A Two-Step Computation of the Exact GAN Wasserstein Distance

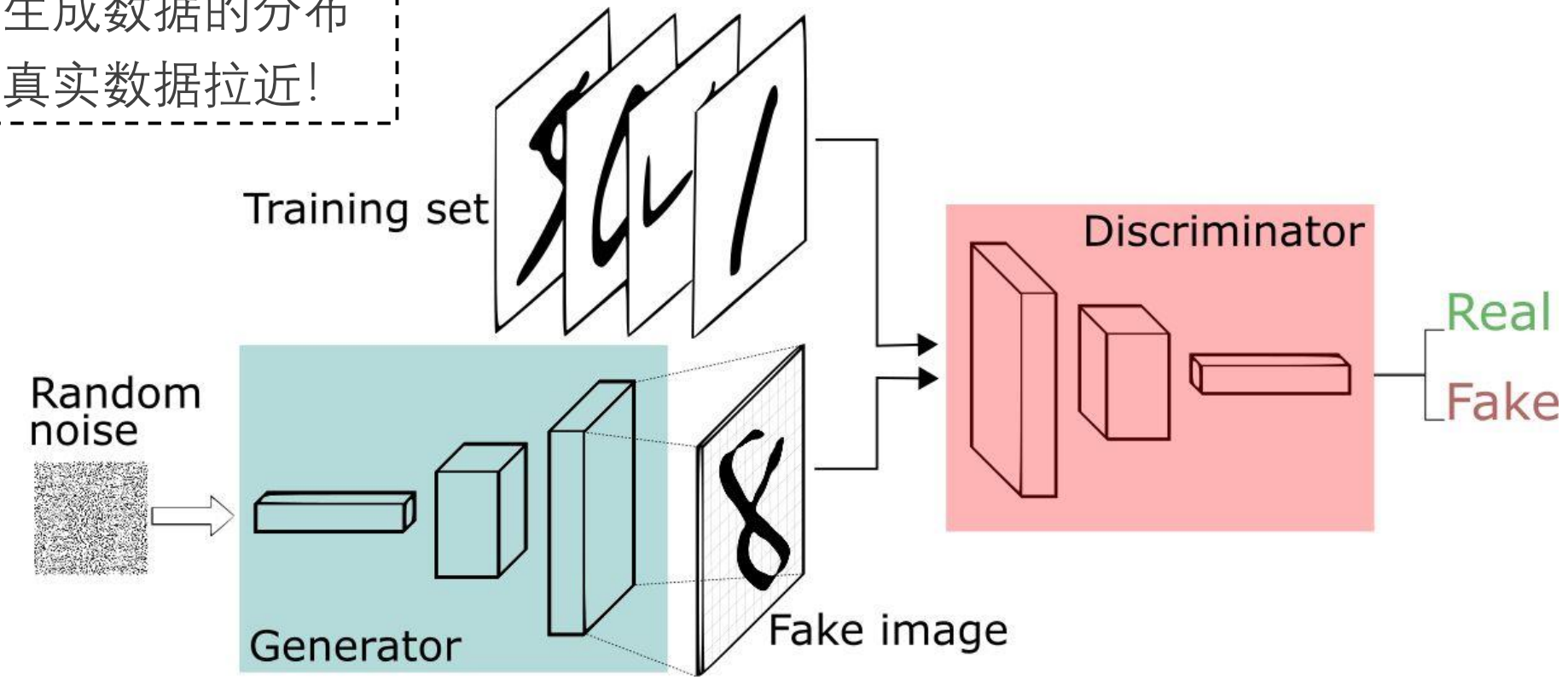
Huidong Liu [1] **Xianfeng Gu** [1] **Dimitris Samaras** [1]

[1]Department of Computer Science, Stony Brook University, New York, USA. Correspondence to: Huidong Liu <huidliu@cs.stonybrook.edu>, Xianfeng Gu <gu@cs.stonybrook.edu>, Dimitris Samaras <samaras@cs.stonybrook.edu>.

