Data Anonymization for Open Science

useR! 2024

Jiří Novák[1,2] Oscar Thees[2,5] Marko Miletic[3] Alžběta Beranová[4]

[1]University of Zurich [2]University of Applied Sciences Northwestern Switzerland

[3]Bern University of Applied Sciences [4]Czech Statistical Office [5]TU Wien

July 8, 2024

Jiří Novák, Oscar Thees, Marko Miletic

Table of Content

1. General methodological overview
2. Disclosure control methods

    A. Non-perturbation methods

    B. Perturbation methods

    C. Synthetic methods

    - Generating synthetic data using synthpop
    - Generating synthetic data using simPop
    - Generating synthetic data with GANs

Data Anonymization in which context?

This tutorial is about Data Anonymization in the context of the field of **Statistical Disclosure Control** (SDC).
SDC is also known as Statistical disclosure limitation or Disclosure avoidance.

Statistical Disclosure Control seeks to protect statistical data in such a way that they can be released without giving away confidential information that can be linked to specific individuals or entities.

Importance of Data Anonymization

There are several main reasons:

1. **Principle** It is a fundamental principle of Official Statistics that the statistical records of individual persons, businesses, or events used to produce Official Statistics are strictly confidential and to be used only for statistical purposes.
2. **Legal** Legislation imposes a legal obligation to protect individual business and personal data. Legal frameworks regulate what is allowed and what is not allowed regarding the publication of private information.
3. **Quality** Respondents need confidence in the preservation of the confidentiality of individual information. If they do not trust the confidentiality of the data, they may not provide accurate information.
4. **Ethical** Disclosing information that can be linked to specific individuals or entities is unethical.

Relevance to Open Science

Open Science, **Open Access**, **Open Data** are important trends in the scientific community.
Research data that results from publicly funded research should be **FAIR**:
findable, **accessible**, **interoperable**, **reusable**

- therefore replicable, transparent, trustworthy, verifiable and accountable
- Principle: **As open as possible, as closed as necessary**
- Enables data sharing and collaboration
- Facilitates reproducible research
- Balances transparency with privacy

Commission Recommendation (EU) 2018/790 on access to and preservation of scientific information

Outputs to protect

Different outputs require different approaches to SDC and different mixtures of tools.

- **Macrodata** (Tabular data)
- **Microdata**
- **Dynamic databases**
- **Statistical analyses**

Disclaimer: Imposing a single solution for all types of data is not possible.
This tutorial will focus on **Microdata** and **Tabular data**.

Key Concepts

Key Concepts are:

- **Disclosure (Re-identification)**
  - A disclosure occurs when a person or an organisation recognises or learns something that they did not know already about another person or organisation, via released data.

Key Concepts

Key Concepts are:

- **Disclosure (Re-identification)**
  - A disclosure occurs when a person or an organisation recognises or learns something that they did not know already about another person or organisation, via released data.
- **Re-identification risk**
  - Re-identification risk is the risk that an intruder can link a record in the released data to a specific individual in the population.

Key Concepts

Key Concepts are:

- **Disclosure (Re-identification)**
  - A disclosure occurs when a person or an organisation recognises or learns something that they did not know already about another person or organisation, via released data.
- **Re-identification risk**
  - Re-identification risk is the risk that an intruder can link a record in the released data to a specific individual in the population.
- **Data utility**
  - Data utility is the usefulness of the data for the intended purpose.

Disclosure

A disclosure occurs when a person or an organisation recognises or learns something that they did not know already about another person or organisation, via released data.

Types of disclosure risk:

(1) **Identity disclosure** Revealing the identity of an individual.

(2) **Attribute disclosure** Revealing sensitive attributes of an individual.

(3) **Inferential disclosure** Making inferences about an individual based on the released data.

Disclosure

Types of disclosure risk:

(1) **Identity disclosure**

Residency	Age	Sex	Occupation
Salzburg	50	Male	Professor

(2) **Attribute disclosure**

Group	Males	Females	Total
Football fans	22	0	22
Non Football fan	93	85	178
Total	115	85	200

Risk and utility

SDC seeks to optimise the trade-off between the disclosure risk and the utility of the protected released data.

