A comparative study of synthetic dataset generation techniques

Ashish Dandekar[1], Remmy A. M. Zen[1], Stéphane Bressan[1]

National University of Singapore, Singapore
(ashishdandekar, remmy)@u.nus.edu, steph@nus.edu.sg

Abstract. Unrestricted availability of the datasets is important for the researchers to evaluate their strategies to solve the research problems. While publicly releasing the datasets, it is equally important to protect the privacy of the respective data owners. Synthetic datasets that preserve the utility while protecting the privacy of the data owners stands as a midway.

There are two ways to synthetically generate the data. Firstly, one can generate a fully synthetic dataset by subsampling it from a synthetically generated population. This technique is known as fully synthetic dataset generation. Secondly, one can generate a partially synthetic dataset by synthesizing the values of sensitive attributes. This technique is known as partially synthetic dataset generation. The datasets generated by these two techniques vary in their utilities as well as in their risks of disclosure. We perform a comparative study of these techniques with the use of different dataset synthesisers such as linear regression, decision tree, random forest and neural network. We evaluate the effectiveness of these techniques towards the amounts of utility that they preserve and the risks of disclosure that they suffer.

Keywords: synthetic datasets, risk of disclosure, privacy, utility

1 Introduction

On one hand, the philosophy of open data dictates that if the valuable datasets are made publicly available, the problems can be crowdsourced in the expectation to obtain the best possible solution. On the other hand, business organizations have their concerns regarding the public release of the datasets which may lead to the breach of private and sensitive information of stakeholders. In order to mitigate the risk of confidentiality breach, agencies employ different techniques such as reordering or recoding of sensitive variables, shuffling values among different records. In spite of these efforts by agencies, we have examples of confidentiality breaches [19, 10] in anonymised datasets.

Fully synthetic datasets proposed by Rubin [17] and partially synthetic datasets proposed by Little [8] bridge the gap between utility and privacy. They use multiple imputation, a technique used for repopulating the missing values in a dataset, to generate synthetic records which preserve relationships in the population. Following up on these works of multiple imputation, Reiter et al. [1, 6, 13, 15] use different machine learning tools to generate synthetic datasets. These works treat values of synthetically generated attributes as missing values that are generated using models such as Decision Trees, Random Forest, Support Vector Machine, etc.

In this work, we comparatively evaluate fully synthetic dataset generation and partially synthetic dataset generation using different dataset synthesisers: namely linear regression, decision tree, random forest and neural network. We comparatively evaluate effectiveness, as utility preservation and risk of disclosure, and efficiency of the synthetic dataset generation techniques. Given the tradeoff between the efficiency and effectiveness, we observe that decision yreet are not only efficient but also competitively effective compared to other dataset synthesisers.

2 Related Work

Synthetic dataset generation work stems from the early works of data imputation to fill in the missing values in the surveys [16]. In [17], Rubin proposes a procedure to generate fully synthetic dataset that uses multiple imputation technique to synthetically generates values for a set of attributes for all datapoints in the dataset. Although it is advantageous to synthetically generate values for all datapoints, it is not always a necessity. Partially synthetic datasets, proposed by Little [8], are generated by synthetically generating the values of the attributes that are sensitive to public disclosure. Various dataset synthesisers such as decision tree [1, 2, 15] have been used to generate fully and partially synthetic datasets. Drechsler et al. [6] have performed an empirical comparative study between different dataset synthesisers. Comparison between fully and partially synthetic datasets can be found in [4]. Recently, Nowok et al. [11] have created an R package, *synthpop*, which provides basic functionalities to generate synthetic datasets and perform statistical evaluation.

The effectiveness of the synthetic dataset lies in the amount of utility it retains from the original dataset. Most of the works [1, 6, 13, 15] use statistical methods of estimation for the evaluation of utility. They use estimators of mean and variance to calculate confidence intervals. Regression analyses are used to test whether the relationships among different variables are preserved. Aside from these analysis specific measures, Woo et al. [20] and Karr et al. [7] have proposed global measures such as Kullback-Leibler (KL) divergence, extension of propensity score, cluster analysis measure.