目录

GAN

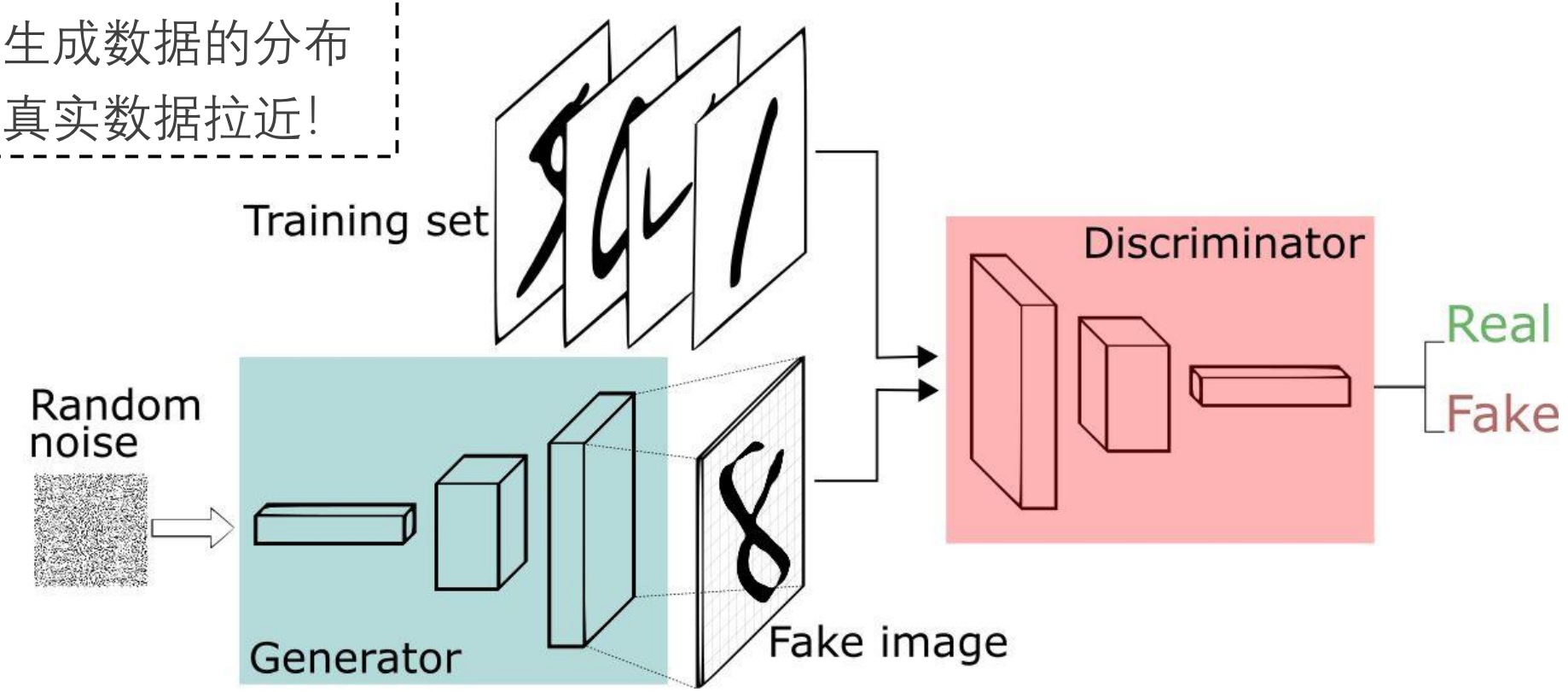
- 将生成数据的分布向真实数据拉近!



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

GAN

- 将生成数据的分布向真实数据拉近!



Discriminator

Domain label: 1

Domain label: 0

$$\min_{G} \max_{D} V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Generator

Distribution of real data

Distribution of fake data

Distance of Distributions

Let $(\chi, \sum)$ *be a measurable space.*

- The *Total Variation* (TV) distance

$$\delta(\mathbb{P}_r, \mathbb{P}_g) = \sup_{A \in \Sigma} |\mathbb{P}_r(A) - \mathbb{P}_g(A)| \ .$$

when $\mathcal{X}$ *is a finite space, we have that* $\delta(\mathbb{P}, \mathbb{Q}) = \max_{x \in \mathcal{X}} |\mathbb{P}(x) - \mathbb{Q}(x)|$.

