SDNist: Benchmark Data and Evaluation Tools for Data Synthesizers

Grégoire Lothe,[2] Christine Task,[3] Isaac Slavitt,[4] Nicolas Grislain,[2] Karan Bhagat,[3] Gary S Howarth,[1]

[1] National Institute of Standards and Technology, Public Safety Communications Research Division,
[2] Sarus Technologies,
[3] Knexus Research Corporation,
[4] DrivenData, Inc.
gl@sarus.tech, christine.task@knexusresearch.com, isaac@drivendata.org, karan.bhagat@knexusreserach.com, ng@sarus.tech, gary.howarth@nist.gov,

Abstract

We present a set of benchmark data and metrics for the evaluation of synthetic data generators on structured tabular data. These benchmarks are distributed as a simple open-source Python package to allow standardized and reproducible comparison of synthetic generator models on real-world data and use cases. These data and metrics were developed for and vetted through the NIST PSCR Differential Privacy Temporal Map Challenge; the evaluation tools, $k-$marginal and Higher Order Conjunction, proved effective in distinguishing competing models in the competition environment.

Introduction

High performing, differentially private synthetic data generators have the potential to unlock sensitive datasets, allowing for the exchange of valuable information while strictly limiting how much can be learned about an individual record in the dataset. When developing new approaches to private synthetic data, researchers and practitioners need public and reproducible benchmarks.

Here, we present SDNist (SDNist source code 2021), a set of benchmark data and evaluation metrics packaged as an open-source Python implementation tailored for the evaluation of synthetic data generators.

Shared benchmarks allow researchers and developers the ability to compare their results with common (pseudo)metrics against the same datasets without onerous preparation. The ambiguity surrounding scientific communication on dataset preparation and synthetic data generation harms reproducibility. This lack of objective, common methods of evaluating the quality of private synthetic data is a key barrier to exploration and adoption of its use. For this reason, fully automated benchmark data, evaluation methods, and metrics computation expressed in a programming language — hence unambiguous — are needed to compare models, examine limits and capabilities, test performance, and track progress over time in a reproducible, error-free and rigorous way.

These benchmarks were developed from the 2020 Differential Privacy Temporal Map Challenge (Task et al. 2021), and informed by the previous 2018 Synthetic Data Challenge (Ridgeway et al. 2021), both hosted by the Public Safety Communications Research Division (PSCR) of the National Institute of Standards and Technology (NIST). This online competition was cash incentivized and run in three sprints on the DrivenData platform (Bull, Slavitt, and Lipstein 2016), where contestants trained their algorithms on public data and submitted containerized versions of their code for scoring on sequestered data of the same schema.

The Differential Privacy Temporal Map Challenge focused on synthesizing time-stamped, geographically-aggregated data in which one individual may contribute to a sequence of events. These types of data capture a diverse array of applications, including epidemiology and medical records, transaction histories from marketing data, longitudinal demographic data, and even traffic data for civic planning.

The benchmarks presented here are derived from challenge sprints that sought to privatize records from the American Community Survey (ACS) (U.S. Census Bureau 2012-2018) and the City of Chicago taxi cab database (City of Chicago Data Portal 2016-2020). These data highlight important challenges found in many real-world applications, such as geographical heterogeneity, sparse data, and longitudinal sequence privacy. These characteristics pose particular difficulties for both differential privacy and synthetic data generation. We provide specifically calibrated scoring functions, described in detail below, to optimize for preservation of geographic and temporal trends.

Using the resources developed in the challenge environment, we have packaged test datasets and evaluation metrics for the efficient comparison of synthetic data generators. The metrics described here have demonstrated their ability to statistically distinguish sets of outputs from competing algorithms in a competition environment. The data, organized in training and evaluation sets, are drawn from real sources, representing genuine use cases.

Related work

There are plenty of standard datasets and associated machine-learning problems such as those from UCI (UC Irvine 2021), Kaggle (Kaggle 2021), DrivenData (DrivenData 2021) or Papers-with-code (Papers with Code 2021). But there are relatively few benchmarks to evaluate pri-

vate synthetic data generators. A recent initiative, SDGym (SDGym 2021; Xu et al. 2020), brings standard problems, an evaluation methodology for synthetic data and many generators to compare to (The Synthetic Data Vault 2021; Patki 2016). Unfortunately, SDGym does not consider differential privacy in its evaluation, it is missing the *privacy-loss* (ϵ) dimension in its output benchmarks and also some sort of empirical testing of privacy guarantees, such as those developed in (Jayaraman and Evans 2019; Wilson et al. 2019).