- **Risk**: the probability of a disclosure event occurring.
- **Utility**: the usefulness of the data for the intended purpose.

The goal is to find a balance between risk and utility, so there is a **risk-utility trade-off**.

Risk-utility trade-off

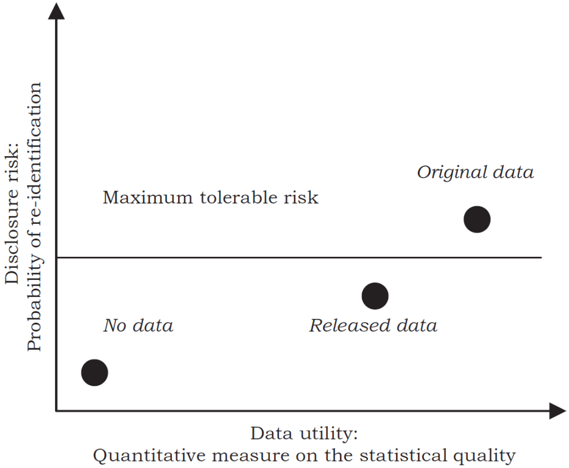


Figure: R-U confidentiality map (Duncan et al.,2001)

Disclosure risk

A unit is at risk of disclosure when it cannot be confused with several other units in the data set.

- **k-anonymity** A data set is said to satisfy k-anonymity for k $>$ 1 if, for each combination of values of quasi-identifiers (e.g. name, address, age, gender, etc.), at least k records exist in the data set sharing that combination.
- Ensures that each record is indistinguishable from at least k-1 other records with respect to the quasi-identifiers.

More robust approaches:

- **l-Diversity** Extends k-anonymity by ensuring that the sensitive attribute has at least l well-represented values
- **t-Closeness** Ensures that the distribution of the sensitive attribute in any equivalence class is close to the distribution in the entire dataset

Attacker Scenarios

Data Intruder: An attacker who tries to re-identify individuals using the anonymized dataset and auxiliary information. For example, consider a dataset of anonymized medical records. An intruder might use publicly available information, like voter registration lists, to link unique combinations of quasi-identifiers (such as age, gender, and zip code) to re-identify individuals.
Data Linkage: This scenario involves an attacker who combines multiple datasets to enhance the chances of re-identification. For example, if an attacker has access to an anonymized dataset from a hospital and another dataset from a social media platform, they might link these datasets through common quasi-identifiers to re-identify patients.
Background Knowledge: An attacker might use their own knowledge about certain individuals to identify them in a dataset. For example, if someone knows a particular person's age, job title, and city, they might find a matching record in an anonymized employment dataset, thereby re-identifying that individual.

Variables

1. **Identifiers** - variables that can directly identify an individual
2. **Quasi-identifiers** or **key variables** - these variables don't identify individuals on their own but can do so when combined with other quasi-identifiers
3. **Confidential outcome variables** - variables that contain sensitive information that should be protected
4. **Non-confidential outcome variables** - these are variables that are not sensitive and don't risk the privacy of individuals if disclosed

Disclosure control methods

1. **Masking original data**
    1. **Non-perturbative masking** - Methods that alter data to hide identities without changing its actual values
    2. **Perturbative masking** - Methods that add noise or alter data values to prevent identification
2. **Generating synthetic data**
    1. **Parametric methods** - Techniques that use statistical models based on the data's distribution to generate synthetic data.
    2. **Non-parametric methods** Techniques that do not assume an underlying distribution, using methods like bootstrapping to generate synthetic data.
    3. **Generative Adversarial Networks (GANs)** Advanced machine learning models that generate highly realistic synthetic data by training two neural networks in tandem.

Which method to use?

The SDC process depends:

1. Type of data
   - Full population vs Sample data
2. Meta information
   - Sampling design
   - Response, coverage
3. Type of variables
   - Categorical vs. Continuous
4. Type of (required) output
   - Microdata files vs tabular data

Packages for SDC - Microdata (Unit-level data)

sdcMicro can be used to anonymize data, i.e. to create anonymized files for public and scientific use. It implements a wide range of methods for anonymizing categorical and continuous (key) variables. The package also contains a graphical user interface, which is available by calling the function sdcGUI.

synthpop using regression tree methods to simulate synthetic data from given data. It is suitable to produce synthetic data when the data have no hierarchical and cluster information (such as households) as well as when the data does not collected with a complex sampling design.

simPop using linear and robust regression methods, random forests (and many more methods) to simulate synthetic data from given complex data. It is also suitable to produce synthetic data when the data have hierarchical and cluster information (such as persons in households) as well as when the data had been collected with a complex sampling design. It makes use of parallel computing internally.