- The *Kullback-Leibler* (KL) divergence

$$KL(\mathbb{P}_r \| \mathbb{P}_g) = \int \log \left(\frac{P_r(x)}{P_g(x)} \right) P_r(x) d\mu(x) \ ,$$

- The *Jensen-Shannon* (JS) divergence

$$JS(\mathbb{P}_r, \mathbb{P}_g) = KL(\mathbb{P}_r \| \mathbb{P}_m) + KL(\mathbb{P}_g \| \mathbb{P}_m) \ ,$$

GAN

- 固定G时，可得此时最优判别器为

$$V(D,G)=\int_x \left[P_r(x)\log(D(x))+P_f(x)\log(1-D(x))\right]dx$$

$$\frac{\partial V}{\partial D}=0\text{，即可得到 }D_G^*(x)=\frac{P_r(x)}{P_r(x)+P_f(x)}$$

- 将D*带回原式，发现生成器G实际上是在最小化**JS散度**

$$\begin{aligned}
&=\mathbb{E}_{x\sim P_r}\left[\log\frac{P_r(x)}{P_r(x)+P_f(x)}\right]+\mathbb{E}_{x\sim P_f}\left[\log\frac{P_f(x)}{P_r(x)+P_f(x)}\right]\\
&=-log(4)+KL\left(P_r||\frac{P_r+P_f}{2}\right)+KL\left(P_f||\frac{P_r+P_f}{2}\right)\\
&=-log(4)+2JS(P_r||P_f).
\end{aligned}$$

Problems

Example 1 (Learning parallel lines)**.** Let $Z \sim U[0,1]$ the uniform distribution on the unit interval. Let $\mathbb{P}_0$ be the distribution of $(0, Z) \in \mathbb{R}^2$ (a 0 on the x-axis and the random variable Z on the y-axis), uniform on a straight vertical line passing through the origin. Now let $g_\theta(z) = (\theta, z)$ with θ a single real parameter. It is easy to see that in this case,

$$p_\theta(x,y) = \begin{cases} 1 & x = \theta \text{ and } 0 \le y \le 1, \\ 0 & \text{else.} \end{cases}$$

- $JS(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ , \end{cases}$
- $KL(\mathbb{P}_\theta \| \mathbb{P}_0) = KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = \begin{cases} +\infty & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ , \end{cases}$
- and $\delta(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} 1 & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ . \end{cases}$

Can not properly estimate the difference between two nonoverlapping distributions!

- Total variation: For any $\theta \neq 0$, let $A = \{(0, y) : y \in [0, 1]\}$. This gives

$$\delta(P_0, P_\theta) = \begin{cases} 1 & \text{if } \theta \neq 0\,, \\ 0 & \text{if } \theta = 0\,. \end{cases}$$

- KL divergence and reverse KL divergence: Recall that the KL divergence $KL(P\|Q)$ is $+\infty$ if there is any point (x, y) where $P(x, y) > 0$ and $Q(x, y) = 0$. For $KL(P_0\|P_\theta)$, this is true at $(\theta, 0.5)$. For $KL(P_\theta\|P_0)$, this is true at $(0, 0.5)$.

$$KL(P_0\|P_\theta) = KL(P_\theta\|P_0) = \begin{cases} +\infty & \text{if } \theta \neq 0\,, \\ 0 & \text{if } \theta = 0\,, \end{cases}$$

- Jenson-Shannon divergence: Consider the mixture $M = P_0/2 + P_\theta/2$, and now look at just one of the KL terms.

$$KL(P_0\|M) = \int_{(x,y)} P_0(x, y) \log \frac{P_0(x, y)}{M(x, y)}\, dy\, dx$$

For any x, y where $P_0(x, y) \neq 0$, $M(x, y) = \frac{1}{2}P_0(x, y)$, so this integral works out to $\log 2$. The same is true of $KL(P_\theta\|M)$, so the JS divergence is

$$JS(P_0, P_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0\,, \\ 0 & \text{if } \theta = 0\,, \end{cases}$$