One of the prime motivations behind publicly releasing synthetic dataset instead of original datasets is to maintain the privacy of the data owners. In [14], Reiter introduces formalism to calculate risk of disclosure in synthetically generated datasets using multiple imputation. The same formalism has been used in [6, 15] to evaluate the risk of disclosure. For further details, readers are requested to refer to [3].

In this work, we comparatively evaluate efficiency and effectiveness of fully and partially synthetic dataset generation techniques using different dataset synthesisers including neural networks.

3 Synthetic dataset Generation using Multiple Imputation

3.1 Multiple Imputation

Consider a dataset of size n sampled from a population of size N. Let Y_{nobs} denote subset of attributes in the dataset whose values are either missing for some datapoints or sensitive towards the public disclosure. Rubin [16] proposes to synthetically generate values for Y_{nobs} given the knowledge of rest of the attributes in the dataset, say Y_{obs}.

Let, $\mathcal{M}$ be a dataset synthesiser that generates values for an attribute Y_i given the information about rest of the attributes, denoted as Y_{-i}. With the help of $\mathcal{M}$, an imputer independently synthesises values of Y_{nobs} m times and releases m synthetic datasets $\mathcal{D} = \{\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^m\}$. In order to synthesise multiple sensitive attributes, we follow the procedure presented in [2]. It defines an order on the attributes that are to be synthesised. Values of the first attribute are synthesised by training the dataset synthesiser on the original dataset. For any later attributes, the dataset synthesiser is trained on the dataset with synthetic values of the attributes preceding it. Interested readers can refer to [15] for a detailed discussion on choosing the order of synthesis.

The reason behind releasing m different datasets and combining estimators on each dataset is two folds. Firstly, there is randomness in the dataset due to sampling from the population. Secondly, there is randomness in the dataset due to imputed values. In order two capture these variabilities, framework of multiple imputation proposes the release of m datasets.

3.2 Fully Synthetic Dataset Generation

Consider a dataset of size n sampled from a population of size N. Suppose that an imputer knows the values of a set of variables X for the entire population and values for rest of the variables, Y, only for a selected small sample. Let, Y_{inc} and Y_{exc} denote values of variables which are included in the sample and excluded from the sample respectively. The imputer synthetically generates values of Y_{exc} using a dataset synthesizer $\mathcal{M}$ trained on Y_{inc} and X. This synthesis is equivalent to performing multiple imputation with Y_{exc} as Y_{nobs} and Y_{inc} as Y_{obs}. Publicly released datasets, $\mathcal{D}$, comprise of m samples selected synthetically generated population. In order to statistically estimate an attribute Q, we use the estimators of mean and variance presented in [13].

Theoretically, fully synthetic datasets provides 100% guarantee against the disclosure of value of sensitive attribute [17]. Since $n << N$, it is less probable to have record from the original sample in the final dataset. Final datasets are sampled from synthetic population datasets in which $N - n$ records are synthetically generated.

3.3 Partially Synthetic Dataset Generation

Let S be a dataset of size n sampled from a population of size N. In order to protect the sensitive information, an imputer decides to alter values of a set of attributes, Y, for a subset of datapoints in S. Let Z be a binary vector of size n. Z_i takes value one if Y values of the i^{th} datapoint are to be synthetically generated and Z_i takes value zero if values of Y attributes are not be altered.

Let $Y_{syn} = \{Y_i | \forall i, Z_i = 1\}$ and $Y_{org} = \{Y_i | \forall i Z_i = 0\}$. We generate m partially synthetic datasets by multiple imputation. In this case, Y_{syn} are the datapoints, with missing values, that we synthetically generate by training a dataset synthesiser on the available data, i.e Y_{org}. This synthesis is equivalent to performing multiple imputation with Y_{syn} as Y_{nobs} and Y_{org} as Y_{obs}. Publicly released datasets, $\mathcal{D}$, comprise of m datasets sampled from the population wherein values of attributes in Y are synthetically generated, as specified by Z, for each of the dataset. In order to statistically estimate an attribute Q, we use the estimators of mean and variance presented in [12].