Packages for SDC - Tabular data (Aggregated data)

sdcTable can be used to provide confidential (hierarchical) tabular data. It includes the HITAS and the HYPERCUBE technique and uses linear programming packages (Rglpk and lpSolveAPI) for solving (a large amount of) linear programs.

sdcSpatial can be used to smooth or/and suppress raster cells in a map. This is useful when plotting raster-based counts on a map. sdcHierarchies provides methods to generate, modify, import and convert nested hierarchies that are often used when defining inputs for statistical disclosure control methods.

SmallCountRounding can be used to protect frequency tables by rounding necessary inner cells so that cross-classifications to be published are safe.

GaussSuppression can be used to protect tables by suppression using the Gaussian elimination secondary suppression algorithm.

Non-perturbation methods

Non-perturbative masking does not rely on distortion of the original data but on partial suppressions or reductions of detail.

Method	Continuous data	Categorical data
Sampling		X
Global recoding	X	X
Top and bottom coding	X	X
Local suppression		X

Table: Non-perturbative methods vs. data types

sdcMicro

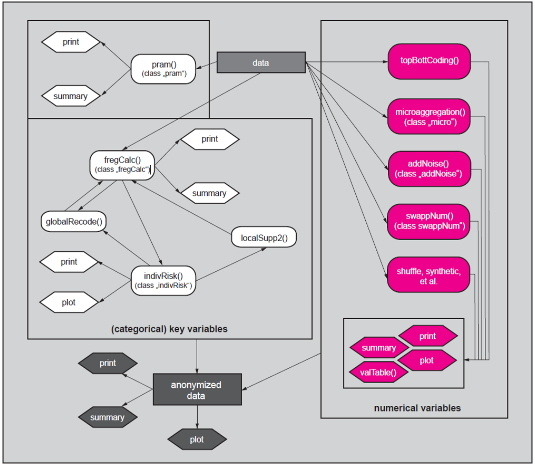


Figure: Certain procedures in package *sdcMicro* and their relationship

Non-perturbation methods - Example

Time to Code!

Perturbation methods

This approach allows wider dissemination of data, although released are perturbed (modified) data.
There is a wide range of available methods for perturbation of data. Namely:

- **Noise masking**: adding e.g. normally distributed errors $Z = X + \varepsilon_j \sim N(0, \sigma^2_{\varepsilon_j})$
- **Microaggreagtion**: grouping similar records together and replacing them with aggregate values, e.g. with their mean
- **Record swapping**: exchanging values of specific attributes between records
- **Rounding**: replacing original values with rounded ones
- **Resampling**: creating multiple versions of a dataset by repeatedly drawing samples from the original data, releasing e.g. mean calculated from this samples
- **PRAM**: probabilistically altering the values of categorical variables in a dataset according to a predefined probability matrix

Perturbation methods - Example

As the example for this approach were selected methods recommended by **Centre of Excellence on Statistical Disclosure Control** in **Harmonised protection of census data**.

1. **Target Record Swapping**
   - pre-tabular method applied directly to microdata
   - selectively exchanging (swapping) values between specific pairs of records in a dataset. The pairs are chosen based on certain criteria, such as being within the same geographic area or having similar attribute values.
2. **Cell Key Method**
   - post-tabular method applied to aggregated data
   - protect sensitive data in frequency tables
   - adding small random noise to the cell counts

Perturbation methods - Example

Time to Code!

Synthetic methods - Definition and Origins

What is Synthetic data?

