Co-clustering for differentially private synthetic data generation

Tarek Benkhelif, Françoise Fessant, Fabrice Clérot, Guillaume Raschia

Co-clustering for differentially private synthetic data generation

Tarek Benkhelif[1,2], Françoise Fessant[1], Fabrice Clérot[1], and Guillaume Raschia[2]

[1] Orange Labs
2, avenue Pierre Marzin, F-22307 Lannion Cédex, France
{tarek.benkhelif,francoise.fessant,fabrice.clerot}@orange.com
[2] LS2N - Polytech Nantes
Rue Christian Pauc, BP50609, 44306 Nantes Cédex 3, France
guillaume.raschia@univ-nantes.fr

Abstract We propose a methodology to anonymize microdata (i.e. a table of n individuals described by d attributes). The goal is to be able to release an anonymized data table built from the original data while meeting the differential privacy requirements. The proposed solution combines co-clustering with synthetic data generation to produce anonymized data. First, a data independent partitioning on the domains is used to generate a perturbed multidimensional histogram; a multidimensional co-clustering is then performed on the noisy histogram resulting in a partitioning scheme. This differentially private co-clustering phase aims to form attribute values clusters and thus, limits the impact of the noise addition in the second phase. Finally, the obtained scheme is used to partition the original data in a differentially private fashion. Synthetic individuals can then be drawn from the partitions. We show through experiments that our solution outperforms existing approaches and we demonstrate that the produced synthetic data preserve sufficient information and can be used for several datamining tasks.

Keywords: differential privacy, co-clustering, synthetic individual data

1 Introduction

There is an increasingly social and economic demand for open data in order to improve planning, scientific research or market analysis. In particular, the public sector via its national statistical institutes, healthcare or transport authorities, is pushed to release as much information as possible for the sake of transparency. Private companies are also implicated in the valorization of their data through exchange or publication. Orange has recently made available to the scientific community several mobile communication datasets collected from its networks in Senegal and Ivory Coast as part of D4D challenges (Data for Development). These challenges have shown the potential added-value of analyzing such data for several application domains which address both development projects and improvement of public policies effectiveness [3]. This demand for publicly available

data motivated the research community to propose several privacy preserving data publishing solutions.

Problem statement. The literature about privacy preserving data publishing is mainly organized around two privacy concepts i) group anonymization techniques such as k-anonymity [13] and ii) random perturbation methods with in particular the concept of Differential Privacy (DP) [6]. K-anonymity seeks to prevent re-identification of records by making each record indistinguishable within a group of k or more records and allows the release of data in its original form. The notion of protection defended by DP is the strong guarantee that the presence or absence of an individual in a dataset will not significantly affect the result of aggregated statistics computed from this dataset. DP works by adding some controlled noise to the computed function. There are two models for differential privacy: the interactive model and the non-interactive model. A trusted third party collects data from data owners and make it available for data users. In the interactive model, the trusted party catches the queries sent by data users and outputs a sanitized response. In the non-interactive model, the trusted party publishes a protected version of the data. In this paper, we study the problem of differentially private data generation. We consider the non-interactive model and seek to release synthetic data, providing utility to the users while protecting the individuals represented in the data.

Contributions. We present an original differentially private approach that combines co-clustering, an unsupervised data mining analysis technique, and synthetic data generation. We summarize our contributions below.

- We study and implement a two-phase co-clustering based partitioning strategy for synthetic data generation.
- We experimentally evaluate the released data utility, by measuring the statistical properties preservation and the predictive performance of the synthetic data.
- We compare our approach with other existing differentially private data release algorithms.

The paper is organized as follows. Section 2 first identifies the most related efforts to our work, Sections 3 and 4 give the necessary background on differential privacy and co-clustering, in Section 5 the proposed approach is described. The utility of the produced synthetic datasets is evaluated in Section 6. The final section gathers some conclusions and future lines of research.

2 Related Work

There are many methods designed for learning specific models with differential privacy, but we briefly review here the most related approaches to our work, and we only focus on histogram and synthetic data generation.