目录

1-Wasserstein Distance

Problem 5. *Find a transport plan to minimize the following total cost:*

$$C(\pi^*) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{X \times Y} c(x,y) d\pi(x,y) \qquad (9)$$

where $\Pi(\mu,\nu)$ *is the set of transport plans defined as:*

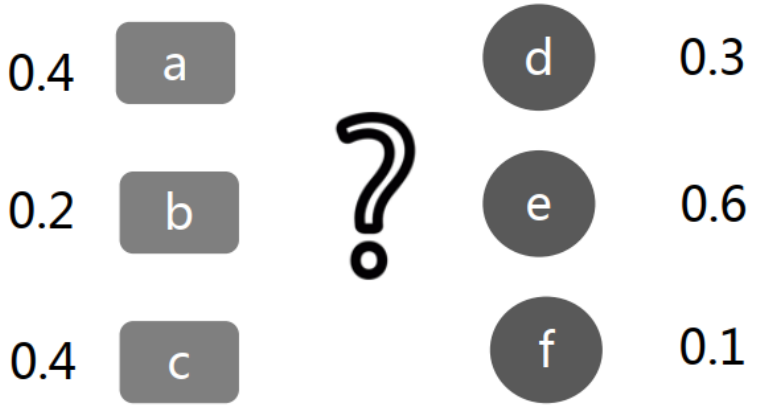
$$\Pi(\mu,\nu) = \{\pi \in \mathbb{P}(X \times Y) : \pi_x = \mu, \pi_y = \nu\} \qquad (10)$$

where π_x *and* π_y *are marginal distributions of* π *on* X *and* Y*, respectively.*

$c(x,y)$ *is a cost function, which can be implemented by* ℓ_1 *norm,* ℓ_2 *norn, etc.*

1-Wasserstein Distance (Earth-Mover Distance)

Intuitively, $\gamma(x, y)$indicates how much "mass" must be transported from x to y in order to transform the distributions P_r into the distribution P_r. The EM distance is the "cost" of the optimal transport plan.



$d(s,t)$ $m(s,t)$

$$min \sum_{\pi(s,t)} d(s,t) * m(s,t)$$

$$\sum_{s} m(s,t) = b(t)\ \forall t$$

$$\sum_{t} m(s,t) = a(s)\ \ \forall s$$

1-Wasserstein Distance

Example 1 (Learning parallel lines). Let $Z \sim U[0,1]$ the uniform distribution on the unit interval. Let $\mathbb{P}_0$ be the distribution of $(0, Z) \in \mathbb{R}^2$ (a 0 on the x-axis and the random variable Z on the y-axis), uniform on a straight vertical line passing through the origin. Now let $g_\theta(z) = (\theta, z)$ with θ a single real parameter. It is easy to see that in this case,

- $W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|$, (Take ℓ_2 norm as the cost function)

- Earth Mover distance: Because the two distributions are just translations of one another, the best way transport plan moves mass in a straight line from $(0, y)$ to (θ, y). This gives $W(P_0, P_\theta) = |\theta|$

Wasserstein distance can estimate the difference between two nonoverlapping distributions

目录

Dual Form

- The Monge-Kantorovich dual problem*

Problem 2. *Find a function ψ such that*

$$C(\mu,\nu) = \sup_{\psi} \left\{ \int \psi^c(y) d\nu(y) - \int \psi(x) d\mu(x) \right\} \quad (2)$$

where $C(\mu,\nu)$ is the Wasserstein distance between μ and ν and ψ^c is the c-transform of ψ defined below:

$$\forall y \in Y \qquad \psi^c(y) = \inf_{x \in X} \big(\psi(x) + c(x,y) \big) \quad (3)$$

* Villani, C. Optimal transport: old and new, volume 338. Springer Science & Business Media, 2008

Dual Form

- Discrete form

$\hat{X} = \{x_j\}_{j \in \mathcal{J}}$ sampled from μ and $\hat{Y} = \{y_i\}_{i \in \mathcal{I}}$ sampled from ν, where $\mathcal{I}$ and $\mathcal{J}$ are disjoint index sets. Let $m = |\mathcal{I}|$ and $n = |\mathcal{J}|$ be the numbers of elements in the two sets.