- Data generated by models trained on real world data
- Origins
  - *Statistical community:* multiple imputation for nonresponse, i.e. replace sensitive values with "imputed" values and aimed at ensuring valid statistical inference
  - *Computer science community:* mitigating risk of disclosure & easier data access for ML models
- Joint and conditional/sequential models
  - Joint modeling captures entire data distribution simultaneously, powerful but computationally heavy
  - Conditional/sequential modeling generates data variable-by-variable based on conditional relationships, offering flexibility and scalability, but can miss intricate dependencies if not explicitly included
- Preserve correlation structure (quasi-identical distribution) so analysis on synthetic data provide approximately the same answers

Synthetic methods - Purpose of Synthetic Data

Why generate synthetic micro data?

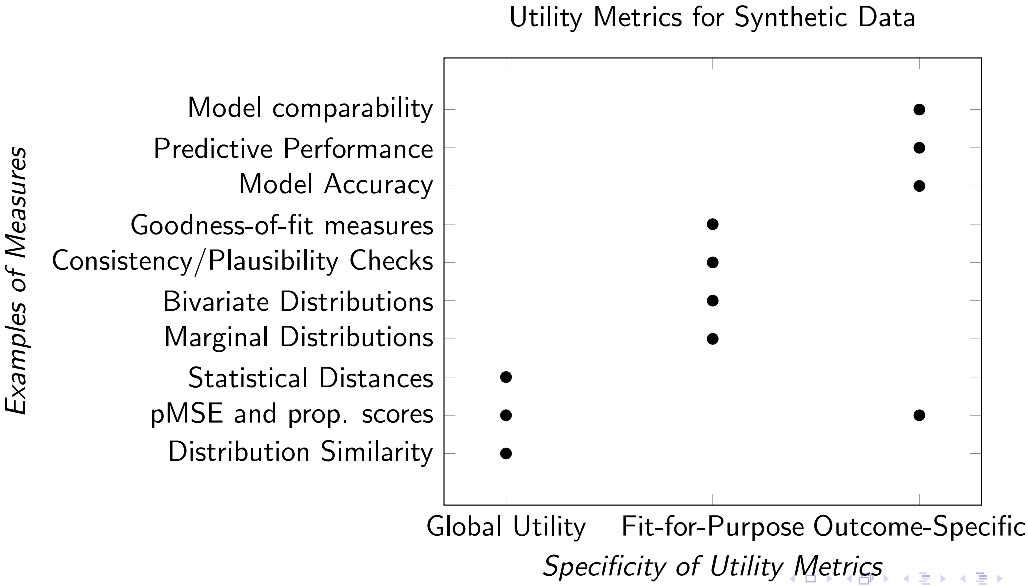
- Demand from scientific community and public
- GDPR and other national data protection laws often make dissemination of micro data impossible
- Highly censored microdata
- Long and tedious process to get access

Synthetic methods - Utility and Risk Evaluation

Utility and Risk Evaluation

- Trade-off
- Utility Evaluation of synthetic data (many methods common to measures measuring validity of perturbated data)
  - Global (direct comparison on an aggregated level)
  - Fit-for-Purpose (starting point of assessment)
  - Outcome-Specific (for a specific analysis task)
- Risk Assessment remains a challenge

Synthetic methods - Utility measures



Synthetic methods - Risk measures

Link between original and synthetic data is **broken**, however this does not mean that fully synthetic data can be assumed to have no risk of spilling sensitive information.

- Synthetic identical records with unique combination of attributes in the original data
- Average of distances to closest neighbours
- Probability measures
  - Within Equivalence Class Attribution Probability (WEAP): Probability of matching individuals in synthetic data to their counterparts within quasi-identifier variable groups
  - Targeted Correct Attribution Probability (TCAP): Likelihood that synthetic data correctly classifies individuals
- Measures based on attacker scenarios
  - Membership attacks (attacker will learn that a certain record was present in the original data, e.g. Survey of Prison Inmates)
  - Inference attacks (what an attacker can learn about unknown sensitive value after seeing the synthetic data)

Synthetic methods: synthpop

Based on conditional modeling, for non complex data structures, includes analysis functions

Procedure

- Data are synthesised via the function *syn()*
- Choice of synthesizing method
  - Random sampling with replacement from the original data, if not specified
  - Parametric
    - various types of regression, e.g. normal linear, logistic, polytomous, ordered, etc.
  - Non-parametric
    - classification and regression trees models
    - CART (default)
  - Log-linear model approach for categorical data implemented via an ipf procedure (light joint-modelling)
- Use implemented comparison functions for fast utility measures (e.g. *compare()* *utility.gen()*)

synthpop - Example

Time to Code!

