FRI Data Science Project Competition 2023
Interim report

A Survey and Empirical Comparison of Synthetic Data Generation Methods

Valter Hudovernik and Martin Jurkovič

Abstract

Synthetic relational data generation is a niche field with growing interest in the last years from the academia and industry. We have researched the methods for generation and evaluation of synthetic tabular relational data. We will evaluate and use the best performing model to generate data from Zurich Insurance Group. They will be able to use this data for better ML models, faster data ingestion from their branches and easier GDPR compliance.

Keywords

Synthetic data generation, relational data, synthetic data evaluation

Advisors: prof. dr. Erik Štrumbelj

Introduction

Synthetic data generation is a growing area of research, with 1,751 articles published in 2022 more than double the amount published in 2019. This is not surprising as the applications of such methods may have significant effects on industry with over 70,000 patent for synthetic data generation filled in the last year.

In this report we examine and evaluate methods for generating relational synthetic data. This kind of data most impacts medical and financial fields. It is also relevant for insurance companies such as our industry partner Zurich Insurance . In these domains synthetically generated data addresses privacy concerns, costs and speed of development and may even improve performance of existing models.

Synthetic data can help to ensure the privacy of sensitive data. By generating artificial data instead of using real data, it is possible to prevent the disclosure of sensitive information. This is particularly important in fields such as healthcare, finance and insurance where privacy laws and regulations must be adhered to.

In many cases generating synthetic data can be more cost-effective and faster than collecting real data as it does not require additional resources or time-consuming data collection processes. This is especially important in time critical situation when quick iteration and prototyping is important, as is the case with the Zurich Insurance data department.

Lastly with the rising complexity of predictive models, requiring more and more data, synthetically generated data can provide additional data points to supplement existing datasets. Especially in granular and segmentation analyses and can alleviate issues like class imbalance, impurities, and privacy concerns, potentially resulting in better performing models.

Synthesizing relational databases poses additional challenges to tabular data synthesis. Additionally to modeling the distribution of each column and the relationships between them, the relationships between tables within the same database must also be taken into account. This includes four types of parent-child relationships, namely linear (with only parent-child relationships), multiple-child (with a primary key referenced by various tables), multiple-parent (with columns referencing primary keys of multiple tables), and multiple-child and multiple-parent . These relationships introduce new non-trivial constraints that generative models must address in order to effectively generate synthetic data. In the following section we formally define relational data and take an extensive look at relevant works for synthetic relational data generation.

Related Work

Relational Data Generation Methods

The Synthetic data vault [1]by Patki N., Wedge R. and Veeramachaneni K. (2016) is an open source python library for automatic synthetic data generation and evaluation. They focus on

[1]Citations for all methods ommited due to 2 page report limit.

4 key points: creation of synthetic data for one table with their own algorithm, recursive conditional parameter aggregation technique which is a method for recursive table modeling in a relational database, privacy protection and demonstration of the utility of synthetic data. To model the data the user must specify the structure (metadata) of the data. Then the SDV model iterates through tables sequentially using a modelling algorithm designed to account for relationships between tables. For a table in the database, if other tables reference it, dependence exists and the SDV computes aggregate statistics for the other tables, which are then added to the original table, forming an extended table.

Row Conditional-Tabular Generative Adversarial Network (RC-TGAN) is a generative adversarial network (GAN) model that extends the tabular GAN to support modeling and synthesizing relational databases, proposed by Gueye M., Atta. and Dumas M. (2022). The model extends the original TGAN model to support relational datasets by incorporating conditional data from parent rows into the design of the GAN model corresponding to the child table. RC-TGAN has the inherent ability to address all relationship schemas without additional processing steps. They also extend RC-TGAN to maximize the capture of the influence that grandparent (or higher-level ancestor) rows may also have on their grandchild rows, thus preventing the loss of this connection when the parent table rows fail to transfer this relationship information.

Realistic Relational and Tabular Transformer proposed by Solatorio A. and Dupriez O. (2023) is a relational synthetic data generation model based on GPT-2 . To the best of our knowledge this model may only generate single parent relational data. The method treats the parent table independently and models it using a non-relational tabular data model with a GPT-2 encoder with a causal language model (LM) head. After training the parent table model, the encoder part of the generator is frozen and used to conditionally model the child tables. For each child table a new model needs to be constructed with the following structure. The conditional model is a sequence-to-sequence (Seq2Seq) transformer. It uses the pretrained parent encoder and trains the GPT-2 decoder with a causal LM head to conditionally generate observations from the child table of arbitrary length.

Additionally we found many commercial tools for synthetic data generation. However, of the ones we evaluated only two support generation of relational data: **MostlyAI** and **GretelAI**.

Synthetic Data Evaluation

Like synthetic data generation methods, evaluation methods are split between evaluating single table or hierarchical data.

SDMetrics is an open source evaluation library for synthetic data, developed by the organization DataCebo, which is also the organization behind the SDV library. They split their metrics by data granularity.

Metrics implemented in the SDMetrics library

For single column:
- *Category* and *Range Coverage*: measures whether a synthetic column covers all the possible categories or covers the full range of the values present in a real column.
- *Boundary Adherence*: the comparison of minimum/maximum ranges
- *KSComplement, TVComplement*: comparison of shapes (marginal distributions, histograms). KSComplement uses the Kolmogorov-Smirnov statistic for numerical data, whereas TVComplement computes the Total Variation Distance (TVD) between the real and synthetic categorical columns.
- *Statistic Similarity*: comparison of summary statistics (mean, median and standard deviation).
- *Missing value similarity*

For column pairs:
- *Contingency Similarity*: comparison of 2D distributions.
- *Correlation Similarity*

For relational data, there are not many developed metrics. The most used metric is Logistic Detection (LD) metric, where for each table in the hierarchical dataset it's synthetic pair is used to calculate LD. An extension of LD is also used to evaluate the ability of the generative model to preserve the parent-child relationship by applying LD on the denormalized synthetic tables, referred to as parent-child logistic detection (P-C LD).

Experimental Results

Datasets

Linear relationships: Airbnb , Rossmann , Biodegradability , Mutagenesis
Multiple-child relationships: Telstra , Walmart
Multiple child and parent relationships: Coupon Purchase Prediction

Zurich Customers Dataset

Zurich Insurance Company provided an anonymized and sampled dataset from the usage data of their platform. The data was automatically generated based on the real data from the company's database. The obtained data is split into three datasets: customer data, policy data and claim data. Datasets are connected using primary and foreign keys. Primary key of a customer is available as a foreign key for policies and claims. Primary key of a policy is available as a foreign key in claims.

Future work

Now that we have thoroughly researched the methods for generating synthetic relational data we will apply and evaluate the methods on the Zurich Customers dataset, as well as on the other datasets used in research papers to evaluate the performance of the models. Beside that we will also try to evaluate the performance of the models used by ZCAM d.o.o. by training them on the synthetic data.