The first propositions that started addressing the non-interactive data release while achieving differential privacy are based on histogram release. Dwork et al. [7] proposed a method that publishes differentially private histograms by adding a Laplacian random noise to each cell count of the original histogram, it is considered as a baseline strategy. Xu et al. [15] propose two approaches for the publication of differentially private histograms: NoiseFirst and StructureFirst. NoiseFirst is based on the baseline strategy: a Laplacian random noise is first added to each count as in [7]. It is followed by a post-optimization step in which the authors use a dynamic programming technique to build a new histogram by merging the noisy counts. StructureFirst consists in constructing an optimal histogram using the dynamic programming technique to determine the limits of the bins to be merged. The structure of this optimal histogram is then perturbed via an exponential mechanism. And finally the averages of the aggregated bins are perturbed using the Laplacian mechanism. The authors in [2] propose a method that uses a divisible hierarchical clustering scheme to compress histograms. The histogram bins belonging to the same cluster have similar counts, and hence can be approximated by their mean value. Finally, only the noisy cluster centers, which have a smaller sensitivity are released. All the mentioned contributions deal only with unidimensional and bidimensional histogram publication and are not adapted to the release of multidimensional data. The closest approach to our work is proposed in [14], first, a cell-based partitioning based on the domains is used to generate a fine-grained equi-width cell histogram. Then a synthetic dataset D_c is released based on the cell histogram. Second, a multidimensional partitioning based on kd-tree is performed on D_c to obtain uniform or close to uniform partitions. The resulted partitioning keys are used to partition the original database and obtain a noisy count for each of the partitions. Finally, given a user-issued query, an estimation component uses either the optimal histogram or both histograms to compute an answer of the query. Other differentially private data release solutions are based on synthetic data generation [10][16]. The proposed solution in [10] first probabilistically generalizes the raw data and then adds noise to guarantee differential privacy. Given a dataset D, the approach proposed in [16] constructs a Bayesian network N, that approximates the distribution of D using a set P of low dimensional marginals of D. After that, noise is injected into each marginal in P to ensure differential privacy, and then the noisy marginals and the Bayesian network are used to construct an approximation of the data distribution in D. Finally, tuples are sampled from the approximate distribution to construct a synthetic dataset that is released. Our work focuses on releasing synthetic data and complements the efforts of [14] and [16] in the way that we also study a differentially private aggregation of multidimensional marginals. As in [14], we use the co-clustering like a multidimensional partitioning that is data-aware. And, unlike the variance threshold used in [14] or the θ parameter that determines the degree of the Bayesian network in [16] our solution is parameter-free.

3 Preliminaries and Definitions

3.1 Differential Privacy

Definition 1 (ε-Differential Privacy [5])**.** A random algorithm $\mathcal{A}$ satisfies ε-differential privacy, if for any two datasets D_1 and D_2 that differ only in one tuple, and for any outcome O of $\mathcal{A}$, we have

$$Pr[\mathcal{A}(D_1) = O] \leq e^{\varepsilon} \times Pr[\mathcal{A}(D_2) = O], \tag{1}$$

where $Pr[.]$ denotes the probability of an event.

Laplace Mechanism. To achieve differential privacy, we use the Laplace mechanism that adds random noise to the response to a query. First, the true value of $f(D)$ is computed, where f is the query function and D the data set, then a *random* noise is added to $f(D)$ And the $\mathcal{A}(D) = f(D) + noise$ response is finally returned. The amplitude of the noise is chosen as a function of the biggest change that can cause one tuple on the output of the query function. This amount defined by Dwork is called sensitivity.

Definition 2 (L_1-sensitivity)**.** The L_1-sensitivity of $f : D \to \mathbb{R}^d$ is defined as

$$\Delta(f) = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|_1 \tag{2}$$

For any two datasets D_1 and D_2 that differ only in one tuple.

The density function of the Laplace distribution is defined as follows.

$$Lap(x|\mu, b) = \frac{1}{2b} \exp\left(\frac{-|x - \mu|}{b}\right) \tag{3}$$

Where μ is called the position parameter and $b > 0$ the scale parameter.