Synthetic methods: simPop

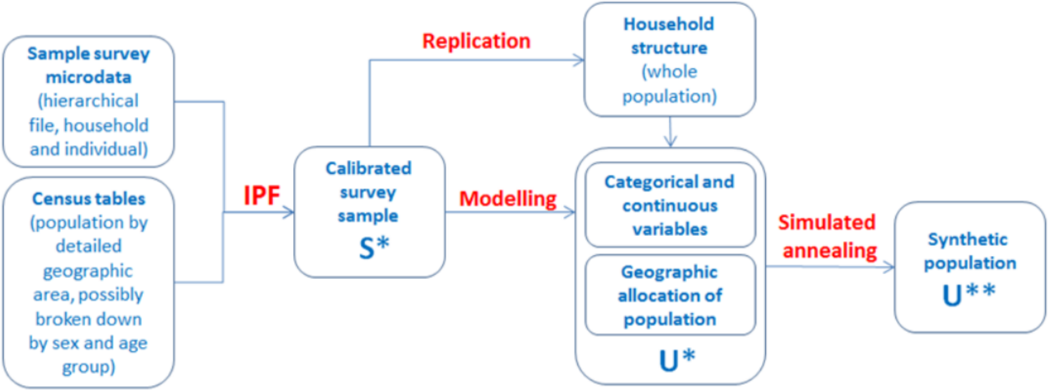

Sample survey microdata (hierarchical file, household and individual)
Census tables (population by detailed geographic area, possibly broken down by sex and age group)
IPF
Calibrated survey sample S*
Replication
Household structure (whole population)
Modelling
Categorical and continuous variables
Geographic allocation of population
U*
Simulated annealing
Synthetic population U**

Figure: Certain procedures in package *simPop* and their relationship

Synthetic methods: simPop

Specifically designed for synthesising populations (e.g. persons in households)

Model based Procedure

1. Specify inputs
2. Initialise synthetic population by defining household structure
3. Simulation of categorical variables sequentially
4. Simulation of continuous variables sequentially

Additional modelling functions

- Simulate categorical variables taking relationships between household members into account *simRelation()*
- Splitting continuous variables into components *simComponents()*
- Fix age heaping *correctHeaps()*, *correctSingleHeap()*
- Calibrate synthetic population to “fit” selected distributions with *calibPop()* using a simulated annealing algorithm

simPop - Example

Time to Code!

Synthetic methods: GANs - Era of GANs

Prompt:
An illustration of an avocado sitting in a therapist's chair, saying "I just feel so empty inside" with a pit-sized hole in its center. The therapist, a spoon, scribbles notes.

