Lessons learned from the NIST DP Synthetic Data Competition

Ryan McKenna [1] **Gerome Miklau** [1]

Abstract

The NIST *Differential Privacy Synthetic Data Competition* recently completed. In this paper we describe the winning solution of the final competition round, while highlighting lessons learned and open problems for the privacy community.

1. Introduction

The recent *Differential Privacy Synthetic Data Competition* (www.nist.gov, 2018) was designed by the National Institute of Standards and Technology (NIST) to encourage researchers to design algorithms for an important open problem in the privacy community: synthetic data generation under a formal privacy standard.

The goal of the competition was to advance the state-of-the art, generating new research ideas and algorithms that protect contributing users while releasing data in its original tabular form, preserving general-purpose utility.

The competition had three rounds of increasing complexity. The first author of this paper competed as team rmckenna, producing the top-ranked solution in the final round (as well as solutions ranked in the top five in the prior first two rounds). This paper focuses on the most challenging final competition round, describes the winning solution, as well as other top solutions, and lessons learned.

2. Competition setup

In each round, competitors had approximately one month to design their algorithm after the problem specification was released along with detailed materials in a competitor pack. The competitor pack included a provisional dataset and a corresponding file containing domain information. It also included a description of the "workload" — the statistics by which utility of the synthetic data would be evaluated.

Competitors could use this provisional dataset to develop and debug their algorithms. Preliminary submissions of generated synthetic data were scored according to a provisional version of the workload, on the provisional data, and a leaderboard was maintained throughout the one month competition period. After the final submission, scoring was performed on holdout versions of the workload and data and competitors were ranked by their score. While the provisional data was treated as public information for the purposes of the competition, solutions that overfit to the provisional dataset would not necessarily perform well under final scoring.

A team of experts unknown to the competitors checked the final submitted algorithms to ensure that they satisfied (ϵ, δ)-differential privacy (where the privacy parameters were supplied as input.) This was done by checking the written description of the algorithm and the source code submitted by the competitors.

Dataset In the third round of the competition, algorithms were evaluated on data from the 1940 U.S. decennial census. The provisional dataset consisted of data for one state (Colorado) and the final holdout data was for a different state (unknown at the time of the competition). This dataset contained 98 columns and about 661 thousand individuals. All data was treated as categorical, having integer values between 0 and $maxval$, where $maxval$ for each column was provided in a separate specs file. The $maxval$ was very large for some columns, e.g. for INCOME it was 10^7. The total number of possible database rows (i.e. the full multi-dimensional domain size) was enormous, at about 5×10^{205}.

Evaluation metrics Algorithms were evaluated in a high, medium, and low privacy regime. In particular, the privacy parameters were $\epsilon = 0.3, 1.0, 8.0$ and $\delta \approx 2 \cdot 10^{-12}$. The utility of the synthetic data was measured with respect to three main criteria:

1. Absolute error on 100 random 3-way marginals. These queries are of the form $P(A_1 = a_1, A_2 = a_2, A_3 = a_3)$, where A_1, A_2, A_3 are randomly chosen attributes and a_1, a_2, a_3 range over all the possible settings of those attributes.
2. Relative error on 300 random high-order conjunctions. These queries are of the form $P(A_1 \in S_1, \ldots, A_k \in$

[1]University of Massachusetts, Amherst, Massachusetts. Correspondence to: Ryan Mckenna <rmckenna@cs.umass.edu>.

S_k) where $A_1, \ldots, A_k$ is a simple random sample of the attributes in the data where each attribute has a 0.1 chance of being chosen, and S_i is a randomly chosen set of possible values for attribute A_i.

3. Income inequality and gender wage gap statistics broken down by city. These queries require computing: (i) the Gini index of Income for each city (which is a measure of income inequality relying on the full Income distribution); and (ii) the rank of cities by gender wage gap (which relied on average income by gender). We refer the reader to the competitor pack for the precise details of this evaluation criteria (www.nist.gov, 2018).