The use of a noise drawn from a Laplacian distribution, $noise = Lap(\Delta f/\varepsilon)$, with the position parameter = 0, and the scale parameter = $\Delta f/\varepsilon$ guarantees the ε-differential privacy [11].

Composition. For a sequence of differentially private mechanisms, the composition of the mechanisms guarantees privacy in the following way:

Definition 3 (Sequential composition [9])**.** For a sequence of n mechanisms $\mathcal{A}_1$,...,$\mathcal{A}_n$ where each $\mathcal{A}_i$ respects the ε_i-differential privacy, the sequence of the $\mathcal{A}_i$ mechanisms ensures the $(\sum_{i=1}^{n} \varepsilon_i)$-differential privacy.

Definition 4 (Parallel composition [9])**.** If D_i are disjoint sets of the original database and $\mathcal{A}_i$ is a mechanism that ensures the ε-differential privacy for each D_i, then the sequence of $\mathcal{A}_i$ ensures the ε-differential privacy.

3.2 Data model

We focus on microdata. Each record or row is a vector that represents an entity and the columns represent the entity's attributes. We suppose that all the d attributes are nominal or discretized. We use d-dimensional histogram or data cube, to represent the aggregate information of the data set. The records are the points in the d-dimensional data space. Each cell of a data cube represents the count of the data points corresponding to the multidimensional coordinates of the cell.

3.3 Utility metrics

Hellinger distance. In order to measure the utility of the produced data, we use the Hellinger distance between the distributions in the original data and our synthetic data. We considered the Kullback-Leibler divergence, but we found the Hellinger distance to be more robust given that multidimensional histograms are highly sparse.

Definition 5 (Hellinger distance)**.** The Hellinger distance between two discrete probability distributions $P = (p_1, ..., p_k)$ and $Q = (q_1, ..., q_k)$ is given by : $D_{Hellinger}(P, Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}(\sqrt{p_i} - \sqrt{q_i})^2}$.

Random range queries. We use random range queries as a utility measure of the synthetic data. We generate random count queries with random query predicates over all the attributes:

Select COUNT(*) From D Where $X_1 \in I_1$ and $X_2 \in I_2$ and ... and $X_d \in I_d$.

For each attribute X_i, I_i is a random interval generated from the domain of X_i. We use the relative error to measure the accuracy of a query q, where $A_{original}(q)$ denotes the true answer of q on the original data and $A_{perturbed}(q)$ is the noisy count computed when the synthetic data generated from a differentially private mechanism are used.

Definition 6 (Relative error)**.** $RelativeError(q) = \frac{|A_{perturbed}(q) - A_{original}(q)|}{A_{original}(q)}$

4 Co-clustering

Co-clustering is an unsupervised data mining analysis technique which aims to extract the existing underlying block structure in a data matrix [8]. The data studied in the co-clustering problems are of the same nature as the data processed by the clustering approaches: they are composed of m observations without label, described by several variables, denoted $\{X_1, X_2, ..., X_d\}$. These variables can be continuous or nominal, then taking a finite number of different

values. The values taken by the descriptive variables are partitioned in order to obtain new variables $\{X_1^M, X_2^M, ..., X_d^M\}$ that are called variables-partitions. The values of these new variables are the clusters obtained by the partitions of the values of the variables $\{X_1, X_2, ..., X_d\}$. Each of the X_i^M variables has $\{k_1, k_2, ..., k_d\}$ values which are groups of values if the variable is nominal and intervals if the variable is continuous. The MODL approach makes it possible to achieve a co-clustering on the values of d descriptive variables of the data, we will use this particular feature in our work.

4.1 MODL Co-clustering

We choose the MODL co-clustering [4] because: First, MODL is theoretically grounded and exploits an objective Bayesian approach [12] which turns the discretization problem into a task of model selection. The Bayes formula is applied by using a hierarchical and uniform prior distribution and leads to an analytical criterion which represents the probability of a model given the data. Then, this criterion is optimized in order to find the most probable model given the data. The number of intervals and their bounds are automatically chosen. Second, MODL is a nonparametric approach according to C. Robert [12]: the number of modeling parameters increases continuously with the number of training examples. Any joint distribution can be estimated, provided that enough examples are available.