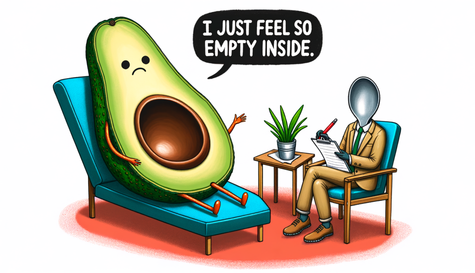


Figure: Result of Text to Image

https://openai.com/dall-e-3

Synthetic methods: GANs - Overview of GAN Structure

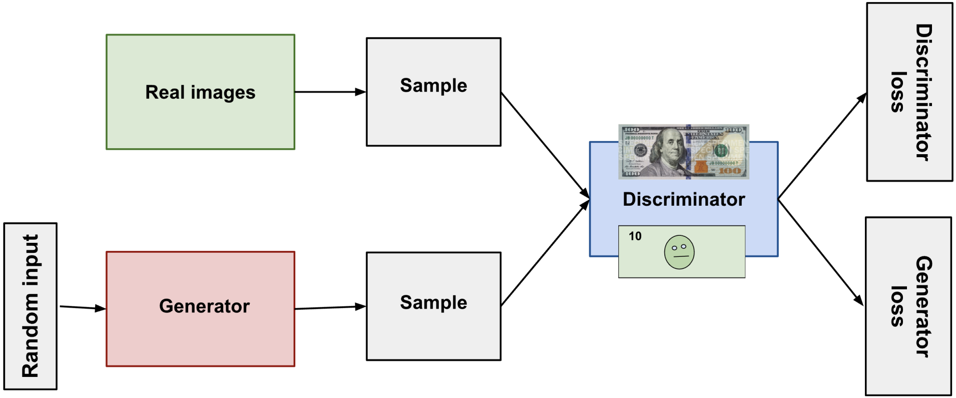


Figure: The GAN-Framework

Synthetic methods: GANs - Minimax Loss

- The generator tries to minimize the following function while the discriminator tries to maximize it:

$$\mathcal{L}(G, D) = \min_{G} \max_{D} \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

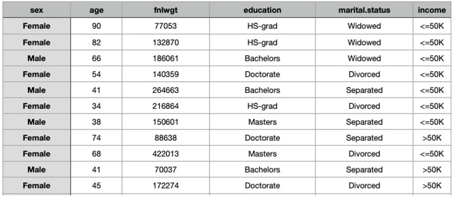
- $D(x)$ is the discriminator's estimate of the probability that real data instance x is real
- $\mathbb{E}_x$ is the expected value over all real data instances
- $G(z)$ is the generator's output when given noise z
- $D(G(z))$ is the discriminator's estimate of the probability that a fake instance is real
- $\mathbb{E}_z$ is the expected value over all random inputs to the generator (in effect, the expected value over all generated fake instances $G(z)$)

https://developers.google.com/machine-learning/gan/loss

Synthetic methods: GANs - CTGAN: Conditional Tabular GAN for Tabular Data Synthesis

- **Purpose**: Generate realistic synthetic tabular data.
- **Core Idea**: Uses GANs to capture the distribution of tabular data, including complex relationships between features.
- **Advantages**:
  - Handles mixed data types (continuous and categorical).
  - Captures complex dependencies in the data.
  - Useful for data augmentation, privacy preservation, and data sharing.

Synthetic methods: GANs - CTGAN: Why use it?



sex	age	fnlwgt	education	marital.status	income
Female	90	77053	HS-grad	Widowed	<=50K
Female	82	132870	HS-grad	Widowed	<=50K
Male	66	186061	Bachelors	Widowed	<=50K
Female	54	140359	Doctorate	Divorced	<=50K
Male	41	264663	Bachelors	Separated	<=50K
Female	34	216864	HS-grad	Divorced	<=50K
Male	38	150601	Masters	Separated	<=50K
Female	74	88638	Doctorate	Separated	>50K
Female	68	422013	Masters	Divorced	<=50K
Male	41	70037	Bachelors	Separated	>50K
Female	45	172274	Doctorate	Divorced	>50K

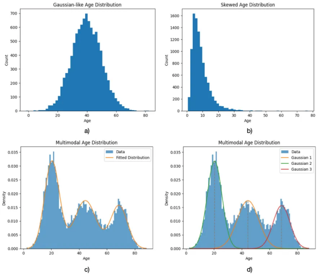
Figure: Heterogeneous Tabular Data

CTGAN was developed to deal with the challenges posed by tabular datasets, handling mixed data types (numeric and categorical).

1. Other GANs, e.g., ADS-GAN that use Wasserstein GAN with Gradient Penalty, in order to improve training stability and convergence time, aren't able to handle mixed data types.

https://miro.medium.com/v2/resize:fit:1400/format:webp/1*GnQJiD9-jAz9dR8vKP4Thg.png

Synthetic methods: GANs - CTGAN: How does it work?



a) Gaussian-like Age Distribution

b) Skewed Age Distribution

c) Multimodal Age Distribution

d) Multimodal Age Distribution

Figure: Heterogeneous Tabular Data: Gaussian like versus skewed data distribution (fig. a & b). Multimodal distribution decomposed into distributions with distinct modes (fig. c & d).

Numerical Data in tabular data are often non-Gaussian.

