Sythetic Data Generation

School of Data science

The Chinese University of Hong Kong, Shenzhen

August 26, 2022

Outline

1. Background of tabular data generation

2. Overview of generative model for tabular data

Part 1. Background of tabular data generation

Background of tabular data generation

- Heterogeneous Property: mixed type data.
  - Different from image and language data, tabular data has dense numerical and sparse categorical features.
    e.g.,categorical, ordinal, continuous

- Ubiquitous in many crucial applications:
  - medical diagnosis based on patient history
  - predictive analytics for financial applications,
    e.g., risk analysis, estimation of creditworthiness, the recommendation of investment strategies, and portfolio management

Difficulties for deal with tabular data

- Low-quality training data,
    - e.g., missing values, class-imbalanced

- Complex or irregular dependencies between different columns,
    - e.g., a change of a categorical feature can entirely flip a prediction on tabular data
    - Many features are uninformative

- Handling the categorical features remains particularly challenging

Therefore, for classification and regression problems with tabular data, using tree ensemble models can outperform deep learning methods.[1][2]

[1] Grinsztajn, et al. Why do tree-based models still outperform deep learning on tabular data? NIPS 2022 workshop track.
[2] Shwartz et al. Tabular Data: Deep Learning is Not All You Need. ICML 2021 workshop track.

The goal of tabular data generation

- Utility: Incorporate more training data to enhance the performance
- Privacy: Sensitive data from users,
  - e.g., Information Leakage of medical diagnosis, Membership Inference Attacks
- Control generation,
  - e.g., class conditional generation, imputation

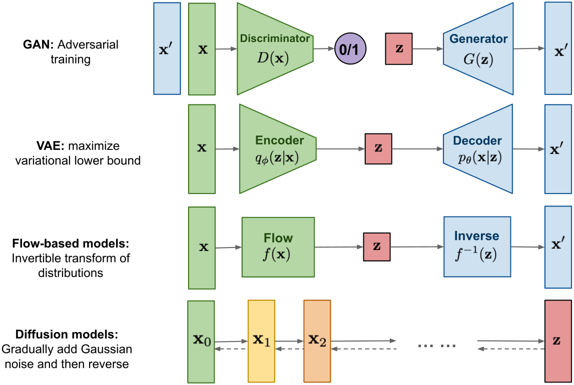
Part 2. Overview of generative model for tabular data

A survey of synthesizing tabular data[3]

Method	Based upon	Application
medGAN [46]	Autoencoder+GAN	Medical Records
TableGAN [145] DCGAN	General	
Mottini et al. [149]	Cramér GAN	Passenger Records
Camino et al. [150]	medGAN, ARAE	General
medBGAN, medWGAN [151]	WGAN-GP, Boundary seeking GAN	Medical Records
ITS-GAN [124]	GAN with AE for constraints	General
CTGAN, TVAE [144]	Wasserstein GAN, VAE	General
actGAN [126]	WGAN-GP	Health Data
VAEM [143]	VAE (Hierarchical)	General
OVAE [152]	Oblivious VAE	General
TAEI [44]	AE+SMOTE (in multiple setups)	General
Causal-TGAN [153]	Causal-Model, WGAN-GP	General
Copula-Flow [45]	Invertible Flows	General

TABLE III: Generation of tabular data using deep neural network models (in chronological order).

[3] Borisov, et al. (2021). Deep Neural Networks and Tabular Data: A Survey. ArXiv, abs/2110.01889.

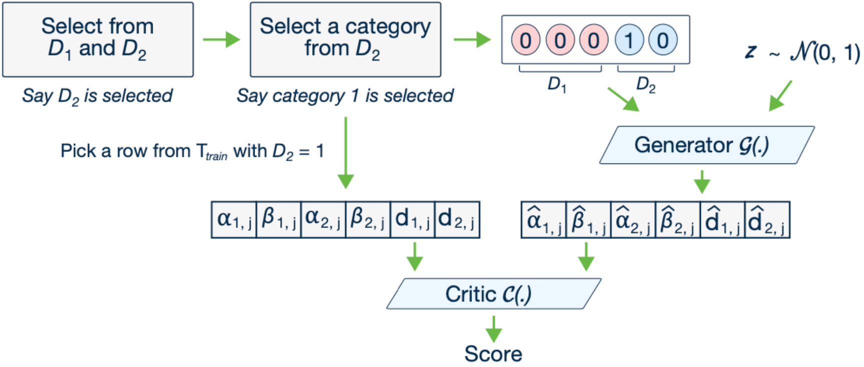


GFlowNet: A sampling method for discrete type data training by reinforcement criterion

