# 1 Algorithm description

We give an overview of the PRIVATE-GSD mechanism and how it satisfies differential privacy. There are two main steps: 1) measure which adds independent Gaussian noise to the answer of a large set of statistical queries on the data. 2) The project steps, which finds a synthetic data that best explains the noisy statistics.

Differential privacy is enforce only by the measure steps. The project steps only has access to the noisy statistics, thus, it satisfies differential privacy by post-processing.

## 1.1 Gaussian mechanism

For the problem of estimating statistics $Q(D)$ subject to differential privacy, we follow the framework of the projection mechanism [Nikolov et al., 2013, Dwork et al., 2015], which consists of perturbing the statistics with a differentially private mechanism and then finding a synthetic dataset that matches these perturbed statistics.

In this study, we consider two versions of the private projection mechanism. The first is the standard projection mechanism, referred to as the *one-shot* version. The second is an *adaptive* version of the projection mechanism, which proves useful in scenarios where a large number of queries are required. A comprehensive description of both frameworks is presented below.

**One-shot projection mechanism:** Suppose we are given $m$ statistical queries, represented by $Q$, and a dataset $D$. Then the first step consists of independently perturbing every coordinate of $Q(D)$ to obtain a private estimate, denoted by $\hat{a}$. Given a privacy parameter $\rho$, the noisy answer vector is computed as follows:

$$\hat{a} \leftarrow Q(D) + \mathcal{N}\left(0, \frac{\Delta(Q)}{2\rho} \cdot \mathbb{I}\right) \tag{1}$$

Then, the *project* step involves finding a synthetic dataset $\widehat{D}$ that best explains the estimate $\hat{a}$. To represent the projection step, we construct a loss minimization problem, where the loss function is defined by $\hat{a}$ and $Q$ as follows:

$$L_{\hat{a},Q}(\widehat{D}) = \|\hat{a} - Q(\widehat{D})\|_2^2 \tag{2}$$

By the properties of the Gaussian mechanism and the post-processing property , the projection mechanism framework satisfies $\rho$-zCDP regardless of the optimization procedure used to solve (2).

**Implementation** The code that implements differential privacy is found in the the function

```
def private_measure_all_statistics(self, key: chex.PRNGKey, rho: float)
    # Gaussian mechanism.
```

from the file *stats/chained_statistics.py*. This function computes the sensitivity of the statistics and then adds Gaussian noise.

## 1.2 Projection step: A genetic algorithm.

The generation of high-quality synthetic data through the "projection mechanism" framework, as described in **??**, presents a challenging task that necessitates the solution of an NP-hard optimization problem. Hardt and Talwar [2010]. Therefore, we propose the PRIVATE-GSD algorithm, which solves the projection step in the "projection mechanism" using a genetic algorithm (GA). GAs are a class of optimization algorithms inspired by natural selection and genetic recombination, which have found extensive application in artificial intelligence. A crucial advantage of GAs is that they do not require differentiability of the optimization objective.

Genetic algorithms (GAs) are a class of optimization algorithms inspired by the process of natural selection and genetic recombination. Introduced by Holland [1992], GAs provide a heuristic search technique to solve complex optimization problems by evolving a population of candidate solutions towards an optimal or near-optimal solution. The basic structure of a genetic algorithm consists of the following components: representation, initialization, evaluation,

**Algorithm 1** Private Genetic Algorithm for Synthetic Data (PRIVATE-GSD)

**Require:** The search space $\mathcal{X}^n$ of mixed-type synthetic datasets with $n$ rows and $d$ features, number of generations $G$, population size $P$, elite set size $E$.
**Input:** A set of $m$ queries $Q : \mathcal{X}^n \to [0,1]^m$, (noisy) query answers $\hat{a} \in [0,1]^m$.
Set the objective function for the synthetic datasets:

$$L(\widehat{D}) = \|\hat{a} - Q(\widehat{D})\|_2$$

Initialize elite set $\mathcal{E}_1 \leftarrow \{\bar{D}_{1,i} \in \mathcal{X}^n : i \in [E]\}$.
**for** $g = 1, \ldots, G$ **do**
  Set $\widehat{D}_g^\star \leftarrow \arg\min_{\widehat{D} \in \mathcal{E}_g} L(\widehat{D})$.
  Define a population of $P + E$ synthetic datasets:

$$\mathcal{P}_g = \{\widetilde{D}_{g,1}, \ldots, \widetilde{D}_{g,P}\} \cup \mathcal{E}_g$$

  where each $\widetilde{D}_{g,j}$ as follows:
  **for** $p = 1, \ldots, P$ **do**
    **Mutate:** Sample a row $i \sim U([n])$, feature $j \sim U([d])$ and set $\widetilde{D}_{g,p}$ as

$$\forall_{r \in [n], c \in [d]} \quad \widetilde{D}_{g,p}(r,c) = \begin{cases} v \sim U(\mathcal{X}_j) & \text{if } r = i \text{ and } c = j \\ \widehat{D}_g^\star(r,c) & \text{otherwise} \end{cases}$$