Background

Differential privacy

Differential privacy is a theoretical framework to measure and bound the privacy loss occurring when one publishes an analysis on a private dataset (Dwork and Roth 2013). It provides a strong protection by requiring that the revealed information does not allow users to infer the presence of an individual in the dataset.

We now move on to more formal definitions. Consider a dataset, D, which is comprised of columns corresponding to features and rows corresponding to observations about individuals. Each individual can have several observations. We say that two datasets D and D' are neighbors if they differ by omission of one individual. The output $\mathcal{M}(D)$ of a randomized mechanism $\mathcal{M}$ is (ϵ, δ)-differentially private if for all pairs of neighboring datasets and all sets S of possible outputs, we have

$$\mathbb{P}(\mathcal{M}(D) \in S) \leq \exp(\epsilon)\mathbb{P}(\mathcal{M}(D') \in S) + \delta$$

ϵ is usually referred to as the *privacy-loss budget*, and δ, a parameter that relaxes absolute privacy guarantees. A smaller ϵ yields stronger protections and privacy guarantees. Typical values of δ are $\delta \leq 1/n^2$ where n is the number of individuals in the dataset.

Synthetic data A useful property of the differential privacy framework is the *post-processing* property. It states that any post-treatment of the output of a mechanism does not incur any further privacy loss. This hints towards a generic and streamlined method of privacy-preserving analysis. Indeed, if a private dataset can be replaced by a synthetic version generated under differential privacy constraints, the post-processing property guarantees that there are no treatments we can apply to the synthetic dataset that recover meaningful information about actual individuals.

The challenge then becomes to synthesize a dataset that is *close enough* to the private one so that the results of the subsequent analyses are meaningful, while ensuring a differential privacy constraint. In recent years, this problem has received considerable attention in the scientific literature (Zhang et al. 2014; Abay et al. 2018; Ge et al. 2021; Yoon, Jordon, and van der Schaar 2019). A key issue is how to evaluate such closeness: for discrete distributions, there exist several metrics which do not require an estimate of the underlying density, such as the *Wasserstein metric* (Peyré and Cuturi 2020) or the *Maximum Mean Discrepancy* (Gretton et al. 2012). However, due to the underlying dimension and size of real-life datasets, we focus on simpler approaches to quantify the utility of the synthetic data.

Table 1: Dataset transformation example for longitudinal data

individual id	year	age
0	2012	36
0	2013	37
0	2014	38
1	2012	19
1	2014	21
⋮		

→

age_{2012}	age_{2013}	age_{2014}
36	37	38
19	?	21

$k-$marginal score

In this section we describe a method, $k-$marginal, to quantify the utility of the synthetically generated data. This concept has also been called Mean Absolute Density Difference (Raab, Nowok, and Dibben 2021).

Ideally, any statistical analysis query performed on the real or the synthetic dataset should yield the same results. However, when dealing with statistical datasets with many features, analysts typically only select a few features that they consider as relevant for their tasks and study correlations among them. For instance, on census data, an analyst could be interested in answering *"how has the level of education impacted the annual income in Ohio since 2010?"*

Even though the number of features d of the original dataset can be quite large, only $k = 2$ or $k = 3$ features are usually considered at once, so that for most applications, two datasets can be considered close if all $k-$way marginals are close in some norm. This greatly reduces problem size.

Furthermore, for categorical datasets, a simple metric to compare $k-$way marginals is the total variation norm, which can be computed efficiently on low-dimensional histograms. To alleviate the burden of computing d *choose* k distributions when the dataset contains a large number of features, we randomly select a set of permutations of k features and take the average total variation norm over the corresponding $k-$way marginals. We notice that taking a reasonable subset of all possible permutations still yields stable results.

Extension to individual longitudinal data Most real-life datasets include repeated measurements about their individuals. When several rows of the dataset might refer to a single individual, the $k-$marginal metric introduced in the previous section does not directly capture how the behavior of each individual differs in both datasets.