1. CTGAN therefore uses VGM (Variational Gaussian Mixture) model for mode-specific normalization

https://arxiv.org/pdf/1907.00503
https://iopscience.iop.org/article/10.1088/1757-899X/1294/1/012024

Synthetic methods: GANs - CTGAN: How does it work?

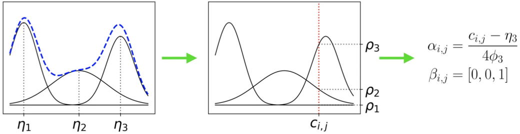
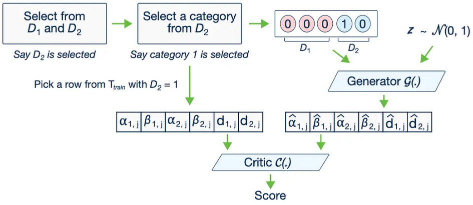

η_1
η_2
η_3
ρ_3
ρ_2
ρ_1
$c_{i,j}$
$\alpha_{i,j} = \frac{c_{i,j} - \eta_3}{4\phi_3}$
$\beta_{i,j} = [0, 0, 1]$

Figure: Example of mode-specific normalization

Using a VGM (Variational Gaussian Mixture) model, each value in a continuous feature is represented by a one-hot vector indicating its sampled mode and a scalar that represents the value normalized according to that mode

https://arxiv.org/pdf/1907.00503
https://iopscience.iop.org/article/10.1088/1757-899X/1294/1/012024

Synthetic methods: GANs - CTGAN: How does it work?



Select from D_1 and D_2
Say D_2 is selected
Select a category from D_2
Say category 1 is selected
0 0 0 1 0
D_1
D_2
$z \sim \mathcal{N}(0, 1)$
Generator $\mathcal{G}(.)$
Pick a row from T_{train} with $D_2 = 1$
$\alpha_{1,j}$ $\beta_{1,j}$ $\alpha_{2,j}$ $\beta_{2,j}$ $d_{1,j}$ $d_{2,j}$
$\hat{\alpha}_{1,j}$ $\hat{\beta}_{1,j}$ $\hat{\alpha}_{2,j}$ $\hat{\beta}_{2,j}$ $\hat{d}_{1,j}$ $\hat{d}_{2,j}$
Critic $\mathcal{C}(.)$
Score

Figure: Training-by-sampling in the CTGAN model

Input:

- Conditional vector specifying desired attributes. Noise vector from a standard normal distribution.
- With training-by-sampling, examples are conditioned on the possible values of categorical features, sampled according to their log-frequency.

https://arxiv.org/pdf/1907.00503
https://iopscience.iop.org/article/10.1088/1757-899X/1294/1/012024

Synthetic methods: GANs - CTGAN: How does it work?

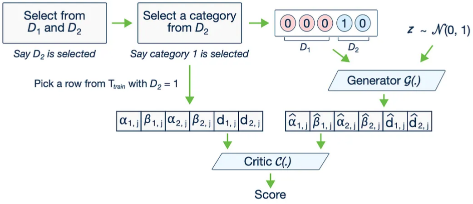

Select from D_1 and D_2
Say D_2 is selected
Select a category from D_2
Say category 1 is selected
0 0 0 1 0
D_1
D_2
$z \sim \mathcal{N}(0, 1)$
Generator $\mathcal{G}(.)$
Pick a row from T_{train} with $D_2 = 1$
$\alpha_{1,j}$ $\beta_{1,j}$ $\alpha_{2,j}$ $\beta_{2,j}$ $d_{1,j}$ $d_{2,j}$
$\hat{\alpha}_{1,j}$ $\hat{\beta}_{1,j}$ $\hat{\alpha}_{2,j}$ $\hat{\beta}_{2,j}$ $\hat{d}_{1,j}$ $\hat{d}_{2,j}$
Critic $\mathcal{C}(.)$
Score

Figure: Training-by-sampling in the CTGAN model

Conditional Vector Encoding:

- One-hot encoding for categorical columns.
- Gaussian Mixture Models (GMM) for continuous columns.

https://arxiv.org/pdf/1907.00503
https://iopscience.iop.org/article/10.1088/1757-899X/1294/1/012024

Synthetic methods: GANs - CTGAN: How does it work?

