Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心

基于GAN的表格数据生成

表格数据生成

李班　硕士研究生

2020年10月9日

内容提要

- 背景简介
- 基本概念
  - 生成模型
  - GAN 原理
  - 原始 GAN 的问题及解决方案
  - GAN 应用
  - GAN 优缺点
- CTGAN 算法原理
- 参考文献

背景简介

- 预期收获
  - 1. 了解生成对抗网络及其优缺点
  - 2. 了解如何利用 GAN 生成表格数据
  - 3. 了解 CTGAN 算法原理

背景简介

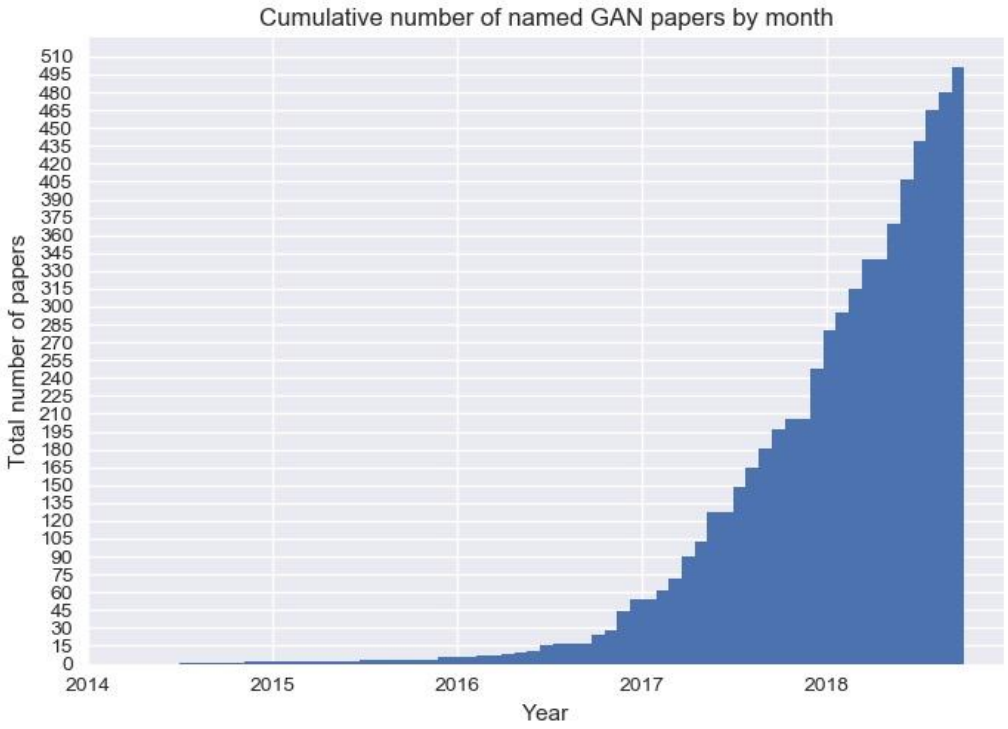
- 2014年，Ian Goodfellow 提出一种深度生成模型(Generative Adversarial Networks, GAN)