Problem 3. *(Discrete Case of Problem 2) Let*

$$\hat{d}(\psi) = \frac{1}{m} \sum_{i \in \mathcal{I}} \psi^c(y_i) - \frac{1}{n} \sum_{j \in \mathcal{J}} \psi(x_j) \tag{4}$$

Find a function ψ such that $\hat{C}(\mu, \nu) = \sup_\psi \hat{d}(\psi)$ where $\hat{C}(\mu, \nu)$ is the Wasserstein distance between μ and ν and ψ^c is the c-transform of ψ defined below:

$$\forall y_i \in \hat{Y} \qquad \psi^c(y_i) = \inf_{x \in \hat{X}} \left(\psi(x) + c(x, y_i) \right) \tag{5}$$

WGAN （Kantorovich-Rubinstein duality)

In order to make $\psi^c = \psi$ to simplify this problem, the WGAN restricts function ψ to be 1-Lipschitz (Arjovsky

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta}[f(x)]$$

DEFINITION 12.6 (Lipschitzness) Let $C \subset \mathbb{R}^d$. A function $f : \mathbb{R}^d \to \mathbb{R}^k$ is ρ-Lipschitz over C if for every $\mathbf{w}_1, \mathbf{w}_2 \in C$ we have that $\|f(\mathbf{w}_1) - f(\mathbf{w}_2)\| \leq \rho \|\mathbf{w}_1 - \mathbf{w}_2\|$.

Arjovsky, Martín, Soumith Chintala, and Léon Bottou. "Wasserstein Generative Adversarial Networks." *ICML*. 2017.

目录

A New Formulation (main contribution)

- **Without making the Lipschitz continuous assumption**

Problem 4. *Solve the following problem:*

$$\begin{aligned} \max_{f} \quad & \hat{h}(f) = \left\{ \frac{1}{m} \sum_{i \in \mathcal{I}} f(y_i) - \frac{1}{n} \sum_{j \in \mathcal{J}} f(x_j) \right\} \\ \text{s.t.} \quad & f(y_i) - f(x_j) \leq c(x_j, y_i), \ \ \forall j \in \mathcal{J}, \ \forall i \in \mathcal{I} \end{aligned}$$

Solving Problem 4 is equivalent to solving Problem 3 under a mild assumption that the cost function $c(\cdot, \cdot)$ satisfies the triangle inequality in Lemma 3.1.

Lemma 3.1. *If the cost function $c(\cdot,\cdot)$ satisfies the triangle inequality, i.e., $c(x,y)+c(y,z)\geq c(x,z), \forall x,y,z$, then $\forall x_j \in \hat{X}$, $\forall y_i \in \hat{Y}$, if $x_j = y_i$, and ψ^* is the optimizer to Problem 3, then $(\psi^c)^*(y_i) = \psi^*(x_j)$, where $(\psi^c)^*(y_i) = \inf_{x\in\hat{X}}\left(\psi^*(x)+c(x,y_i)\right)$.*

Proof. We prove this by contradiction. Without loss of generality, suppose x_s overlaps with y_t, i.e., $x_s = y_t$, and $(\psi^c)^*(y_1) \neq \psi^*(x_s)$. According to the definition of the c-transform in Eq. (5), $(\psi^c)^*(y_t) = \inf\{\psi^*(x_s), \inf_{x\in\hat{X}\setminus x_s}\psi^*(x)+c(x,y_t)\}$. Since $(\psi^c)^*(y_t)\neq\psi^*(x_s)$, for any y_i, we have

$$
\begin{aligned}
& \psi^*(x_s)+c(x_s,y_i) \\
> & \inf_{x\in\hat{X}\setminus x_s}\left(\psi^*(x)+c(x,y_t)\right)+c(x_s,y_i) \\
= & \inf_{x\in\hat{X}\setminus x_1}\left(\psi^*(x)+c(x,y_t)+c(x_s,y_i)\right) \\
\geq & \inf_{x\in\hat{X}\setminus x_s}\left(\psi^*(x)+c(x,y_i)\right)
\end{aligned}
$$