The first two evaluation criteria require the synthetic data to be accurate with respect to sets of linear queries, a class of queries well-studied in the differential privacy community (Zhang et al., 2018). However, we emphasize that the specific 3-way marginals and high-order conjunctions that were randomly selected were *not* known to competitors ahead of time, and they changed from the provisional scoring phase to the final scoring phase. The last evaluation criterion requires the synthetic data to be accurate with respect to non-linear queries, something that has received much less attention in the community.

The computational efficiency of submitted algorithm was not taken into consideration; several of the solutions (including the winning solution) required hours to run.

3. Algorithm Description

In this section we describe the main components of the winning solution, but leave the insights and reasoning behind design choices to the next section. The structure of the winning solution was straightforward and combined ideas from several recent papers (Abadi et al., 2016; Zhang et al., 2017; Chen et al., 2015; McKenna et al., 2019). At a high-level, the algorithm invokes the Gaussian mechanism to answer a carefully chosen subset of 1, 2, and 3-way marginals. Then the resulting noisy query answers are post-processed to obtain synthetic data that is most consistent with those marginals. The details are more involved, as there is an interplay between the different components of the mechanism. Thus, we describe a more comprehensive four step algorithm below.

Step 1: Preprocessing The first step of the algorithm is understanding and encoding the schema of the data. The competitor pack contained a "specs" file that listed every attribute and the maximum possible value for that attribute. Given this information, the set of possible values are $0, 1, \ldots maxval$. In addition to this specs file, the IPUMS website has additional documentation about the schema including descriptions of each attribute and a *list* of possible values for many attributes, which is often much smaller than it would be by just looking at $maxval$. This information about the domain is hardcoded within the algorithm so that it avoids introducing invalid rows to the synthetic data.

A couple attributes have a very large number of possible values, in which case they are discretized into a finer granularity. The $INCOME$ attribute additionally gets special treatment. In particular, values above 5000 are truncated to 5000 and the values are further decomposed into $INCOME_A$ and $INCOME_B$, where $INCOME = 100 \cdot INCOME_A + INCOME_B$. Thus, $INCOME_A$ is between 0 and 50 and $INCOME_B$ is between 0 and 99. Future measurements about $INCOME$ are taken with respect to $INCOME_A$ and/or $INCOME_B$. This is done to exploit periodic patterns in the income distribution and simultaneously coarsen the granularity to reduce the number of small counts.

Step 2: Measurement Selection

The second step collects queries for private measurement in Step 3. First, all 1-way marginals are added to a queue to be measured. Then, using the provisional (public) dataset, the algorithm identifies a set of 2 and 3 way marginals to measure using a *mutual information* criteria and adds them to the queue of queries to be measured. In particular, a complete graph is constructed where nodes are attributes in the data and edges are weighted by the mutual information between the two vertices in the edge. It then computes the *maximum spanning tree* of this graph, and adds a 2-way marginal to the queue for every edge in the tree. Then for every pair of connected edges (u, v) and (v, w) in the tree, it checks if w is statistically independent of u given v (up to a threshold), and if not, the (u, w) and (u, v, w) marginals are added to the queue as well.

Additionally, the weights of the graph are modified so that $INCOME$, SEX, and $CITY$ are guaranteed to be adjacent in the maximum spanning tree.

Step 3: Noise Calibration and Query Answering

The third step of the algorithm is to load the private data and invoke the Gaussian mechanism to answer the queries in the queue created in Step 2. Queries are answered in two stages, where each stage invokes the Gaussian mechanism with a function that has L_2 sensitivity 1. In the first stage, all 1-way marginals are answered, where each marginal is weighted equally and the L_2 norm of the weights is 1. The entries of these noisy marginals are truncated to 0 if the noisy count is less than 3σ, where σ is the standard deviation of the Gaussian noise added. The cells that are 0 after this step are treated as structural zeros in the rest of the algorithm, and the domain is appropriately compressed.