- 理论成果：DCGAN，CGAN，InfoGAN，WGAN，WGAN-GP，PacGAN……

- 应用成果：CycleGAN，SeqGAN，SGAN，TableGAN，GAIN，CTGAN……

背景简介

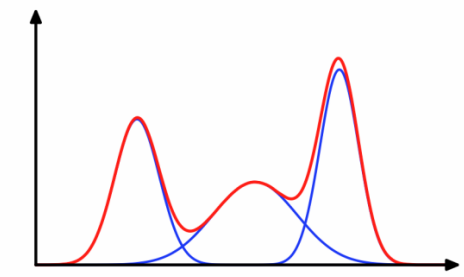
- The GAN Zoo



基本概念

基本概念

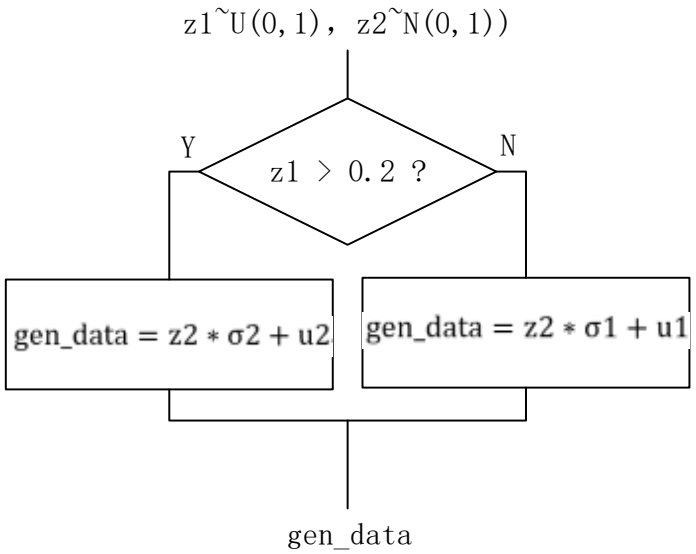
- 生成模型 – 能够随机生成观测数据的模型
  - 概率分布模型 + 采样
  - 概率分布模型：高斯混合模型（和其他类型的混合模型），隐马尔可夫模型，贝叶斯网络（如朴素贝叶斯，自回归模型），玻尔兹曼机（如受限玻尔兹曼机，深度信念网络），变分自编码器，生成对抗网络……
  - 生成观测数据的方法：模拟采样
    - 以高斯混合模型为例

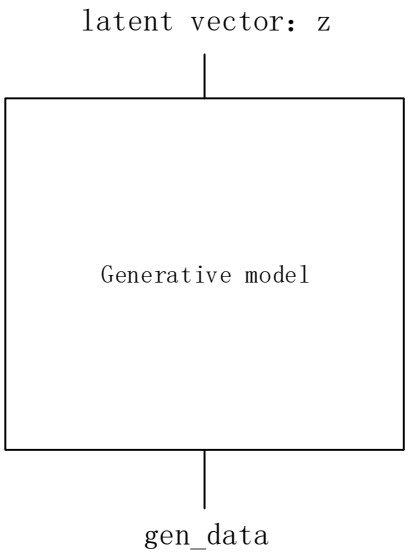


$$P(x|\theta) = \sum_{k=1}^{K} \alpha_k \phi(x|\theta_k) \quad \theta_k = (\mu_k, \sigma_k^2)$$

基本概念

$$\alpha 1 = 0.2, u1 = 1, \sigma 1^2 = 9$$
$$\alpha 2 = 0.8, u2 = -1, \sigma 2^2 = 9$$

