledgment

This research was supported in part by NSF Grants SES-1534412 and CAREER-1652431 to Duke University, and SES-1534433 to Pennsylvania State University.

References

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

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

V. Balcer and S. Vadhan. Differential privacy on finite computers. *arXiv preprint arXiv:1709.05396*, 2017.

M. Bun, K. Nissim, and U. Stemmer. Simultaneous private learning of multiple concepts. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 369–380. ACM, 2016.

A.-S. Charest. How can we analyze differentially-private synthetic datasets? *Journal of Privacy and Confidentiality*, 2(2):3, 2011.

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 and L. Vilhuber. Replicating the synthetic lbd with german establishment data. 2013.

D. B. Dunson and C. Xing. Nonparametric bayes modeling of multivariate categorical data. *Journal of the American Statistical Association*, 104(487):1042–1051, 2009.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pages 265–284. Springer, 2006.

C. Dwork, A. Smith, T. Steinke, and J. Ullman. Exposed! a survey of attacks on private data. *Annual Review of Statistics and Its Application*, 4:61–84, 2017.

S. E. Fienberg and A. B. Slavkovic. Preserving the confidentiality of categorical statistical data bases when releasing information for association rules. *Data Mining and Knowledge Discovery*, 11(2):155–180, 2005.

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

A. Gelman and T. Speed. Corrigendum: Characterizing a joint probability distribution by conditionals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(2):483–483, 1999.

M. Götz, A. Machanavajjhala, G. Wang, X. Xiao, and J. Gehrke. Privacy in search logs. *arXiv preprint arXiv:0904.0682*, 2009.

R. Hall, A. Rinaldo, and L. Wasserman. Random differential privacy. *arXiv preprint arXiv:1112.2680*, 2011.

K. Jung, S. Shavitt, M. Viswanathan, and J. M. Hilbe. Female hurricanes are deadlier than male hurricanes. *Proceedings of the National Academy of Sciences*, page 201402786, 2014.

V. Karwa, P. N. Krivitsky, and A. B. Slavković. Sharing social network data: differentially private estimation of exponential family random-graph models. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 66(3):481–500, 2017.

S. K. Kinney, J. P. Reiter, A. P. Reznek, J. Miranda, R. S. Jarmin, and J. M. Abowd. Towards unrestricted public use business microdata: The synthetic longitudinal business database. *International Statistical Review*, 79(3):362–384, 2011.

A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. In *Proceedings of the 18th international conference on World wide web*, pages 171–180. ACM, 2009.

J. Lane, V. Stodden, S. Bender, and H. Nissenbaum. *Privacy, Big Data, and the Public Good: Frameworks for Engagement.* Cambridge University Press, 2014.

J. Lei, A.-S. Charest, A. Slavkovic, A. Smith, and S. Fienberg. Differentially private model selection with penalized and constrained likelihood. *arXiv preprint arXiv:1607.04204*, 2016.

A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In *24th International Conference on Data Engineering (ICDE)*, pages 277–286. IEEE, 2008.

Y. Park and J. Ghosh. Perturbed Gibbs Samplers for synthetic data release. Technical report, 2013. URL `arXiv:.1312.537`.

T. E. Raghunathan, J. P. Reiter, and D. B. Rubin. Multiple imputation for statistical disclosure limitation. *JOURNAL OF OFFICIAL STATISTICS-STOCKHOLM-*, 19(1): 1–16, 2003.

J. Reiter. Using cart to generate partially synthetic public use microdata. *Journal of official statistics*, 21(3):441–462, 2005a.

J. P. Reiter. Releasing multiply imputed, synthetic public use microdata: an illustration and empirical study. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 168(1):185–205, 2005b.

B. B. Ronny Kohavi. UCI machine learning repository, 1996. URL http://archive.ics.uci.edu/ml.

D. B. Rubin. Satisfying confidentiality constraints through the use of synthetic multiplyimputed microdata. *Journal of Official Statistics*, 9(2):461–468, 1993.

A. B. Slavkovic. Partial information releases for confidential contingency table entries: present and future research efforts. *Journal of Privacy and Confidentiality*, 1(2):9, 2010.

S. Vadhan. The complexity of differential privacy. 2016.

G. Valiant and P. Valiant. Instance optimal learning of discrete distributions. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 142–155. ACM, 2016.

L. Wasserman. Minimaxity, statistical thinking and differential privacy. *Journal of Privacy and Confidentiality*, 4(1):3, 2012.