    **Cross:** Sample two rows $i_1, i_2 \sim U([n])$, a feature $j \sim U([d])$, and elite member $\bar{D} \sim U(\mathcal{E}_g)$ and set $\widetilde{D}_{g,p}$:

$$\forall_{r \in [n], c \in [d]} \quad \widetilde{D}_{g,p}(r,c) = \begin{cases} \bar{D}(i_2,c) & \text{if } r = i_1 \text{ and } c = j \\ \widehat{D}_g^\star(r,c) & \text{otherwise} \end{cases}$$

  **end for**
  Set $\mathcal{E}_{g+1}$ as top $E$ members from population $\mathcal{P}_g$ with respect to *minimizing* objective $L(\cdot)$.
**end for**
**return** $\widehat{D}_G^\star$

selection, crossover, mutation, and termination Golberg [1989]. A population of candidate solutions, called chromosomes or individuals, is encoded using a suitable representation. Then, the genetic process commences with an initial population of $E$ elite individuals, initialized at random, and operates over $G$ generations. On each generation $g \in [G]$, a new population of candidate solutions is generated by randomly applying crossover (recombination) and mutation operators over the set of elite candidates. Thus, each candidate solution inherits the traits of their elite parents. Each candidate is assigned a fitness value based on an objective function. And individual candidates with higher fitness values will be selected as next generation of elite candidates. This iterative process continues until a termination criterion is met, such as reaching a predefined number of generations or achieving a satisfactory level of fitness.

Genetic algorithms (GAs) offer several benefits for optimization problems; however, GAs do not represent a one-fit-all solution since they require a suitable problem representation (i.e., encoding) and choice of genetic operators to work effectively. For instance, the choice of problem representation can significantly affect the GA's computational efficiency because they typically require multiple iterations and evaluations of the objective function, which can be time-consuming. Furthermore, failure to choose suitable genetic operators could produce poor candidates or a lack of diversity in the population, making the GA suffer from slow convergence or convergence to the local minimum.

Therefore, in this work, we introduce a GA tailored to synthetic data generation that avoids the limitations above of GAs. For example, our algorithm PRIVATE-GSD algorithm exhibits fast convergence for optimizing the projection step and is computationally efficient. PRIVATE-GSD introduces two critical genetic operators, *sparse mutations* and *row crossover*, tailored to finding synthetic data consistent with a large set of statistical properties.

**Encoding** The first step of GA is defining an appropriate encoding representation. Since of goal is to generate synthetic data, PRIVATE-GSD encodes candidate solutions as datasets with $n$ rows, where $n$ is a parameter of the algorithm. Therefore, PRIVATE-GSD's solution space is given by $\mathcal{X}^n = \mathcal{X}_1^n, \ldots, \mathcal{X}_d^n$, where $d$ is the data dimension and each $\mathcal{X}_i$ is the schema of the $i$-th feature, which could be real-valued or discrete.

**Sparse Mutation.** In a genetic algorithm, the mutation phase serves as a critical component that introduces small random perturbations to the set of candidate solutions to promote diversity and discover better solutions. The particular algorithm PRIVATE-GSD employs mutation operations on the most optimal dataset up to a certain point, symbolized by $\widehat{D}_g^\star$, with the intention of generating novel candidate solutions.

The mutation procedure, as described in algorithm Algorithm 1, commences with the random selection of a row, denoted as $i \in [n]$, as well as a column, represented by $j \in [d]$. Subsequently, the entry of $\widehat{D}_g^\star$ corresponding to the chosen row $i$ and column $j$ is replaced by a value derived from the uniform distribution encompassing all permissible values for the column $j$ to produce a new dataset that is added to the population of candidate solutions. This alteration of the optimal solution produces a candidate dataset that diverges from the optimal dataset by only one entry.

**Crossover** The crossover operator uses the idea of genetic recombination to produce new candidate datasets by combining parameters of two parent candidates to produce a new offspring. In particular, in each generation $g$, a crossover operation in PRIVATE-GSD samples an elite dataset $\bar{D}$ from the elite set $\mathcal{E}_g$ and then combines the parameter of $\bar{D}$ with the parameters of the best dataset $\widehat{D}_g^\star$ to produce a new dataset $\widetilde{D}_{g,p}$.

To combine parameters, the crossover operation samples two rows $i_1, i_2$, and a feature $j$. Then the new dataset $\widetilde{D}_{g,p}$ has its $(i_1, j)$ entry set to the value of $\bar{D}$ at row $i_2$-th and feature $j$ (i.e., $\widetilde{D}(i_1, j) = \bar{D}(i_2, j)$) and all other entries of $\widetilde{D}$ equals $\widehat{D}_g^\star$.

# References


Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. *Discret. Comput. Geom.*, 53(3):650–673, 2015. doi: 10.1007/s00454-015-9678-x. URL https://doi.org/10.1007/s00454-015-9678-x.

David E Golberg. Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, 1989(102):36, 1989.

Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 705–714, 2010.

John H Holland. Genetic algorithms. *Scientific american*, 267(1):66–73, 1992.

Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: The sparse and approximate cases. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 351–360, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320290. doi: 10.1145/2488608.2488652. URL https://doi.org/10.1145/2488608.2488652.