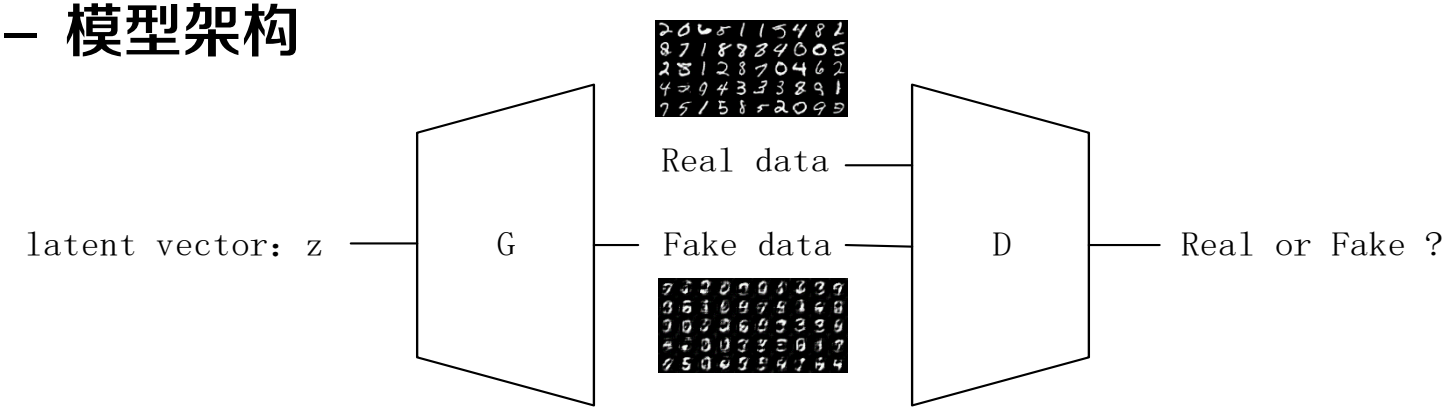




T	构建生成模型，生成和真实数据同分布的合成数据
I	真实训练数据
P	求解概率分布模型和对应采样方法
O	概率分布模型，采样方法

基本概念

- 生成对抗网络(Generative Adversarial Networks, GAN)
  - 模型架构



  - 目标函数

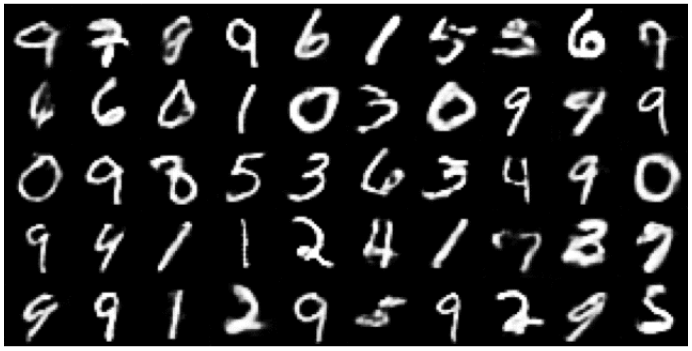
$$J^{(D)}(\boldsymbol{\theta}^{(D)}, \boldsymbol{\theta}^{(G)}) = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(z)\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

基本概念

– 训练过程：以手写数字数据集为例

1. 固定 G，训练 k 次 D，使得输入真实手写数字时，D 输出高分；输入 G 生成的手写数字时，D 输出低分；
2. 固定 D，训练 G 使得 G 生成的图片在 D 处得到更高分
3. 重复执行步骤 1，2



基本概念

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

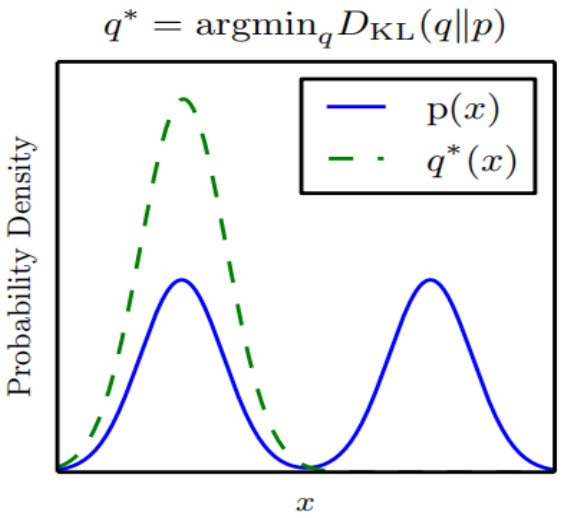
基本概念

- 原始GAN存在问题
  - 难以训练至最优状态
    - 梯度消失

$$KL(p||q) = \int p(x) log_2 \frac{p(x)}{q(x)} \mathrm{dx}$$

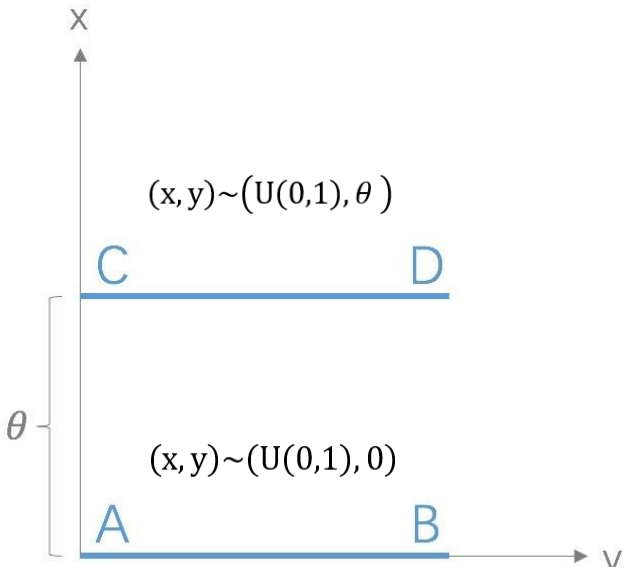
$$JS(p||q) = \frac{1}{2} KL(p||\frac{p+q}{2}) + \frac{1}{2} KL(q||\frac{p+q}{2})$$

    - 模式崩坏（丢失）：训练数据包含‘0~9’，只生成了‘1~3’



