Integrating domain constraints in tabular data generation process

Salijona Dyrmishi[1], Mihaela Cătălina Stoian[2], Eleonora Giunchiglia[3], and Maxime Cordy[1]

[1] University of Luxembourg, Luxembourg: firstname.surname@uni.lu
[2] University of Oxford, United Kingdom: mihaela.stoian@st-hildas.ox.ac.uk
[3] Imperial College London, United Kingdom: e.giunchiglia@imperial.ac.uk

Abstract. Deep Generative Models (DGMs) excel at generating synthetic tabular data but struggle to enforce domain-specific constraints essential in applications like ML robustness testing. To overcome this, we introduce Constrained Deep Generative Models (C-DGMs), which use a Constraint Layer to ensure that generated data adhere to predefined rules while staying true to the original distributions. We extend this approach to create Constrained Adversarial DGMs (C-AdvDGMs), which generate adversarial examples that both satisfy domain constraints and effectively assess the robustness of machine learning models.

Keywords: Deep Generative Models · Tabular · Adversarial Attack

1 Introduction

Deep Generative Models (DGMs) are widely used for generating synthetic tabular data, addressing challenges like data scarcity, promoting fairness, and ensuring privacy in sensitive datasets. They have also been adapted for adversarial data generation to test the robustness of ML models. However, whether used for data synthesis or adversarial attacks, DGMs must comply with domain constraints, particularly in tabular data, where such rules are explicit. For example, in clinical data sets, values such as "maximum hemoglobin level" must always exceed "minimum hemoglobin level." Existing DGMs, while adept at capturing distributions, cannot inherently enforce these constraints. Sample rejection is not a viable solution in cases where constraint violations dominate the generated data (up to 100%)[1]. To address this, we propose Constrained DGMs (C-DGMs) that ensure compliance with predefined constraints while maintaining the fidelity of the original DGM output. These C-DGMs are then extended into C-AdvDGMs, which generate domain-compliant adversarial examples.

2 Constrained Deep Generative Models for Tabular Data

We introduce the Constraint Layer (CL), a differentiable component that enforces user-defined constraints on data generated by Deep Generative Models (DGMs). While the explanation below uses Generative Adversarial Networks (GANs) as an example, the CL can be integrated with most DGM architectures.

GANs map random noise through a generator to produce samples resembling real data but lack mechanisms to enforce essential constraints. The CL addresses this by transforming DGM outputs into the data feature space, evaluating them against linear inequality constraints, and minimally adjusting violations with computed bounds to ensure feasibility. It operates during training (C-DGM) or as a post-processing step (P-DGM). In adversarial tasks, C-AdvDGMs extend this by generating adversarial examples from an input x (e.g., real data), repairing them via the CL, and evaluating them for generative (L_{GAN}) and adversarial (L_{adv}) objectives (Fig. Figure 1). This ensures generated data is both task-relevant and constraint-compliant.

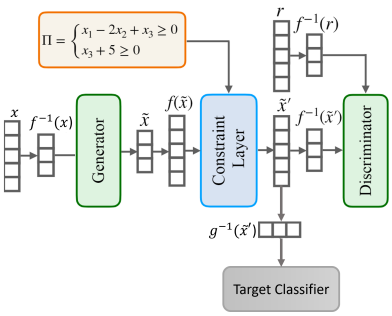

Fig. 1. Overview on how to integrate CL into a Adversarial GAN-based model.

3 Results

The addition of the constraint repair layer reduced the average constraint violation rate from 49.89% to 0% across 6 datasets and 5 models. Constrained Deep Generative Models (C-DGMs) outperformed their standard counterparts in 28 out of 30 cases, while PGD 17 times out of 30cases. The improvement in some cases was considerable achieving up to 6.5% improvement in utility (F1-score). Similarly, P-AdvDGMs and C-AdvDGMs demonstrated higher attack success rates (ASR) in most cases compared to their unconstrained counterparts, with ASR improvements reaching up to 62%. These results highlight the effectiveness of adding constraints to network topology.

4 Future directions

In this work, we focus on constraints that can be expressed as linear inequalities. While this covers a wide range of scenarios, some relationships among features may demand more expressive constraint representations. In the case of iterative generation process for adversarial examples some temporal restrictions might exist as well.

References

1. Stoian, M., Dyrmishi, S., Cordy, M., Lukasiewicz, T. & Giunchiglia, E. How realistic is your synthetic data? Constraining deep generative models for tabular data. *The Twelfth International Conference on Learning Representations (ICLR)*. (2024)
2. Dyrmishi, S., Stoian, M., Giunchiglia, E. & Cordy, M. Deep generative models as an adversarial attack strategy for tabular machine learning. *International Conference on Machine Learning and Cybernetics (ICMLC)*. (2024)