3.4 Dataset synthesisers

Now, we discuss different dataset synthesisers namely linear regression, decision tree, random forest and neural network.

We use **linear regression** [9] as a baseline dataset synthesiser. In order to generate synthetic data, for every dataset and for every attribute Y_i to be synthetically generated, we learn the parameters of the regression model using the dataset with attributes in Y_{-i}. We generate values by sampling from a Gaussian distribution with a constant variance and the mean as determined parameters of regression.

We use the technique, proposed by Reiter [15], that uses **classification and regression tree** [9] to generate partially synthetic datasets. The procedure starts with building a decision tree using the values of the attributes that are available in the dataset Y_{-i}. In order to synthesise the value of an attribute Y_i for a datapoint j, we trace down the tree using the known attributes of j until we reach the leaf node. Let L_j be the set of values of Y_i in the leaf node. For a categorical attribute Y_i, Reiter proposes Bayesian bootstrap sampling to choose m different values. For a continuous attribute Y_i, we fit a kernel density estimator over the values in L_j and sample m values from the estimate.

We use the technique, proposed by Caiola et al. [1], that uses **random forest** [9] to generate partially synthetic datasets. In order to synthesise values for a certain attribute Y_i, they train a fixed number decision trees on random samples of training dataset Y_{-i}. For a categorical attribute, the collection of results from constituent decision tree forms a multinomial distribution. m values are sampled from this distribution as the synthetic values for Y_i. For a continuous attribute, they propose use of a kernel density estimator over the results from decision trees and sample values from the estimator.

We use **Neural network** [9] that learns an abstract function mapping an input to the corresponding output as a data synthesiser. If we consider K-class classification problem, the output layer of a neural network comprises of K nodes, with each node representing the probability of the respective class being the output of the model. We treat every attribute as a categorical variable. In order to synthesise value of an attribute Y_i, we train a neural network using features in Y_{-i}. We sample a value for attribute Y_i using the output layer as a multinomial distribution.

4 Empirical Evaluation

4.1 Dataset and Experimental Setup

We conduct experiments on a microdata sample of US Census in 2000 provided by IPUMS International [18]. Following the approach presented in [6] to consider the records of the heads of households, we consider the records of 316,276 heads of households as the population.

All programs are run on Linux machine with quad core 2.40GHz Intel® Core i7™processor with 8GB memory. The machine is equipped with two Nvidia GTX 1080 GPUs. Python® 2.7.6 is used as the scripting language.

4.2 Evaluation of utility

The utility of generated dataset needs to be evaluated at two different levels. Firstly, we need to evaluate differences between the distribution of values of original attribute and synthetically generated attributes. Secondly, we need to evaluate the difference between the quality of a statistical estimator for an attribute on synthetic dataset and original data.

In order to evaluate the first level, we calculate the similarity between the overall distribution of values of an attribute by calculating **normalised KL-divergence** between the distribution of values of the attribute in population and the distribution of the same attribute in synthetically generated dataset. In order to evaluate the second level, we use the **overlap** [7] between 95% confidence intervals of an statistical estimator that are obtained using original dataset and synthetically generated dataset. If the intervals are similar to each other, synthetic dataset generation procedure preserves the utility. Therefore, maximum extent of overlap, which is 1, implies preservation of the utility.

4.3 Evaluation of risk of disclosure

We follow the procedure in Reiter [5, 13] to estimate the **risk of disclosure** in the synthetically generated dataset. We assume that the intruder has complete information about an auxiliary variable, say region of birth, which is not a sensitive variable. Let $\mathbf{t}$ be a vector of information possessed by an intruder. For every datapoint j in the dataset, the intruder calculates the probability of the datapoint j being the record of interest.

The intruder selects datapoints with maximum probability value. This process is repeated for every target datapoint in $\mathbf{t}$. In order to evaluate the risk of disclosure, we calculate *true match rate* and *false match rate* as defined in [5, 13]. Smaller the true match rate, better is the performance of a dataset synthesiser.