[4] https://lilianweng.github.io/posts/2021-07-11-diffusion-models/#connection-with-noise-conditioned-score-networks-ncsn

Origin GAN: $\min_G \max_D \underset{\boldsymbol{x}\sim\mathbb{P}_r}{\mathbb{E}}[\log(D(\boldsymbol{x}))] + \underset{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}{\mathbb{E}}[\log(1-D(\tilde{\boldsymbol{x}}))]$

WGAN: $\min_G \max_{D\in\mathcal{D}} \underset{\boldsymbol{x}\sim\mathbb{P}_r}{\mathbb{E}}[D(\boldsymbol{x})] - \underset{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}{\mathbb{E}}[D(\tilde{\boldsymbol{x}}))]$



[5] Xu, Lei et al. Modeling Tabular data using Conditional GAN. NeurIPS (2019).

Details for CTGAN

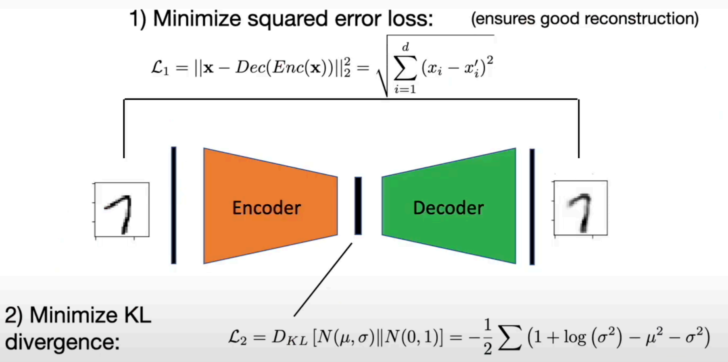
Table 3: Ablation study results on mode-specific normalization, conditional generator and training-by-sampling module, as well as the network architecture. The absolute performance change on real classification datasets (excluding MNIST) is reported.

	Mode-specific Normalization			Generater		Network Architechture		
Model	GMM5	GMM10	MinMax	w/o S.	w/o C.	GAN	WGANGP	GAN+PacGAN
Performance	-4.1%	-8.6%	-25.7%	-17.8%	-36.5%	-6.5%	+1.75%	-5.2%

Rule of thumb:

- Conditional generation
- mode-specific normalization
- WGAN+gradient penalty

TVAE [7]



[6] Comments:

- The generator in GANs does not have access to real data during the entire training process; thus, we can make CTGAN achieve differential privacy easier than TVAE.

[6] Image credits to Sebastian Raschka

[7] Xu, Lei et al. Modeling Tabular data using Conditional GAN. NeurIPS 2019.

VAE can generate discrete type data

- For categorical columns, we can use the softmax over all categories.
- For continuous-discrete columns (like salary), we can model it as a continuous variable and discretize it in the end
- For ordinal-discrete columns (like ratings), we can use ordinal regression likelihood[8].

[8] Paquet, et al. A hierarchical model for ordinal matrix factorization. Statistics and Computing, 22, 945-957.

Heterogeneous VAE [9]

Basic idea: VAEM uses a hierarchy of latent variables, which fits in two stages.

- In the first stage, learn one type-specific VAE for each dimension. These initial one-dimensional VAEs capture marginal distribution properties and provide a latent representation that is more homogeneous across dimensions.
- In the second stage, another VAE is used to capture dependencies among the one-dimensional latent representations from the first stage.

[9] Ma, et al. VAEM: a Deep Generative Model for Heterogeneous Mixed Type Data. NIPS 2020.