基本概念

- 解决：WGAN 和 WGAN-GP
  - WGAN 和 WGAN-GP 主要贡献：修改生成数据分布和目标分布的距离度量方式为 Wasserstein 距离（推土机距离）



$$L_{WGAN-GP}(D) = -\mathbb{E}_{x\sim p_r}[D(x)] + \mathbb{E}_{x\sim p_g}[D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim p_{\hat{x}}}\left[\|\nabla_x D(x)\|_p - 1\right]^2$$

$$L_{WGAN-GP}(G) = -\mathbb{E}_{x\sim p_g}[D(x)]$$

- $W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|,$
- $JS(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0, \\ 0 & \text{if } \theta = 0, \end{cases}$
- $KL(\mathbb{P}_\theta \| \mathbb{P}_0) = KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = \begin{cases} +\infty & \text{if } \theta \neq 0, \\ 0 & \text{if } \theta = 0, \end{cases}$

基本概念

- 评价 GAN 的生成器（评价生成数据的质量）
  - 影响因素：生成数据的真实性（清晰度）和多样性
  - 具体指标：
    - 边缘概率分布：累积概率分布曲线……
    - 相关性：假设检验，散点图
    - 机器学习效果：分别利用真实数据和生成数据训练机器学习模型，计算模型性能差异
    - Inception Score，FID， Wasserstein distance……
    - 根据具体的任务确定评价方式

基本概念

- GAN 应用场景
  - 去隐私（TableGAN）：利用 GAN 学习真实数据的概率分布合成不含隐私信息的假数据
  - 缺失值填充（GAIN）：利用GAN的生成器学到的数据各维度之间的相关性预测数据的缺失值
  - 半监督（SGAN）：利用有标签和无标签的数据训练生成模型生成更多有标签数据提升模型性能
  - 数据增强：GAN 有没有创造‘新数据’？
  - 图像领域：超分辨，风格迁移，图像翻译……
  - 序列生成：语音生成，文本生成，心（脑）电图生成……

基本概念

- GAN 优缺点
  - 相比以高斯混合模型为代表的显式建模方法：
    - 优点是模型的拟合能力更强，可以拟合更复杂的概率分布；
    - 缺点是GAN是一种隐式建模的方法，只可以从模型采样出生成数据，无法得到目标分布的具体表达式
  - 相比以自编码器为代表的其他深度生成模型：
    - 优点是生成数据的真实性更高；采样效率更高，可以并行批量生成数据；
    - 缺点是模型训练更困难，容易出现模式崩坏等问题

CTGAN 算法原理

CTGAN 算法原理

- 基于 GAN 的表格数据生成方法

T	构建表格数据生成模型
I	真实表格数据
P	1. 针对离散类型，连续类型数据预处理 2. 根据表格数据特点设计 GAN 模型架构，训练模型 3. 从训练好的生成模型采样出生成数据
O	和真实数据同分布的生成数据

CTGAN 算法原理

P	生成和真实数据同分布的合成数据
C	表格类型数据，一般具备混合的数据类型（离散和连续）
D	1. 生成多模式分布的连续数据； 2. 生成 one-hot 编码的离散数据； 3. 不平衡的训练数据引起模式崩坏的问题。
L	NeurIPS 2019