Data grid models. The MODL co-clustering approach allows one to automatically estimate the joint density of several (numerical or categorial) variables, by using a data grid model [4]. A data grid model consists in partitioning each numerical variable into intervals, and each categorical variable into groups. The cross-product of the univariate partitions constitutes a data grid model, which can be interpreted as a nonparametric piecewise constant estimator of the joint density. A Bayesian approach selects the most probable model given the dataset, within a family of data grid models. In order to find the best M^* model (knowing the data D), the MODL co-clustering uses a Bayesian approach called Maximum A Posteriori (MAP). It explores the space of models by minimizing a Bayesian criterion, called *cost*, which makes a compromise between the robustness of the model and its precision:

$$cost(M) = -\log(P(M|D))\alpha - log(P(M) * P(D|M)) \tag{4}$$

The MODL co-clustering also builds a hierarchy of the parts of each dimension using an ascending agglomerative strategy, starting from M^*, the optimal grid result of the optimization procedure up to $M_\emptyset$, the Null model, the unicellular grid where no dimension is partitioned. The hierarchies are constructed by merging the parts that minimize the dissimilarity index $\Delta(c_1, c_2) = cost(M_{c_1 \cup c_2}) - cost(M)$, where c_1, c_2 are two parts of a partition of a dimension of the grid M and $M_{c_1 \cup c_2}$ the grid after fusion of c_1 and c_2. In this way the fusion of the parts minimizes the degradation of the cost criterion, and thus, minimizes the loss of information.

5 DPCocGen

We present our DPCocGen algorithm, a two-phase co-clustering based partitioning strategy for synthetic data generation. First, a data independent partitioning on the domains is used to generate a multidimensional histogram, the Laplace mechanism is used as in the baseline strategy [7] to perturb the histogram. Then, a multidimensional MODL co-clustering is performed on the noisy histogram. This first phase corresponds to a differentially private co-clustering (as shown in figure 1) and aims to produce a partitioning scheme. In the second phase, DPCocGen uses the partitioning scheme to partition the original data and computes a noisy count for each of the partitions (using Laplace mechanism). Finally, the noisy counts are used to draw synthetic individuals from each partition.

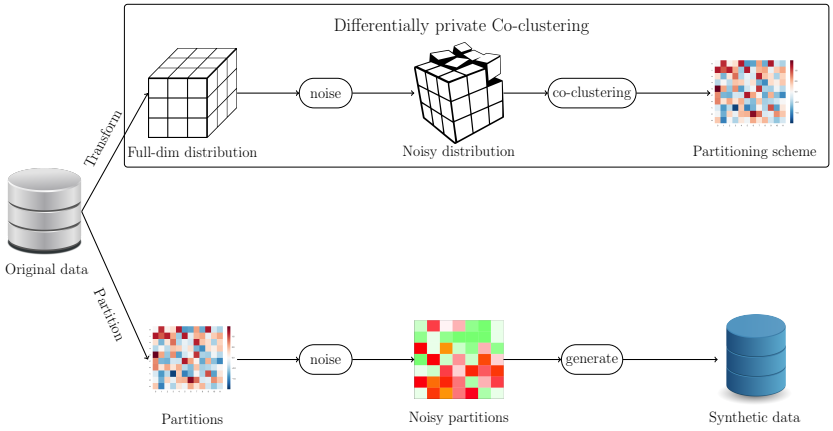


Figure 1: DPCocGen: a two-phase co-clustering based partitioning strategy for synthetic data generation.

The advantage of this approach lies in the fact that the partitioning scheme obtained through the co-clustering is indirectly dependent on the data structure, the intuition is that even after perturbing the multidimensional histogram, the co-clustering phase will preserve some of the relations between the clusters of attribute values (partitions) of the various dimensions. The resulting cell fusions limit the impact of the noise addition in the second phase. The original data is not consulted during the co-clustering construction which saves the privacy budget that is divided between the two phases to perturb the counts. The detailed steps of DPCocGen are given in Algorithm 1.