Normalizing Flows [10]

Modeling:

$$\mathbf{z} \sim \pi(\mathbf{z}), \mathbf{x} = f_\theta(\mathbf{z}), \mathbf{z} = f_\theta^{-1}(\mathbf{x})$$

$$p_\theta(\mathbf{x}) = \pi(\mathbf{z}) \left| \det \frac{d\mathbf{z}}{d\mathbf{x}} \right| = \pi\left(f_\theta^{-1}(\mathbf{x})\right) \left| \det \frac{df_\theta^{-1}}{d\mathbf{x}} \right|$$

Training objective:

$$\min_\theta \mathcal{L}(\theta|\mathcal{D}) = -\min_\theta \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \log p_\theta(\mathbf{x})$$

Key problem:

- Normalizing Flow cannot model discrete type data

Solution:

- Dequantization, i.e., adding real-valued noise to the discrete data.
  - Uniform dequantization
  - variational dequantization

[10] Lee, et al. Differentially Private Normalizing Flows for Synthetic Tabular Data Generation. AAAI 2022.

Autoregressive model

Modeling:

$$p(\mathbf{x}) = \prod_{i=1}^{D} p\left(x_i \mid x_1, \ldots, x_{i-1}\right) = \prod_{i=1}^{D} p\left(x_i \mid x_{1:i-1}\right)$$

Key problem:

- Tabular data is not sequential data like images or language

Solution:

- Using Masked Autoencoder, e.g., MADE (Masked Autoencoder for Distribution Estimation)

Comments:

- No work using Autoregressive model for tabular data
- A library[11] using traditional machine learning methods is available.
- self-supervision loss is attractive/somehow promising, but with high limitations in tabular data

[11] Mahiou, et al. dpart: Differentially Private Autoregressive Tabular, a General Framework for Synthetic Data Generation. ICML 2022 workshop track.

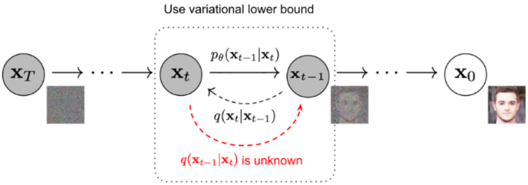


Fig. 2. The Markov chain of forward (reverse) diffusion process of generating a sample by slowly adding (removing) noise. (Image source: Ho et al. 2020 with a few additional annotations)

Training objective:

$$\min_\theta L_{\mathrm{VLB}} = \min_\theta \mathbb{E}_q [\underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \| p\left(\mathbf{x}_T\right)\right)}_{L_T} + \sum_{t>1} \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)}_{L_{t-1}} \underbrace{-\log p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)}_{L_0}]$$

Details for Diffusion model

- Diffusion model seems to be a promising research direction on tabular data generation
- Some key points to be solved:
  - How to deal with discrete columns?
    - Discrete diffusion [12]
  - How to deal with the relationship between discrete columns and continuous columns?
    - Conditional diffusion [13]

[12] Austin, et al. Structured Denoising Diffusion Models in Discrete State-Spaces. NIPS 2021.

[13] Batzolis, et al. Conditional Image Generation with Score-Based Diffusion Models.

GFlowNet [14]

GFlowNet: A sampling method for discrete type data training by reinforcement objective.

Flow consistency equations:

$$\sum_{s,a:T(s,a)=s'} F(s,a) = R\left(s'\right) + \sum_{a'\in\mathcal{A}(s')} F\left(s',a'\right).$$

Training objective:

$$\begin{aligned}\mathcal{L}_{\theta,\epsilon}(\tau) = \sum_{s'\in\tau\neq s_0} \Bigg(\log\left[\ \epsilon + \textstyle\sum_{s,a:T(s,a)=s'} \exp F_\theta^{\log}(s,a)\ \right] \\ - \log\left[\epsilon + R\left(s'\right) + \sum_{a'\in\mathcal{A}(s')} \exp F_\theta^{\log}\left(s',a'\right)\right]\Bigg)^2\end{aligned}$$

[14] Bengio, et al. Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation. NIPS 2021.

Details for GFlowNet

- GFlowNet explicitly models the relationship between discrete columns
- Some key points to be solved:
  - GFlowNet can only deal with discrete columns, we need to deal with the relationship between discrete columns and continuous columns.
    - Conditional diffusion/ Conditinal GAN

Takeaway

- We get familiar with the properties of tabular data and the difficulties for modelling tabular data
- We get familiar with all types of generative model, especially for synthesizing tabular data
- some open questions