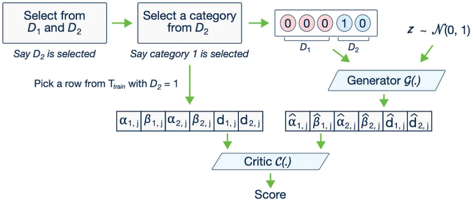

Select from D_1 and D_2
Say D_2 is selected
Select a category from D_2
Say category 1 is selected
0 0 0 1 0
D_1
D_2
$z \sim \mathcal{N}(0, 1)$
Generator $\mathcal{G}(.)$
Pick a row from T_{train} with $D_2 = 1$
$\alpha_{1,j}$ $\beta_{1,j}$ $\alpha_{2,j}$ $\beta_{2,j}$ $d_{1,j}$ $d_{2,j}$
$\hat{\alpha}_{1,j}$ $\hat{\beta}_{1,j}$ $\hat{\alpha}_{2,j}$ $\hat{\beta}_{2,j}$ $\hat{d}_{1,j}$ $\hat{d}_{2,j}$
Critic $\mathcal{C}(.)$
Score

Figure: Training-by-sampling in the CTGAN model

Generator Network:

- Neural network processes the combined input vector.
- Outputs a synthetic sample.

https://arxiv.org/pdf/1907.00503
https://iopscience.iop.org/article/10.1088/1757-899X/1294/1/012024

Synthetic methods: GANs - CTGAN: How does it work?

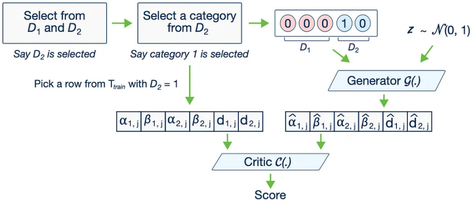

Select from D_1 and D_2
Say D_2 is selected
Select a category from D_2
Say category 1 is selected
0 0 0 1 0
D_1
D_2
$z \sim \mathcal{N}(0, 1)$
Generator $\mathcal{G}(.)$
Pick a row from T_{train} with $D_2 = 1$
$\alpha_{1,j}$ $\beta_{1,j}$ $\alpha_{2,j}$ $\beta_{2,j}$ $d_{1,j}$ $d_{2,j}$
$\hat{\alpha}_{1,j}$ $\hat{\beta}_{1,j}$ $\hat{\alpha}_{2,j}$ $\hat{\beta}_{2,j}$ $\hat{d}_{1,j}$ $\hat{d}_{2,j}$
Critic $\mathcal{C}(.)$
Score

Figure: Training-by-sampling in the CTGAN model

Training:

- Discriminator provides feedback to improve generation and the generator adjusts the synthetic data to match the conditional distribution of the specified column values, ensuring the generated data is representative of the real conditional relationships.

https://arxiv.org/pdf/1907.00503
https://iopscience.iop.org/article/10.1088/1757-899X/1294/1/012024

Synthetic methods: GANs - R Interface for CTGAN

R Interface for CTGAN: A wrapper around CTGAN that brings the functionalities to R users. More details can be found in the corresponding repository: https://github.com/kasaai/ctgan.

- R Interface is obsolete (code was last updated 4 years ago).
- Problems with installation.
- Problems with passing of parameters (CTGAN has seen several changes and refactoring).

Synthetic methods: GANs - Python Interface for CTGAN

Python Interface for CTGAN: CTGAN is a collection of Deep Learning based synthetic data generators for single table data, which are able to learn from real data and generate synthetic data with high fidelity.

- Use CTGAN through the SDV library: https://github.com/sdv-dev/SDV
  1. Preprocessing: Data Preparation
  2. Modelling: CTGANSynthesizer
  3. Sampling: Sample Realistic Data
- Use the CTGAN standalone library: https://github.com/sdv-dev/CTGAN
  1. When using the CTGAN library directly, you may need to manually preprocess your data into the correct format, for example:
     - 1.1 Continuous data must be represented as floats
     - 1.2 Discrete data must be represented as ints or strings
     - 1.3 The data should not contain any missing values

Thank you for your attention

SwissAnon

Swiss Data Anonymization Competence Center

https://swissanon.ch