Algorithm 1 DPCocGen algorithm

Require: Dataset D, the overall privacy budget ε

1: **Phase 1** :
2: Build a multidimensional histogram from D.
3: **Perturb** the counts of each cell using a privacy budget ε_1.
4: Perform a multidimensional co-clustering from the histogram obtained in step 3.
5: **Phase 2** :
6: Partition the data set D based on the partitioning scheme obtained from step 4.
7: **Perturb** the aggregated counts of each partition returned form step 6 using a privacy budget $\varepsilon_2 = \varepsilon - \varepsilon_1$.
8: Generate synthetic individuals from each partition using the perturbed counts returned from step 7 to build a synthetic dataset D'.

Algorithm 2 Perturb algorithm

Require: Count c, privacy budget ε

1: $c' = c + Lap(1/\varepsilon)$
2: **if** $c' < 0$ **then**
3: $\quad c' = 0$
4: **end if**
5: Return c'

5.1 Privacy guarantee

DPCocGen follows the composability property of the differential privacy, the first and second phases require direct access to the database, Steps 3 and 7 of the Algorithm 1 are ε_1, ε_2-differentially private. No access to the original database is invoked during the sampling phase. The sequence is therefore ε-differentially private with $\varepsilon = \varepsilon_1 + \varepsilon_2$.

6 Experiments

In this section we conduct three experiments on a real-life microdata set in order to illustrate the efficiency of our proposition on a practical case. The objective is to explore the utility of synthetic data by measuring the statistical properties preservation, the relative error on a set of random range queries answers and their predictive performance.

6.1 Experimental Settings

Dataset. We experiment with the Adult database available from the UCI Machine Learning Repository[3] which contains 48882 records from the 1994 US census data. We retain the attributes {age, workclass, education, relationship, sex}. We discretize continuous attributes into data-independent equi-width partitions.

[3] https://archive.ics.uci.edu/ml/

Baseline. We implement the baseline strategy [7] to generate a synthetic dataset, a multidimensional histogram is computed and then disturbed through a differentially private mechanism. Records are then drawn from the noisy counts to form a data set.

PrivBayes. We use an implementation of PriveBayes [16] available at [1] in order to generate a synthetic dataset, we use $\theta = 4$ as suggested by the authors.

Privacy budget allocation. The privacy budget is equally divided between the two phases of DPCocGen for all the experiments, $\varepsilon_1 = \varepsilon_2 = \varepsilon/2$.

