# Conditional Sig-Wasserstein GANs for Time Series Generation

## 1 GAN WGAN Sig-WGAN

### 1.1 GAN

原始GAN公式：

$$\min_G \max_D V(D,G)=\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]+\mathbb{E}_{\mathbf{z}\sim p_{\mathbf{Z}}(\mathbf{z})}[\log(1-D(G(\mathbf{z})))]$$

交替优化：先训练k次D，再保持D不变训练1次G。

**图表2： GAN 训练算法的伪代码**

**输入：迭代次数 T，每轮迭代判别器 D 训练次数 K，小批量（minibatch）样本数量 m**

1 随机初始化 D 网络参数 $\theta_d$ 和 G 网络参数 $\theta_g$

2 for t ← 1 to T do

  # 训练判别器 D

3   for k ← 1 to K do

    # 采集小批量样本

4     从标准正态分布 $p_g(z)$ 中采集 m 条样本 $\{z^{(m)}\}$

5     从训练集 $p_{data}(x)$ 中采集 m 条样本 $\{x^{(m)}\}$

6     使用随机梯度上升更新判别器 D，梯度为：

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

7   end

  # 训练生成器 G

8   从标准正态分布 $p_g(z)$ 中采集 m 条样本 $\{z^{(m)}\}$

9   使用随机梯度上升更新生成器 G，梯度为：

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

10 end

**输出：生成器 G**

给定$G$后，$D$的最优解：$D_G^*(x)=\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$

$G$的目标函数：

$$C(G)=-\log 4+2JS\left(p_{\text{data}}\|p_g\right)$$

生成器网络：两个隐藏层的全连接神经网络。

判别器网络：三个卷积层一个全连接层的卷积神经网络

GAN的不足：

- G和D训练不同步。如果D训练不够，G也很难提高;若D训练太好，G容易梯度消失。

  原因：若D训练到最优，G的损失函数是$C(G)=-\log 4+2JS\left(p_r\|p_g\right)$，

  大部分生成分布和真实分布支撑集相交部分的测度为零，JS测度恒为常数$log2$，$C(G)$为常数，故梯度为零，出现梯度消失问题。

- 训练不收敛。G和D处于博弈状态，一方增大，一方减小，两者的损失函数此消彼长，不收敛。只能通过观察生成样本的好坏判断训练是否充分，缺少辅助指标。

- 模式崩溃。GAN生成样本单一，缺乏多样性，生成序列与真实序列十分相近，但不包含市场可能出现的各种情况。

  **Non-saturating GAN**:

  早期$log(1-D(G(z)))$接近0，导数在0附近变化较小，不利于梯度下降，

  将G的优化目标改为$-\mathrm{E}_{\mathrm{z}\sim\mathrm{pz(z)}}[\log \mathrm{D}(\mathrm{G}(\mathrm{z}))]$，此时G和D的损失函数分别为：

  判别器：$J(D)=E_{z\sim p_Z}[\log D(G(z))]-E_{x\sim p_r}[\log(D(x))]$

  生成器：$J(G)=-E_{z\sim p_z}[\log D(G(z))]$

  当D最优时，

$$J(G)=KL\left(p_g\|p_r\right)-2JS\left(p_r\|p_g\right)$$

  模式崩溃原因：

$$KL\left(p_g\|p_r\right)=\int_x p_g(x)\log\frac{p_g(x)}{p_r(x)}dx$$

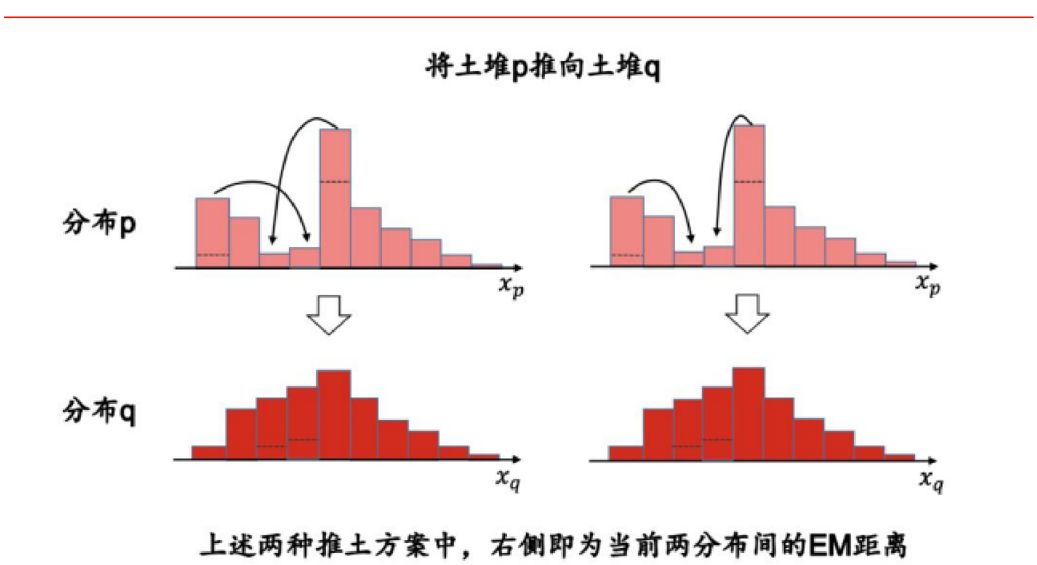
  1. 若生成不真实样本$x$，$p_g(x)>0, p_r(x)\approx 0$,被积项趋于$\infty$
  2. 若不生成真实样本，对于那些没能生成的真实样本$y$,$p_r(y)>0, p_g(x)\approx 0$，被积项趋于0