4.4 Evaluation

The process starts by drawing 1% sample from the population, which we treat as the original dataset. We synthetically generate values for two attributes: income and age, in the same order. We generate 5 synthetic datasets for each original dataset. We repeat this procedure for 500 original datasets and mean of various metrics over 500 iterations is reported. In order to generate partially synthetic datasets, we need to define the cutoffs for the values of attribute that determine quantify the sensitivity of the attribute towards disclosure. We consider data-points that have more than 70000$ income value and less than 26 age value to be the ones with sensitive information.

Utility evaluation results for the *age* attribute for partially synthetic datasets and fully synthetic datasets are presented in Table 1 and Table 2 respectively. We observe that although two techniques show comparable values of synthetic means, the technique of partially synthetic dataset generation shows greater extent of the overlap Partially synthetic dataset generation does not replace all values of the attributes in the sample. Therefore, we observe higher overlap for partially synthetic datasets. We also observe a large deviation in the sample mean of from its original mean in case of linear regression. Linear regression in the absence of any regularization suffers from overfitting [9]. Due to the order of synthesis, linear regression model is fit on the synthetically generated values of *income* while synthesising value for *age*. Thus, it overfits the synthetic data and fails to capture exact distribution of values in the original dataset. Decision tree and other models are not prone to overfitting the training dataset and hence do not show such a degradation in utility. We also conduct similar evaluation for the *income* attribute that we present in the long version of this paper [].

Feature	**Data synthesisers**	**Original Sample Mean**	**Partially Synthetic Data**		
			Synthetic Mean	**Overlap**	**Norm KL Div.**
Age	Linear Regression	49.83	24.69	0.50	0.55
	Decision Tree	49.83	49.83	0.90	0.56
	Random Forest	49.82	49.74	0.95	0.56
	Neural Network	49.87	49.78	0.90	0.56

Table 1. Evaluation of utility for partially synthetic datasets generated using different dataset synthesisers.

Feature	**Data synthesisers**	**Original Sample Mean**	**Fully Synthetic Data**		
			Synthetic Mean	**Overlap**	**Norm KL Div.**
Age	Linear Regression	49.83	-192.21	0.50	0.56
	Decision Tree	49.83	49.83	0.56	0.56
	Random Forest	49.82	46.25	0.68	0.57
	Neural Network	49.76	54.32	0.75	0.99

Table 2. Evaluation of utility for fully synthetic datasets generate using different dataset synthesisers.

In order to evaluate the risk of disclosure, we require a scenario. We assume that an intruder is interested in people who are born in US and have income more than $250,000$$. All these people are the targets of the intruder. Intruder tries to match every single target with the records in the released datasets. We consider two records perfectly match if the people representing the records are born in US, they have income more than $250,000$$ and the age of the person in dataset in within the tolerance of 2 compared to target person.

Two cases arise in the evaluation. For a given target, the intruder may or may not know if the target person is included in the released sample. We observe that, in the case when the intruder does not have certainty about inclusion of target in the sample, risk of disclosure is the least. In most of the cases, the targets might not be present in the released sample which leads to true match rate of 0. Observing the results for the case when a target is present in the sample, we see that neural networks comparatively offer better performance than rest of the dataset synthesisers.

Dataset synthesisers	**Target is in the sample**		**Target may be in the sample**	
	True MR	**False MR**	**True MR**	**False MR**
Linear Regression	0.06	0.82	0.00	0.00
Decision Tree	0.18	0.68	0.00	0.99
Random Forest	0.35	0.50	0.00	0.99
Neural Network	0.03	0.92	0.00	0.99

Table 3. Evaluation of risk of disclosure for different dataset synthesisers

We present the results of comparative efficiency of both these techniques using different dataset synthesisers in Table 4. We observe that the neural networks achieve the low risk of disclosure at the cost of a higher running time than the time taken by linear regression or decision trees.

dataset synthesiser	**Partially Synthetic Dataset Generation**	**Fully Synthetic Dataset Generation**
Linear Regression	0.040	0.068
Decision Tree	0.048	0.533
Random Forest	3.350	103.543
Neural Network	0.510	55.26

Table 4. Efficiency: Each cell shows the running time required, in seconds, to generate 5 synthetic datasets.