CTGAN 算法原理

- Mode-specific Normalization

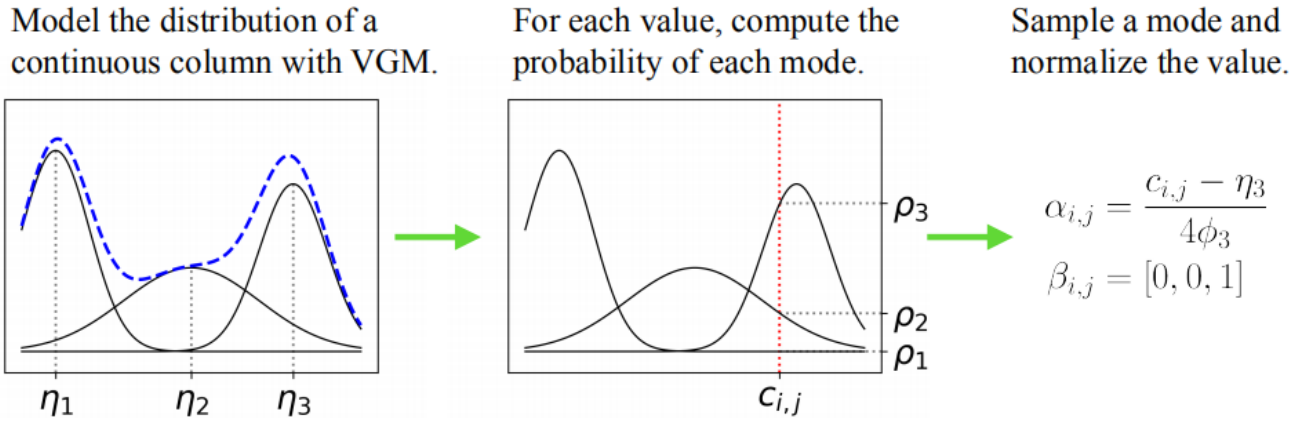
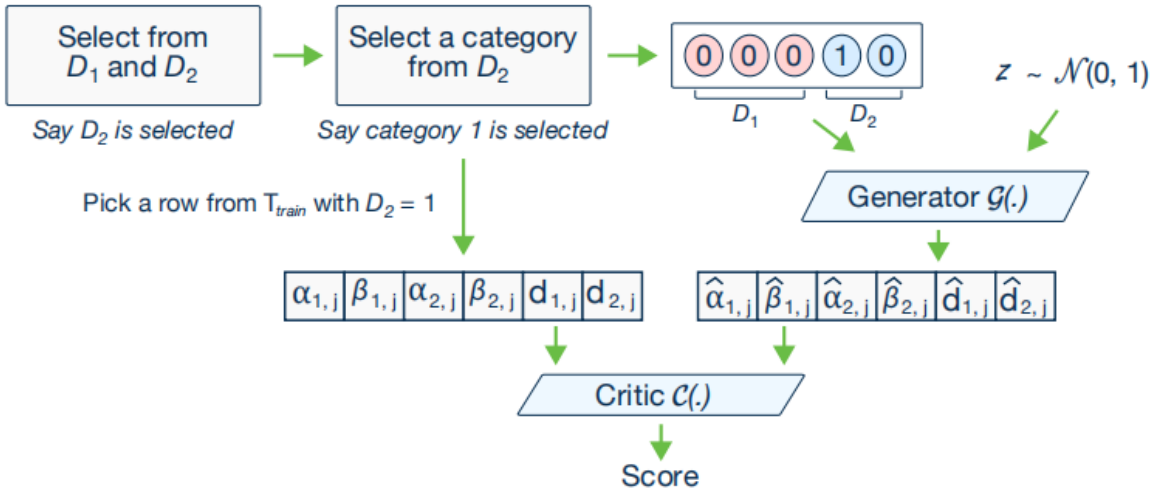


Figure 1: An example of mode-specific normalization.

- 1. 对每一列连续数据，用 VGM 估算高斯混合模型子模型的个数并计算每个子模型的参数；
- 2. 计算每个样本值来自某个子模型的概率；
- 3. 选择概率最大的子模型参数归一化样本值。

CTGAN 算法原理

- 条件生成器（Conditional Generator）
  - Conditional vector



  - Generator loss
    - 添加一项 m_{i*} and d_{i*} 之间的交叉熵鼓励生成器生成和输入 cond 相同的离散数据

CTGAN 算法原理

- 由采样训练（Training-by-sampling）
  - 根据选中列的取值范围构造一个PMF（概率质量函数），以使每个值的概率质量为真实数据的该列中其频率的对数。

- **网络结构：WGAN-GP + PacGAN**

$$
\begin{cases}
h_0 = z \oplus cond \\
h_1 = h_0 \oplus \mathtt{ReLU}(\mathtt{BN}(\mathtt{FC}_{|cond|+|z| \to 256}(h_0))) \\
h_2 = h_1 \oplus \mathtt{ReLU}(\mathtt{BN}(\mathtt{FC}_{|cond|+|z|+256 \to 256}(h_1))) \\
\hat{\alpha}_i = \mathtt{tanh}(\mathtt{FC}_{|cond|+|z|+512 \to 1}(h_2)) & 1 \le i \le N_c \\
\hat{\beta}_i = \mathtt{gumbel}_{0.2}(\mathtt{FC}_{|cond|+|z|+512 \to m_i}(h_2)) & 1 \le i \le N_c \\
\hat{\mathbf{d}}_i = \mathtt{gumbel}_{0.2}(\mathtt{FC}_{|cond|+|z|+512 \to |D_i|}(h_2)) & 1 \le i \le N_d
\end{cases}
$$

$$
\begin{cases}
h_0 = \mathbf{r}_1 \oplus \ldots \oplus \mathbf{r}_{10} \oplus cond_1 \oplus \ldots \oplus cond_{10} \\
h_1 = \mathtt{drop}(\mathtt{leaky}_{0.2}(\mathtt{FC}_{10|\mathbf{r}|+10|cond| \to 256}(h_0))) \\
h_2 = \mathtt{drop}(\mathtt{leaky}_{0.2}(\mathtt{FC}_{256 \to 256}(h_1))) \\
\mathcal{C}(\cdot) = \mathtt{FC}_{256 \to 1}(h_2)
\end{cases}
$$

CTGAN 算法原理

- 总体流程

Algorithm 1: Train CTGAN on step.