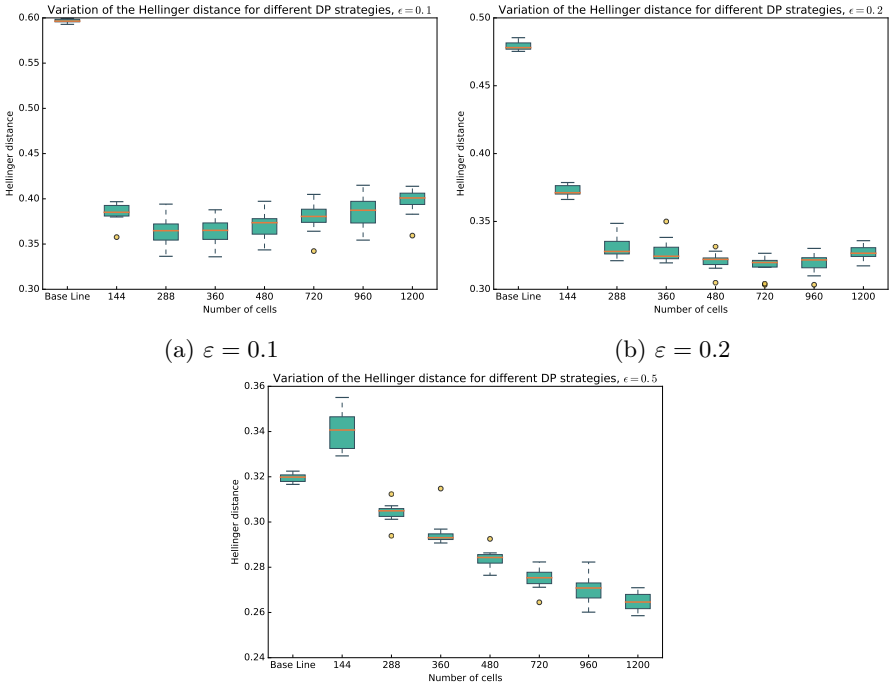
6.2 Descriptive performance

In this experiment, we are interested in preservation of the joint distribution of the original dataset in the generated synthetic data. In order to measure the difference between two probability distribution vectors we choose the Hellinger distance. First, we compute the multivariate distribution vector P of the original dataset, then, we compute the multivariate distribution vector Q of the synthetic data generated using *DPCocGen* and the multivariate distribution vector Q' of the synthetic data generated using *Base line*. Finally, the distances $D_{Hellinger}(P, Q)$ and $D_{Hellinger}(P, Q')$ are measured. For each configuration the distances are calculated through 50 synthetic data sets and represented in Figure 2. We use box-plots diagrams to represent these results where the x-axis represents the synthetic data generation method, the first box in the left represents the baseline strategy, the following boxes correspond to *DPCocGen* with different levels of granularity (number of cells). The y-axis indicates the Hellinger distance measured between the distribution calculated on the generated data and the original distribution.

Regardless of the privacy budget, the joint probability distribution of the synthetic data generated with *DPCocGen* is closer to the original distribution than the distribution of the data that is obtained using *Baseline*, except when $\varepsilon = 0.5$ and for the *DPCocGen* case with a high co-clustering aggregation level (144 cells), in that particular configuration the partitioning was too coarse and failed to correctly describe the data. The optimal aggregation level varies according to noise magnitude, but the finest aggregation level seems to offer a satisfying result for each configuration.

6.3 Random range queries

The goal of this experiment is to evaluate the utility of the produced data in terms of relative error when answering random range queries. We first generate 100 random queries. We produce synthetic datasets using *Base line*, *PrivBayes* and *DPCocGen*. We compute all the queries and report their average error over 15 runs. We use for this experiment the finer co-clustering level. Figure 3 shows that the average relative error decreases as the privacy budget ε grows for the



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.2$

(c) $\varepsilon = 0.5$

Figure 2: Joint distribution distances

three algorithms. One can also observe that *PrivBayes* and *DPCocGen* are close and do better than *Base line* regardless of the privacy budget.

6.4 Predictive performance

In this experiment we are interested in the classification performances obtained with a supervised classifier whose learning is based on synthetic datasets. We randomly select 80% of the observations in order to generate the synthetic data using *DPCocGen*, *Base line* and *PrivBayes*, we use the generated data to train a classifier in order to predict the value of the attributes *Sex* and *Relationship*. The remaining 20% are used for the evaluations. We use for this experiment the finer co-clustering level. The results are presented Figures 4 and 5, they represent the average on 50 runs. The privacy budget value is shown on the x-axis, the y-axis shows the area under the ROC curve (AUC) measured on the test set. The figure also indicates the performances obtained when the real data are used for learning the model (Original Data).

We retain that the classification performances obtained with *DPCocGen* are close to those obtained when the real data are used for learning the model. The