It turns out that there is often a reorganization of datasets that allows the $k-$marginal scoring to take the temporal dependency into account. For instance, if the measurements are evenly spaced out, one can "unstack" the rows of an individual into new features. Using m to denote the number of occurrences of each individual, the number of rows is divided by at most m while the number of features is itself multiplied by m. An example of such transformation is given in table 1. Note that such transformation does not need to be explicitly constructed in memory: the computations can be performed directly on the original data. This is effectively equivalent to computing a $k-$marginal score on the combinations of *(feature, measurement_index)* instead of *(feature)*.

Higher Order Conjunction (HOC)

Like $k-$marginal, the Higher Order Conjunction (HOC) metric was developed for the 2018 NIST Differential Privacy Synthetic Data Challenge (Ridgeway et al. 2021). The HOC metric measures distributional similarity relative to all information associated with a given individual. Like the $k-$marginal metric, it is a randomized heuristic that is computed for many iterations. A single iteration of the HOC metric begins by sampling one target individual from the ground truth data and generating a randomized similarity constraint; for each feature i, it samples a similarity range $(i-k_i, i+k_i)$. If another individual falls within this range $\forall i$, they are considered to be similar to the target individual. The metric then computes the percentage of individuals that are similar to the target individual in both the ground truth and synthetic data, and outputs the absolute difference as the score of this iteration. The final score is averaged across all iterations.

Benchmark Problems

SDNist contains two benchmark problems for use in researching and developing synthetic data generators, one drawn from Chicago taxi trips, and the other from the American Community Survey. These problems and their associated data share several key properties which are helpful for the development of synthetic data generators:

Longitudinal data Each problem consists of varying length sequences of individual records; the maximum records per individual is given as an input parameter. An individual ID feature is used to join the records associated with a single individual. To evaluate performance on non-longitudinal data (i.e., providing privacy at the single record level), the individual ID feature may be ignored.

Use of public data To allow for algorithm development, we assume a common real-world scenario: that the data owners are transitioning from releasing a simply anonymized data product to releasing privacy-preserving synthetic data, and thus there exist public data that can be used for algorithm evaluation and tuning, without loss of privacy. Each benchmark problem has public development data for use in algorithm training and parameter tuning, and at least two private data sets to be used for final scoring. The final scoring data shares the same schema as the development data, but has significantly different distributions to discourage overfitting the public data.

Choice of (ϵ, δ) For general use the benchmark problems may be run at any value of epsilon or delta. To compare performance against the challenge leaderboard, solutions should be scored three times with $\epsilon \in \{0.1, 1, 10\}$ for the ACS problem, $\epsilon \in \{1, 2, 10\}$ for the Taxi problem, and with negligible $\delta = 1/n^2$. These values were selected to encompass both smaller epsilon values that are often used in research, and larger values used in real-world practice.

Data Characteristics

American Community Survey (ACS) The ACS data schema has 35 features, a maximum number of 7 records per individual, covers 7 years as time segments, and contains up to 181 geographic regions per data set.

Processing: The source data contains only one record per individual; to artificially create longitudinal data, records from different years were joined to form simulated individuals with 4-7 annual records. Traits such as age, sex, race, education, and citizenship were used as hard constraints in the matching (for example, ensuring age and education incremented appropriately between years, and citizenship wasn't lost), and the algorithm prioritized matches with similar incomes and geographic locations. These simulated individuals capture the general income and population trends of their geographies, and provide a very realistic challenge in terms of higher sensitivity marginal queries. However, they may not reflect fine-grained individual behavior, which is better addressed in the Taxi problem.

Characteristics: The ACS data is designed to highlight common properties of real-world survey data that are key to effective synthetic data generation. Map segments range from dense and homogeneous to sparse and diverse, and scoring prioritizes good performance across all conditions. Demographic and economic shifts in the data are very clear over the 7 year time-span, which covers a period of significant urban changes in the United States. Within the data schema itself, there are hard constraints and tight correlations across time segments and between select features, which can be leveraged to reduce the data space or used in post-processing to significantly improve performance.

Chicago Taxi Data The Chicago Taxi data schema has 13 features, a maximum number of 200 records per individual, 21 shifts as time segments, and contains 78 geographic regions per data set.

Processing: The original source data contains a complete year of data for each taxi driver, resulting in more than 5,000 trip records for most individuals. To reduce difficulty, every ground truth individual was split into 52 simulated individuals, each containing one continuous week of data (and at most 200 trips). Because each simulated individual is drawn directly from a real driver's data, this problem is designed for evaluating models that preserve fine-grained distributions of individual behaviors.