  优化生成器要求KL散度变小，那么生成器倾向于第2种，避免第1种。所以生成器会生成单一的生成样本。

## 1.2 WGAN

GAN的大部分缺陷和JS散度有关，故用W距离代替JS散度。

### 1.2.1 W距离



“推土”角度，EM距离：

$$W(p,q)=\min_{\gamma\in\Pi}\sum_{x_p,x_q}\gamma\left(x_p,x_q\right)\|x_p-x_q\|$$

$\gamma\left(x_p,x_q\right)$是推土量。

概率分布角度：

$$W\left(p_r,p_g\right)=\inf_{\gamma\sim n\left(p_r,p_g\right)}E_{(x,y)\sim\gamma}[\|x-y\|]$$

$x\sim p_r, y\sim p_g, \gamma$ 表示$(x,y)$ 的联合分布.

W距离和JS散度的区别：当真实分布和生成分布的支撑集相交部分测度为零，JS恒为常数，无法衡量距离；W距离变化是连续的，可以同时衡量距离和概率差异。

### 1.2.2 WGAN原理

根据Kantorovich Rubinstein Duality公式，W距离等价于：

$$\begin{aligned}W\left(p_r,p_g\right)&=\sup_{w:\|f_w\|_L\leq 1}\left(E_{x\sim p_r}\left[f_w(x)\right]-E_{x\sim p_g}\left[f_w(x)\right]\right)\\&=\sup_{w:\|f_w\|_L\leq 1}\left(E_{x\sim p_r}\left[f_w(x)\right]-E_{z\sim p_z}\left[f_w(G(z))\right]\right)\end{aligned}$$

把$f_w$看成是“判别器”，用一个深度学习网络来代替。

二者的损失函数：

判别器: $J(D)=E_{z\sim P_z}\left[f_w(G(z))\right]-E_{x\sim P_r}\left[f_w(x)\right]$

生成器: $J(G)=-E_{z\sim P_z}\left[f_w(G(z))\right]$

“判别器”要满足Lipschitz条件，用权重剪裁或梯度惩罚处理，思想都是对权重和梯度进行限制，让f近似满足Lipschitz条件。

### 1.2.3 GAN和WGAN的比较

针对GAN的三项缺点的改进。

- 训练不同步。GAN先训练k次D，再训练1次G，要调整K的值；WGAN无需调整K,因为不会出现梯度消失问题。

- GAN中D和G的损失函数都不收敛，无法指示训练进程；WGAN中判别器的损失函数是真假样本分布的W距离，可以作为辅助指标。

- 模式崩溃。GAN模式崩溃和KL散度和JS散度有关，W距离没有此类问题。

## 1.3 SigCWGAN

### 1.3.1 重要概念

- 路径签名

  Signature of X: $S(X_J) = \left(1, X_J^1, \cdots, X_J^k, \cdots\right) \in T\left(\left(\mathbb{R}^d\right)\right)$, $X_J^k = \int_{t_1 < t_2 < \cdots < t_k, t_1, \cdots, t_k \in J} dX_{t_1} \otimes \cdots dX_{t_k}$

- 签名的特性

  University: non-linear continuous function $f$ be approximated by linear functional $L$ in signature space.

$$sup_{X \in K} |f(X) - < L, S_M(X) > | < \epsilon$$

- 签名的阶乘衰退

  (Factorial Decay of the Signature):

$$|\pi_m(S(X))| \leq \frac{|X|_{1-var}^m}{m!}$$

### 1.3.2 距离度量变种

- The Kantorovich and Rubinstein dual representation of $W_1$ is:

$$W_1(\mu, \nu) = \sup \left\{ \int f(x) d(\mu - \nu)(x) \right\} \mid \text{ continuous } f : \mathcal{X} \to \mathbb{R}, \operatorname{Lip}(f) \leq 1\}$$

- From $W_1$ and **universality**, consider:

$$Sig - W_1(\mu, \nu) := \sup_{|L| \leq 1, L \text{ is a linear functional}} L\left(\mathbb{E}_\mu[S(X)] - \mathbb{E}_\nu[S(X)]\right)$$

  reduce **nonlinear** optimization of computing $W_1$ distance over the class of Lipschitz functionals to **linear** functions on the signature space.