L. Wasserman and S. Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.

J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. Privbayes: Private data release via bayesian networks. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1423–1434. ACM, 2014.

Appendix A. Simulated Data

In this section, we provide the histogram of the simulated dense data set. In this histogram, each bar corresponds one type of record in the original data set. The height of the bar indicates the frequency of that specific type of record. The histogram suggests that each type of record appears at least once in the simulated data set, which forms a dense set.

Appendix B. The Laplace Mechanism

In this section, we give the algorithm for the Laplace Mechanism in Algorithm 3.

Algorithm 3 Laplace Mechanism

```
Input: $c_1, \ldots, c_m$, $\epsilon$
Output: $\epsilon$ differentially private histogram
1: To each count $c_i$, add Laplace noise with mean 0 and scale parameter $\frac{2}{\epsilon}$
2: Output the noisy histogram.
```

Appendix C. The Gibbs Sampler

In this section, we review the standard Gibbs sampler.

(1) Construct the conditional distribution of each feature, i.e., $\mathbb{P}\left(X_j | X_{-j} = x_{-j}\right)$, $j = 1, \ldots, p$.
(2) Let $x^0 = \{x_1^0, \ldots, x_p^0\}$ be any initial point.
(3) For $t = 1, 2, \ldots, N$ repeat the following:
   (a) Set $x_j^t = x_j^{t-1}$.
   (b) For each $j = 1, \ldots, p$, sample a point x_j^t from $\mathbb{P}\left(X_j | X_{-j} = x_{-j}^{t-1}\right)$.

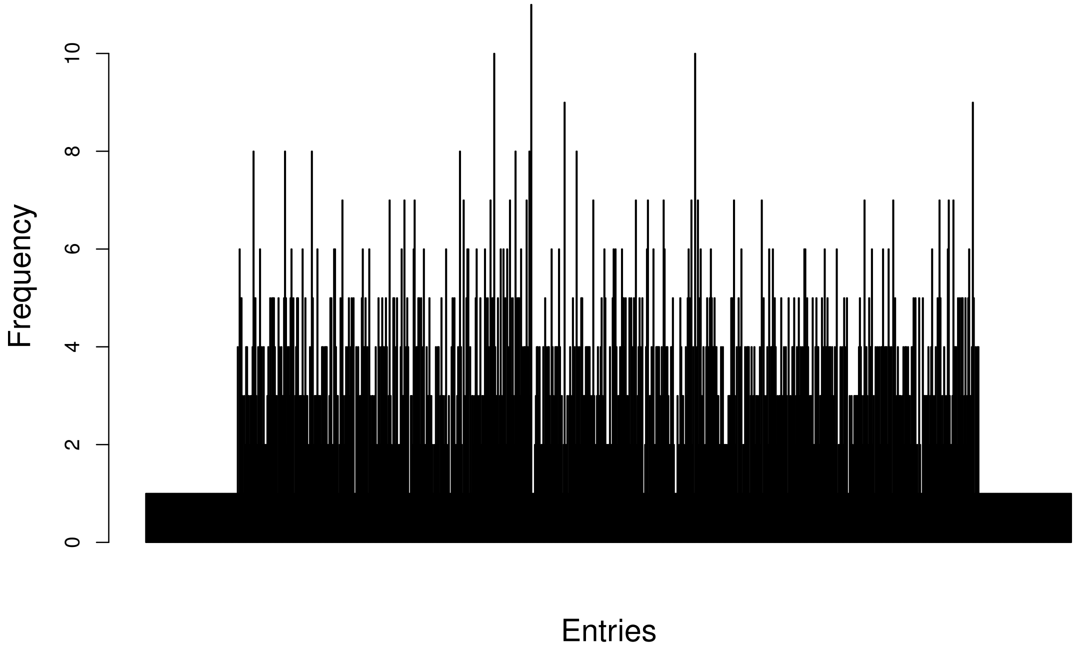

Figure 6: Histogram of the simulated dense data set.

Appendix D. Simulation Settings

In this section, we describe our overall setup for our simulation studies. For both our simulation studies, we randomly generate records by first drawing samples from a 10 dimensional multivariate normal distribution. Next, we bin each feature into 10 categories. Then we set the correlations between each pair of features to be 0.3 from the multivariate normal distribution. This assumption is made as it allows us to study the effect of feature selection given that it depends on the mutual information between the features.