Characteristics: The large number of records per individual, large proportion of sparse map segments, and the requirement to maintain correlations between map segments increase the problem difficulty for differentially private algorithms. However, typical of real-world geographic data, public information such as the distance between neighborhoods can be leveraged to reduce the problem complexity. This problem requires generating a realistic distribution of individual taxi drivers, as well as a realistic distribution of trips. The data sets are large and relatively rich. Major holidays, universities, airports and major industries, the impact of historical red-lining, drivers' preferred shifts and home regions, rising competition from ride-share apps, and the impact of the Covid-19 pandemic are all visible in the data.

Scoring

In this section, we describe the different scoring methods used during each sprint. In each case, the score is computed at least four times and averaged to smooth out the intrinsic randomness of the generation algorithms.

ACS The score computed during this sprint is a variant of the $k-$marginal score introduced in the previous section. Instead of computing the $k-$marginal score over the whole dataset, the score is computed over each pair of *(PUMA, year)*. This strategy is aimed at enforcing representation across geographies.

For future endeavours, we propose to rely on the $k-$marginal extension introduced early in this work, i.e., we propose to compute the $k-$marginal score on combinations of features at given years (similar to table 1), instead of features. This improvement is more responsive to individual behaviors.

Chicago Taxi Data The score of this sprint is the average between a variant, described below, of the $k-$marginal and the Higher-Order Conjunction (HOC) scores. Again,the $k-$marginal evaluates the distributional similarity by sets of columns. Whereas, the HOC evaluates similarity across linked records, thus considering the fidelity of behavior of an individual taxis

We also propose to further extend the $k-$marginal score to measure individual consistencies, as described above in the extension of k-marginal to individual longitudinal data.

Challenge and baseline results

The results of the challenge are presented in table 2. Each score is averaged over 4 runs and remapped from 0 (worse) to 1000 (best). The $\pm$ sign indicates standard deviation. For comparison, the score of subsampled datasets is presented in table 3. Note that subsampling a dataset does not provide any (ϵ, δ) differential privacy guarantee for $\epsilon < \infty$. At the time of writing this article, several contestants have open-sourced their submission or provided insights about their techniques in scientific papers (McKenna, Miklau, and Sheldon 2021; Li, Zhang, and Wang 2021).

Benchmark wide release

The benchmark, including the datasets and the scoring functions of each challenge, is readily available for all users as a Python package (sdnist python package 2021). The library allows differential privacy researchers to evaluate the quality of their synthetic data against real-life datasets and compare their results against the top-ranking teams of the challenge. The package focuses on emphasizing the reproducibility of each algorithm while progressively incorporating new metrics that better capture the subtleties of each dataset and push the domain of synthetic data generation further.

Roadmap

Although `sdnist` proposes some standard problems and some standard synthetic data generators inspired from the past NIST challenges available as an easy-to-use Python package, many improvements are to be developed. We plan to add more problems, more standard models to compare to, and new metrics to evaluate synthetic data quality. Furthermore, the package will eventually propose empirical differential privacy evaluation such as those proposed by (Jayaraman and Evans 2019; Wilson et al. 2019).

Table 2: Challenge results

(a) ACS dataset, $k-$marginal score

	$\epsilon = 0.1$		$\epsilon = 1$		$\epsilon = 10$	
Dataset	(1)	(2)	(1)	(2)	(1)	(2)
N - CRiPT	781±2	807±2	**851**	**865**±1	**893**	**901**
Duke Privacy	796±1	816±3	832	852	881	890
Minutemen	**822**±1	788±1	825±1	834	873	881
DPSyn	805±3	**822**±1	818±1	844±1	822	848±1
Jim King	782±2	803±2	790	814	840	846

(b) Taxi dataset, $k-$marginal score

	$\epsilon = 1$		$\epsilon = 10$	
Dataset	2016	2020	2016	2020
Minutemen	**464**±3	**455**±15	**556**±5	491±3
N - CRiPT	340±7	437±18	456±2	**700**±2
DPSyn	344±1	433±3	416±1	464±2
GooseDP-PSA	251±2	382±1	251±1	382±1