- due to factorial decay of signature, approximate $Sig - W_1(\mu, \nu)$ using truncated signature:

$$Sig - W_1^{(M)}(\mu, \nu) := \sup_{|L| \leq 1, L \text{ is a linear functional}} L\left(\mathbb{E}_\mu\left[S_M(X)\right] - \mathbb{E}_\nu\left[S_M(X)\right]\right)$$

- the norm of $L$ is chosen as $l_2$ norm of the linear coefficients of $L$, the optimization admits analytic solution:

$$Sig - W_1^{(M)}(\mu, \nu) = |\mathbb{E}_\mu\left[S_M(X)\right] - \mathbb{E}_\nu\left[S_M(X)\right]|$$

### 1.3.3 SigCWGAN

- The Conditional AR-FNN Generator

  之前的(W)GAN是用G来捕捉样本的整体分布，从白噪声$z$生成$G(z)$ ，这种方法没有考虑到样本是时间序列。

  所以考虑时间序列，用Conditional AR-FNN generator $G^{\theta}:\mathbb{R}^{d\times p}\times\mathcal{Z}\rightarrow\mathbb{R}^{d}$:

  - take $X_{t-p+1:t}=x\in\mathbb{R}^{d\times p}$ and normal distributed noise $Z_{t+1}\in\mathcal{Z}$ to generate $X_{t+1}\sim\mathbb{P}\left(X_{t+1}\mid X_{t-p+1:t}=x\right)$
  - $G^{\theta}$: feedforward neural network, residual connections and RelUs
  - Algorithm 1: $\hat{X}_{t+1}^{(t)}=G^{\theta}\left(X_{t-p+1:t},Z_{t+1}\right)\rightarrow\hat{X}_{t+2}^{(t)}=G^{\theta}\left(X_{t-p+2:t},\hat{X}_{t+1}^{(t)},Z_{t+2}\right)\ldots\rightarrow\hat{X}_{t+1:t+q}^{(t)}$
  - designed to capture the autoregressive structure (temporal dependence) of time series

- The Conditional Sig-$W_1$ Discriminator

  quantify the distance between $\mu(X_{future}|x_{past})$ and $\nu(X_{future}|x_{past})$

  $$C-Sig-W_{1}^{(M)}(\nu,\mu):=\mid\mathbb{E}_{\nu}\left[S_{M}\left(X_{\text{future}}\right)\mid x_{\text{past}}=x\right]-\mathbb{E}_{\mu}\left[S_{M}\left(X_{\text{future}}\right)\mid x_{\text{past}}=x\right]$$

  loss function is sum of error between true path and generated path:

  $$L(\theta)=\sum_{t}\left|\mathbb{E}_{\nu}\left[S_{M}\left(X_{t+1:t+q}\right)\mid X_{t-p+1:t}\right]-\mathbb{E}_{\mu}\left[S_{M}\left(\hat{X}_{t+1:t+q}^{(t)}\right)\mid X_{t-p+1:t}\right]\right|$$

  $\nu\rightarrow$ real distribution, $\mu\rightarrow$ synethetic generator

  相对比与GAN和WGAN, D的损失函数不仅仅是两个分布的距离，而是分布距离关于时间的序列误差和。

- SigCWGAN Algorithm

  如何估计$\mathbb{E}_{\nu}\left[S_{M}\left(X_{t+1:t+q}\right)\mid X_{t-p+1:t}\right]$和$\mathbb{E}_{\mu}\left[S_{M}\left(\hat{X}_{t+1:t+q}^{(t)}\right)\mid X_{t-p+1:t}\right]$?

  - $\mathbb{E}_{\nu}\left[S_{M}\left(X_{t+1:t+q}\right)\mid X_{t-p+1:t}\right]$
    1. based on autoregressive assumption of X,
       $\mathbb{E}_{\nu}\left[S_{M}\left(X_{t+1:t+q}\right)\mid X_{t-p+1:t}\right]$ does not depend on t
       $\rightarrow$ use supervised learning algorithm to learn from true data.
    2. 根据university,可以apply linear regression on $\left(S_{N}\left(X_{t-p+1:t}\right),S_{M}\left(X_{t+1:t+q}\right)\right)_{t}$,
       obtain $\hat{L}(S_{N}\left(X_{t-p+1:t}\right))$ as estimator for $\mathbb{E}_{\nu}\left[S_{M}\left(X_{t+1:t+q}\right)\mid X_{t-p+1:t}\right]$
  - $\mathbb{E}_{\mu}\left[S_{M}\left(\hat{X}_{t+1:t+q}^{(t)}\right)\mid X_{t-p+1:t}\right]$
    1. Given $X_{t-p+1:t}$, sample noise $Z_{t+1:t+q}$ and generate $\hat{X}_{t+1:t+q}^{(t)}$ by $G^{\theta}$

2. Monte-Carlo method to get estimator for $\mathbb{E}_\mu\left[S_M\left(\hat{X}^{(t)}_{t+1:t+q}\right) \mid X_{t-p+1:t}\right]$

SGD algorithm to update parameters of generator $G^\theta$

生成器是$G^\theta$，判别器是$L$，但是在分析$\mathrm{Sig}-W_1^{(M)}(\mu,\nu)$的时候得到了一个nanlytic solution，所以不需要对判别器进行梯度下降。用university的函数近似误差代替对判别器的近似求解误差

所以该算法的一个特定是不用交替优化，避免了互相博弈的过程。用损失函数直接作为进程辅助指标