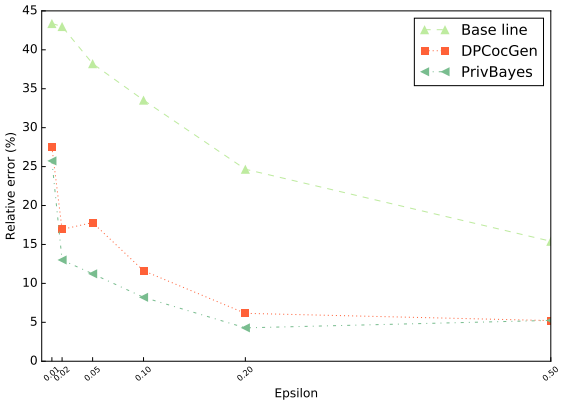


Figure 3: Random range queries

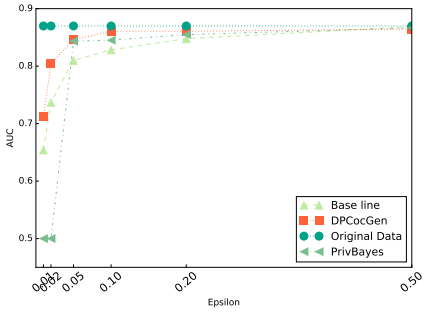


Figure 4: Sex prediction

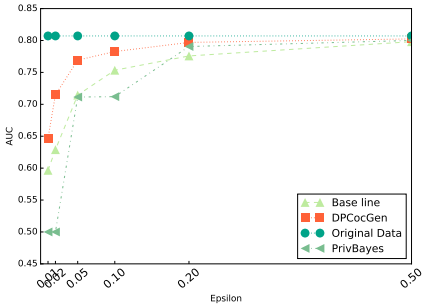


Figure 5: Relationship prediction

performances of *DPCocGen* are always higher than those of the *Base line* and *PrivBayes*.

7 Conclusion

This work presents an approach for the anonymization of microdata sets. The goal was to be able to produce synthetic data that preserve sufficient information to be used instead of the real data. Our approach involves combining differential privacy with synthetic data generation. We use co-clustering a data joint distribution estimation technique, in order to partition the data space in a differentially private manner. Then, we use the resulting partitions to generate synthetic individuals. We have shown that the synthetic data generated in this way retain the statistical properties of the raw data, thus using the synthetic data for various data mining tasks can be envisaged. We have also shown that our parameter-free approach outperforms other existing differentially private data

release algorithms. We now plan to compare our approach to a previous work, that is being published, which is based on a group anonymization technique and we aim to articulate the discussion around the utility/protection trade-off.

References

1. https://sourceforge.net/projects/privbayes
2. Acs, G., Castelluccia, C., Chen, R.: Differentially private histogram publishing through lossy compression. In: 2012 IEEE 12th International Conference on Data Mining. pp. 1–10. IEEE (2012)
3. Blondel, V.D., Esch, M., Chan, C., Clérot, F., Deville, P., Huens, E., Morlot, F., Smoreda, Z., Ziemlicki, C.: Data for development: the d4d challenge on mobile phone data. arXiv preprint arXiv:1210.0137 (2012)
4. Boullé, M.: Data grid models for preparation and modeling in supervised learning. Hands-On Pattern Recognition: Challenges in Machine Learning 1, 99–130 (2010)
5. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 4052, pp. 1–12. Springer Berlin Heidelberg (2006), http://dx.doi.org/10.1007/11787006_1
6. Dwork, C.: Differential privacy: A survey of results. In: International Conference on Theory and Applications of Models of Computation. pp. 1–19. Springer (2008)
7. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography Conference. pp. 265–284. Springer (2006)
8. Hartigan, J.A.: Direct clustering of a data matrix. Journal of the american statistical association 67(337), 123–129 (1972)
9. McSherry, F.D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. pp. 19–30. ACM (2009)
10. Mohammed, N., Chen, R., Fung, B., Yu, P.S.: Differentially private data release for data mining. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 493–501. ACM (2011)
11. Nissim, K., Raskhodnikova, S., Smith, A.: Smooth sensitivity and sampling in private data analysis. In: Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing. pp. 75–84. STOC '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1250790.1250803
12. Robert, C.: The Bayesian choice: from decision-theoretic foundations to computational implementation. Springer Science & Business Media (2007)
13. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(05), 557–570 (2002)
14. Xiao, Y., Xiong, L., Fan, L., Goryczka, S.: Dpcube: differentially private histogram release through multidimensional partitioning. arXiv preprint arXiv:1202.5358 (2012)
15. Xu, J., Zhang, Z., Xiao, X., Yang, Y., Yu, G., Winslett, M.: Differentially private histogram publication. The VLDB Journal 22(6), 797–822 (2013)
16. Zhang, J., Cormode, G., Procopiuc, C.M., Srivastava, D., Xiao, X.: Privbayes: Private data release via bayesian networks. In: Proceedings of the 2014 ACM SIGMOD international conference on Management of data. pp. 1423–1434. ACM (2014)