(c) Taxi dataset, HOC score

	$\epsilon = 1$		$\epsilon = 10$	
Dataset	2016	2020	2016	2020
DPSyn	922	**942**	917	**945**±1
N-CRiPT	924	872±1	924	880
DP Duke	857±22	982±7	900±27	898±15
Minutemen	**931**	918	**929**	817
Jim King	828	845±1	839	885±2
GooseDP-PSA	865	827	864	827

On each table and for each value of ϵ, the left and right column indicate the score on the public and the private leaderboard respectively. (1)=NY-PA, (2)=GA-NC-SC

Table 3: Subsampling baseline, $k-$marginal score

	Census		Taxi	
Fraction	(1)	(2)	2016	2020
1%	572±1	590±1	547±1	472±1
10%	831	839	721	703
50%	940	944	889	887

References

Abay, N.; Zhou, Y.; Kantarcioglu, M.; Thuraisingham, B.; and Sweeney, L. 2018. Privacy Preserving Synthetic Data Release Using Deep Learning. 510–526. ISBN 978-981-13-6048-0.

Bull, P.; Slavitt, I.; and Lipstein, G. 2016. Harnessing the Power of the Crowd to Increase Capacity for Data Science in the Social Sector. *CoRR*, abs/1606.07781.

City of Chicago Data Portal. 2016-2020. Taxi Trips.

DrivenData. 2021. https://www.drivendata.org/.

Dwork, C.; and Roth, A. 2013. The Algorithmic Founda-

tions of Differential Privacy. *Foundations and Trends in Theoretical Computer Science*, 9.

Ge, C.; Mohapatra, S.; He, X.; and Ilyas, I. F. 2021. Kamino: Constraint-Aware Differentially Private Data Synthesis. arXiv:2012.15713.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(25): 723–773.

Jayaraman, B.; and Evans, D. 2019. Evaluating differentially private machine learning in practice. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 1895–1912.

Kaggle. 2021. https://www.kaggle.com/.

Li, N.; Zhang, Z.; and Wang, T. 2021. DPSyn: Experiences in the NIST Differential Privacy Data Synthesis Challenges. arXiv:2106.12949.

McKenna, R.; Miklau, G.; and Sheldon, D. 2021. Winning the NIST Contest: A scalable and general approach to differentially private synthetic data. arXiv:2108.04978.

Papers with Code. 2021. https://paperswithcode.com/datasets.

Patki, N. 2016. *The Synthetic Data Vault: generative modeling for relational databases*. Ph.D. thesis, Massachusetts Institute of Technology.

Peyré, G.; and Cuturi, M. 2020. Computational Optimal Transport. arXiv:1803.00567.

Raab, G. M.; Nowok, B.; and Dibben, C. 2021. Assessing, visualizing and improving the utility of synthetic data. arXiv:2109.12717.

Ridgeway, D.; Theofanos, M. F.; Manley, T. W.; and Task, C. 2021. Challenge Design and Lessons Learned from the 2018 Differential Privacy Challenges. Technical report.

SDGym. 2021. https://github.com/sdv-dev/SDGym.

sdnist python package. 2021. https://pypi.org/project/sdnist/.

SDNist source code. 2021. https://pypi.org/project/sdnist/.

Task, C.; Slavitt, I.; Lipstein, G.; Streat, D.; and Howarth, G. 2021. NIST PSCR Differential Privacy Temporal Map Challenge.

The Synthetic Data Vault. 2021. ttps://sdv.dev/.

UC Irvine. 2021. UC Irvine Machine Learning Repository.

U.S. Census Bureau. 2012-2018. American Community Survey Public Use Microdata Sample.

Wilson, R. J.; Zhang, C. Y.; Lam, W.; Desfontaines, D.; Simmons-Marengo, D.; and Gipson, B. 2019. Differentially Private SQL with Bounded User Contribution. arXiv:1909.01917.

Xu, L.; et al. 2020. *Synthesizing tabular data using conditional GAN*. Ph.D. thesis, Massachusetts Institute of Technology.

Yoon, J.; Jordon, J.; and van der Schaar, M. 2019. PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees. In *International Conference on Learning Representations*.

Zhang, J.; Cormode, G.; Procopiuc, C. M.; Srivastava, D.; and Xiao, X. 2014. PrivBayes: private data release via bayesian networks. In Dyreson, C. E.; Li, F.; and Özsu, M. T., eds., *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, 1423–1434. ACM. ISBN 978-1-4503-2376-5.