5 Conclusion and Future Works

In this work, we comparatively evaluate fully and partially synthetic dataset generation techniques using different dataset synthesisers, namely linear regression, decision tree, random forest and neural network. WE comparatively evaluate effectiveness, in terms of utility preservation and risk of disclosure, and efficiency of these techniques. The analysis shows that decision trees stand as a good dataset synthesiser given its high effectiveness compared to other data synthesisers. This observation agrees with the result in [6].

We use a well-structured dataset in this work. Many real-world datasets do not have a well defined structure. For instance, the social network datasets or the datasets generated from the readings collected by sensors. As a future work, we want to explore how synthetic dataset generation techniques can be adopted for such non-structured or semi-structured datasets.

References

1. Caiola, G., Reiter, J.P.: Random forests for generating partially synthetic, categorical data. Trans. Data Privacy 3(1), 27–42 (2010)
2. Drechsler, J.: Using support vector machines for generating synthetic datasets. In: Privacy in Statistical Databases. pp. 148–161. No. 6344, Springer (2010)
3. Drechsler, J.: Synthetic datasets for statistical disclosure control: theory and implementation, vol. 201. Springer Science & Business Media (2011)
4. Drechsler, J., Bender, S., Rässler, S.: Comparing fully and partially synthetic datasets for statistical disclosure control in the german iab establishment panel. Trans. Data Privacy 1(3), 105–130 (2008)
5. Drechsler, J., Reiter, J.P.: Accounting for intruder uncertainty due to sampling when estimating identification disclosure risks in partially synthetic data. In: International Conference on Privacy in Statistical Databases. pp. 227–238. Springer (2008)
6. Drechsler, J., Reiter, J.P.: An empirical evaluation of easily implemented, nonparametric methods for generating synthetic datasets. Computational Statistics & Data Analysis 55(12), 3232–3243 (2011)
7. Karr, A.F., Kohnen, C.N., Oganian, A., Reiter, J.P., Sanil, A.P.: A framework for evaluating the utility of data altered to protect confidentiality. The American Statistician 60(3), 224–232 (2006)
8. Little, R.J.: Statistical analysis of masked data. Journal of Official statistics 9(2), 407 (1993)
9. Murphy, K.P.: Machine learning: a probabilistic perspective. MIT press (2012)
10. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: Security and Privacy, 2008. SP 2008. IEEE Symposium on. pp. 111–125. IEEE (2008)
11. Nowok, B., Raab, G., Dibben, C.: synthpop: Bespoke creation of synthetic data in r. Journal of Statistical Software, Articles 74(11), 1–26 (2016)
12. Raghunathan, T.E., Reiter, J.P., Rubin, D.B.: Multiple imputation for statistical disclosure limitation. Journal of official statistics 19(1), 1 (2003)
13. Reiter, J.P.: Inference for partially synthetic, public use microdata sets. Survey Methodology 29(2), 181–188 (2003)
14. Reiter, J.P.: Estimating risks of identification disclosure in microdata. Journal of the American Statistical Association 100(472), 1103–1112 (2005)
15. Reiter, J.P.: Using cart to generate partially synthetic public use microdata. Journal of Official Statistics 21(3), 441 (2005)
16. Rubin, D.B.: Basic ideas of multiple imputation for nonresponse. Survey Methodology 12(1), 37–47 (1986)
17. Rubin, D.B.: Discussion statistical disclosure limitation. Journal of official Statistics 9(2), 461 (1993)
18. Ruggles, S., Genadek, K., Goeken, R., Grover, J., Sobek, M.: Integrated public use microdata series: Version 6.0 [dataset] (2015), http://doi.org/10.18128/D010.V6.0
19. Sweeney, L.: Computational disclosure control for medical microdata: the datafly system. In: Record Linkage Techniques 1997: Proceedings of an International Workshop and Exposition. pp. 442–453 (1997)
20. Woo, M.J., Reiter, J.P., Oganian, A., Karr, A.F.: Global measures of data utility for microdata masked for disclosure limitation. Journal of Privacy and Confidentiality 1(1), 7 (2009)