In the second stage, the 2 and 3 way marginals selected in

Step 2 are answered, but only those that have size less than 1 million after the domain compression step. As before, the weights of these marginals are normalized so that their L_2 norm is 1. The $(INCOME, SEX, CITY)$ marginal is given higher weight, but the specific value depends on ϵ and was chosen heuristically by trying different values on the provisional dataset. As before, these queries are answered using the Gaussian mechanism with standard deviation σ (same scale as before).

The minimum σ needed to achieve (ϵ, δ)-differential privacy is chosen using the moments accountant (Abadi et al., 2016).

Step 4: Post-processing

Given noisy answers to a set of queries defined by 1-, 2-, and 3-way marginals, the fourth step of the algorithm is to post-process the noisy query answers into synthetic data that is most consistent with those query answers. This is accomplished by estimating a graphical model representation of the data distribution from the noisy evidence and then transforming that into a tabular representation of the data distribution (McKenna et al., 2019). Finally, the synthetic data is transformed back to the original domain by reversing the transformations made in Step 1 of the algorithm.

4. Lessons Learned

The following describes the rational for the algorithm design decisions made above, along with lessons learned.

Lesson 1. Knowing the workload (i.e., the evaluation criteria) in advance is essential for designing good mechanisms. Trying to produce synthetic data that is accurate with respect to all possible downstream tasks will inevitably lead to sub-optimal accuracy on the actual downstream tasks that the analyst cares about. Thus, it is important for the algorithm and/or algorithm designer to be aware of the workload, and use that information implicitly or explicitly.

The workload used in this competition was particularly challenging due to its enormous size. Because the 100 random 3-way marginals and 300 random high-order conjunctions were not known ahead of time, they could not be provided as input to the mechanism. Essentially this meant the algorithm had to be designed to offer accuracy on *all* 3-way marginals and *all* high-order conjunctions. It would have been interesting to see how the top solutions would have changed if this information was available to the competitors or to the algorithm.

The third part of the workload, on income inequality and gender wage gap statistics, while more complicated and nonlinear, was actually much easier to handle because it was fixed and known in advance. Thus, the algorithm could be specifically designed to offer high accuracy on that task. To accomplish this, it suffices to provide accuracy on the $(Income, Sex, City)$ marginal, which is linear just like the rest of the workload and can be handled in a similar manner.

ϵ	Laplace	Gaussian
0.3	1155	383
1.0	346	116
8.0	43	15

Table 1: Standard deviation of Laplace and Gaussian noise required to obtain (ϵ, δ)-differential privacy.

Lesson 2. Gaussian noise is favorable to Laplace noise in the high-dimensional setting. The algorithm described in the previous section measures a total of 245 marginals. The L_1 sensitivty of this function is thus 245 and the L_2 sensitivity is $\sqrt{245} \approx 15.65$. Thus, the scale of Laplace noise required to achieve (ϵ, δ)-differential privacy is $\sqrt{2} \cdot 245/\epsilon \approx 346/\epsilon$. The scale of Gaussian noise required can be determined computationally using the moments accountant (Abadi et al., 2016). Table 1 compares the magnitude of noise required for each mechanism (in terms of standard deviation). On average, about $3\times$ less Gaussian noise is required than Laplace noise. Additionally, the moments accountant is flexible enough to handle *adaptive* composition, which is necessary for the two stages of measurements described above in Step 3 of the algorithm.

Lesson 3. The domain compression step improved efficiency and utility. It improved the efficiency of the post-processing step, whose complexity depends on the number of cells in the noisy marginals. As many of the counts in the one-way marginals were zero or very low, this shrunk the domain quite substantially. In fact, without this domain compression step, the post-processing would have required orders of magnitude more memory to run. As ϵ decreased the domain compression became more aggresive and efficiency improved. Additionally, by truncating low counts, the algorithm is trading a small amount of bias for a large amount of variance, and consequently improving utility.

Lesson 4. Having access to a provisional (public) dataset greatly simplified algorithm design. In particular, Step 2 of the algorithm used the provisional data to identify a good set of marginals to measure based on correlations between attributes. In principle, it is possible to use a differentially private version of that procedure on the true private data, but it is not obvious how to implement that because the sensitivity of mutual information is not easy to reason about and may be high. It is left as an interesting and important open problem for applying these ideas to settings when provisional data is not available and understanding the impact of provisional data that poorly approximates the final data.