Input: Training data $\mathbf{T}_{train}$, Conditional generator and Critic parameters Φ_G and Φ_C respectively, batch size m, pac size pac.

Result: Conditional generator and Critic parameters Φ_G, Φ_C updated.

1 Create masks $\{\mathbf{m}_1, \ldots, \mathbf{m}_{i^*}, \ldots, \mathbf{m}_{N_d}\}_j$, for $1 \leq j \leq m$

2 Create condition vectors $cond_j$, for $1 \leq j \leq m$ from masks ▷ Create m conditional vectors

3 Sample $\{z_j\} \sim \texttt{MVN}(0, \mathbf{I})$, for $1 \leq j \leq m$

4 $\hat{\mathbf{r}}_j \leftarrow \texttt{Generator}(z_j, cond_j)$, for $1 \leq j \leq m$ ▷ Generate fake data

5 Sample $\mathbf{r}_j \sim \texttt{Uniform}(\mathbf{T}_{train}|cond_j)$, for $1 \leq j \leq m$ ▷ Get real data

6 $cond_k^{(pac)} \leftarrow cond_{k\times pac+1} \oplus \ldots \oplus cond_{k\times pac+pac}$, for $1 \leq k \leq m/pac$ ▷ Conditional vector pacs

7 $\hat{\mathbf{r}}_k^{(pac)} \leftarrow \hat{\mathbf{r}}_{k\times pac+1} \oplus \ldots \oplus \hat{\mathbf{r}}_{k\times pac+pac}$, for $1 \leq k \leq m/pac$ ▷ Fake data pacs

8 $\mathbf{r}_k^{(pac)} \leftarrow \mathbf{r}_{k\times pac+1} \oplus \ldots \oplus \mathbf{r}_{k\times pac+pac}$, for $1 \leq k \leq m/pac$ ▷ Real data pacs

9 $\mathcal{L}_C \leftarrow \frac{1}{m/pac}\sum_{k=1}^{m/pac} \texttt{Critic}(\hat{\mathbf{r}}_k^{(pac)}, cond_k^{(pac)}) - \frac{1}{m/pac}\sum_{k=1}^{m/pac} \texttt{Critic}(\mathbf{r}_k^{(pac)}, cond_k^{(pac)})$

10 Sample $\rho_1, \ldots, \rho_{m/pac} \sim \texttt{Uniform}(0, 1)$

11 $\tilde{\mathbf{r}}_k^{(pac)} \leftarrow \rho_k \hat{\mathbf{r}}_k^{(pac)} + (1-\rho_k)\mathbf{r}_k^{(pac)}$, for $1 \leq k \leq m/pac$

12 $\mathcal{L}_{GP} \leftarrow \frac{1}{m/pac}\sum_{k=1}^{m/pac}(||\nabla_{\tilde{\mathbf{r}}_k^{(pac)}} \texttt{Critic}(\tilde{\mathbf{r}}_k^{(pac)}, cond_k^{(pac)})||_2 - 1)^2$ ▷ Gradient Penalty

13 $\Phi_C \leftarrow \Phi_C - 0.0002 \times \texttt{Adam}(\nabla_{\Phi_C}(\mathcal{L}_C + 10\mathcal{L}_{GP}))$

14 Regenerate $\hat{\mathbf{r}}_j$ following lines 1 to 7

15 $\mathcal{L}_G \leftarrow -\frac{1}{m/pac}\sum_{k=1}^{m/pac} \texttt{Critic}(\hat{\mathbf{r}}_k^{(pac)}, cond_k^{(pac)}) + \frac{1}{m}\sum_{j=1}^{m} \texttt{CrossEntropy}(\hat{\mathbf{d}}_{i^*,j}, \mathbf{m}_{i^*})$

16 $\Phi_G \leftarrow \Phi_G - 0.0002 \times \texttt{Adam}(\nabla_{\Phi_G}\mathcal{L}_G)$

CTGAN 算法原理

- 实验结果（ Experiments Result ）

method	adult F1	census F1	credit F1	cover. Macro	intru. Macro	mnist12/28 Acc	mnist12/28 Acc	news R^2
Identity	0.669	0.494	0.720	0.652	0.862	0.886	0.916	0.14
CLBN(3)	0.334	0.310	0.409	0.319	0.384	0.741	0.176	-6.28
PrivBN(4)	0.414	0.121	0.185	0.270	0.384	0.117	0.081	-4.49
MedGAN(6)	0.375	0.000	0.000	0.093	0.299	0.091	0.104	-8.80
VEEGAN(6)	0.235	0.094	0.000	0.082	0.261	0.194	0.136	-6.5e6
TableGAN(5)	0.492	0.358	0.182	0.000	0.000	0.100	0.000	-3.09
TVAE(1)	**0.626**	0.377	0.098	**0.433**	0.511	**0.793**	**0.794**	**-0.20**
TGAN(1)	0.601	**0.391**	**0.672**	0.324	**0.528**	0.394	0.371	-0.43

TVAE outperforms CTGAN in several cases, but GANs do have several favorable attributes, and this does not indicate that we should always use VAEs rather than GANs to model tables. The generator in GANs does not have access to real data during the entire training process; thus, we can make CTGAN achieve differential privacy [14] easier than TVAE.

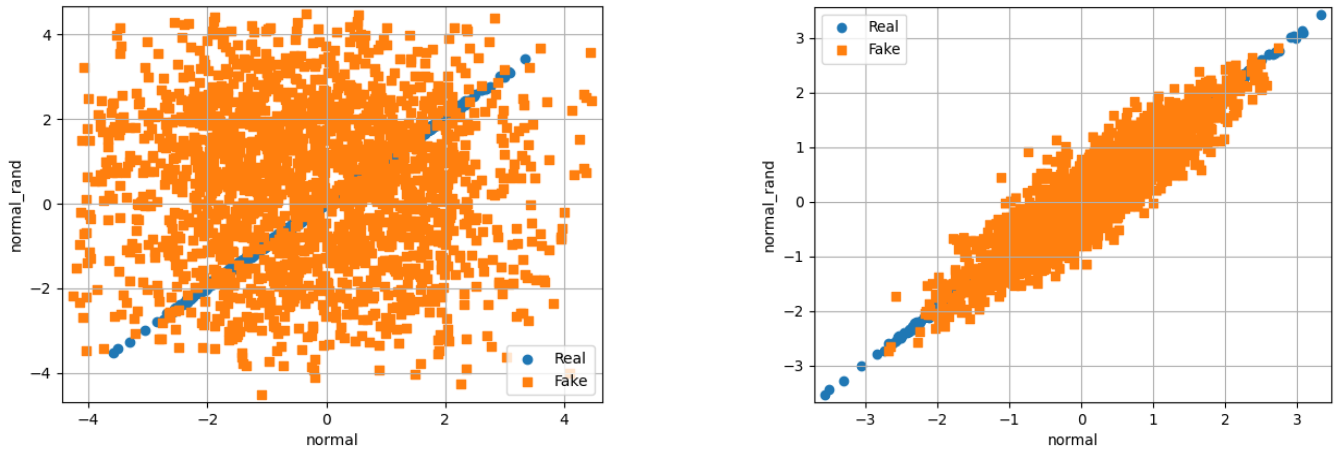
CTGAN 算法原理

- **论文复现**
  - https://github.com/sdv-dev/CTGAN

```
# pip install ctgan or pip install -e CTGAN

from ctgan import CTGANSynthesizer

discrete_columns = [
    ...
]

ctgan = CTGANSynthesizer()
ctgan.fit(data, discrete_columns)
samples = ctgan.sample(1000)
```

优劣分析

- 对于连续数据，CTGAN 利用一个高维的 one-hot 和一维归一化后的数据表示原始数据中的一维数据，会令 GAN 更加难以学到原始数据各维度之间的关系



- CTGAN 没有真正解决训练数据不平衡造成生成数据真实性较低的问题

参考文献

- XU L, SKOULARIDOU M, CUESTA-INFANTE A等. Modeling Tabular data using Conditional GAN[J/OL]. 2019(NeurIPS).
- Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In International Conference on Machine Learning, 2017.

知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者，寿。

谢谢！