Lesson 5. Selecting the right statistics to measure about the dataset is challenging. Measuring too many statistics

would require adding too much noise to those statistics to satisfy privacy. On the other hand, with too few statistics, the synthetic data would not be able to capture enough properties of the true data distribution. Additionally, measurements should only be taken if they improve accuracy on the statistics that matter — those in the workload. The mutual information criteria used in the algorithm draws inspiration from similar ideas that have been succesffully applied in the literature (Chen et al., 2015; Zhang et al., 2017). Another interesting problem is determining the right way to divide the privacy budget among the measurements. The algorithm equally divided the budget among the measurements (except for the special case of the (INCOME, SEX, CITY) marginal), but the problem of privacy budget allocation warrants further research.

Lesson 6. Generating synthetic data that matches the measured statistics is a hard problem. Using recently published ideas, this can be accomplished using graphical models (McKenna et al., 2019). The underlying estimation algorithm is capable of scaling to high-dimensional datasets, but the complexity depends on the structure of the measurements taken in a nuanced way. In particular, if the graphical model implied by the measured marginals has low tree-width (Wainwright & Jordan, 2008), then the estimation algorithm is able to scale. This insight influenced Step 2 of the algorithm, as the marginals selected are those along the edges of a (maximum spanning) tree.

5. Conclusions

Competitors were encouraged to make their solutions open source[1]. Many of the main ideas behind the winning solution were also used by the second-place solution, and hence there is overlap between the lessons learned from these two solutions. In particular, lessons 2, 3, and 5 are relevant to both solutions.

We believe the main differences that set the winning solution apart from the second-place solution are Step 2 (measurement selection) and Step 4 (post-processing). The winning solution measured a large number of overlapping marginals that formed a tree-like structure, whereas the second place solution measured marginals that formed a collection of disjoint clusters. The graphical-model estimation algorithm used in the winning solution was able to handle these complicated measurements on overlapping marginals, whereas the estimation algorithm used by the second-place solution could not effectively handle overlapping marginals. Hence, we believe the algorithm designers were more constrained in the set of marginals that could be measured while maintaining feasible estimation.

Overall, the competition was successful in creating a competitive environment for researchers to design novel and practical solutions to the challenging but important problem of generating synthetic data with differential privacy. Interesting open problems remain, such as how to design algorithms like this in the absence of provisional data. In addition, the winning solution could be further improved by measuring more marginals and/or intelligently splitting the privacy budget among the selected marginals.

[1] https://github.com/usnistgov/PrivacyEngCollabSpace

References

Abadi, Martin, Chu, Andy, Goodfellow, Ian, McMahan, H Brendan, Mironov, Ilya, Talwar, Kunal, and Zhang, Li. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318. ACM, 2016.

Chen, Rui, Xiao, Qian, Zhang, Yu, and Xu, Jianliang. Differentially private high-dimensional data publication via sampling-based inference. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 129–138. ACM, 2015.

McKenna, Ryan, Sheldon, Daniel, and Miklau, Gerome. Graphical-model based estimation and inference for differential privacy. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Wainwright, Martin J. and Jordan, Michael I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2): 1–305, 2008.

www.nist.gov. 2018 differential privacy synthetic data challenge, 2018. URL https://www.nist.gov/communications-technology-laboratory/pscr/funding-opportunities/open-innovation-prize-challenges-1.

Zhang, Dan, McKenna, Ryan, Kotsogiannis, Ios, Hay, Michael, Machanavajjhala, Ashwin, and Miklau, Gerome. Ektelo: A framework for defining differentially-private computations. In *Conference on Management of Data (SIGMOD)*, 2018.

Zhang, Jun, Cormode, Graham, Procopiuc, Cecilia M, Srivastava, Divesh, and Xiao, Xiaokui. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):25, 2017.