Proof

- **In this case**

$$
\begin{aligned}
& \hat{C}^*(\mu,\nu) \\
= \quad & \frac{1}{m}\sum_{i\in\mathcal{I}}(\psi^c)^*(y_i) - \frac{1}{n}\sum_{j\in\mathcal{J}}\psi^*(x_j) \\
= \quad & \frac{1}{m}\sum_{i\in\mathcal{I}}\inf_{x\in\hat{X}\setminus x_s}(\psi^*(x)+c(x,y_i)) - \frac{1}{n}\sum_{j\in\mathcal{J}}\psi^*(x_j)
\end{aligned}
$$

We can always find another function ψ', such that $\psi'(x) = \psi^*(x), \forall x \in \hat{X} \setminus x_s$, and $\psi^*(x_s) > \psi'(x_s) > \inf_{x\in\hat{X}\setminus x_s}(\psi^*(x)+c(x,y_t))$. In this case $(\psi^c)'(y_i) = (\psi^c)^*(y_i), \forall i \in \mathcal{I}$, but $\psi^*(x_s) > \psi'(x_s)$. So, $\hat{d}(\psi') > \hat{d}(\psi^*)$, a contradiction. □

Solving the Monge-Kantorovich Dual Formulation

Problem 4. *Solve the following problem:*

$$\begin{aligned} \max_{f} \quad & \hat{h}(f) = \left\{ \frac{1}{m} \sum_{i \in \mathcal{I}} f(y_i) - \frac{1}{n} \sum_{j \in \mathcal{J}} f(x_j) \right\} \\ \text{s.t.} \quad & f(y_i) - f(x_j) \leq c(x_j, y_i), \ \ \forall j \in \mathcal{J}, \ \forall i \in \mathcal{I} \end{aligned}$$

Directly solving Problem 4 is difficult

$$\begin{aligned} \max_{T} \quad & \frac{1}{m} \sum_{i \in \mathcal{I}} T_i - \frac{1}{n} \sum_{j \in \mathcal{J}} T_j \\ \text{s.t.} \quad & T_i - T_j \leq c_{ij}, \ \ \forall i \in \mathcal{I}, \ \forall j \in \mathcal{J} \end{aligned} \tag{7}$$

A linear programming

Solving the Monge-Kantorovich Dual Formulation

Step 2: After solving the linear programming problem, we optimize the following regression problem:

$$\min_{f} \quad \frac{1}{m+n}\left(\sum_{i\in\mathcal{I}}\left(f(y_i)-T_i^*\right)^2+\sum_{j\in\mathcal{J}}\left(f(x_j)-T_j^*\right)^2\right) \tag{8}$$

Why need a continuous function?

Objective Function

$$\min_{G} \max_{D} \quad \hat{C}(f) = \left\{ \frac{1}{m} \sum_{i \in \mathcal{I}} D(y_i) - \frac{1}{n} \sum_{j \in \mathcal{J}} D(G(z_j)) \right\}$$

$$\text{s.t.} \qquad D(y_i) - D(G(z_j)) \leq c(y_i, G(z_j)), \quad \forall i, \ \forall j \tag{13}$$

where $c(y_i, G(z_j)) = ||y_i - G(z_j)||_1$ (ℓ_1 distance) or $c(y_i, G(z_j)) = ||y_i - G(z_j)||_2$ (ℓ_2 distance).

Optimize D: **Using the proposed two-stage method**

Optimize G: $$\min_{G} \quad -\frac{1}{n} \sum_{j \in \mathcal{J}} D(G(z_j))$$

Algorithm

Algorithm 1 WGAN-TS

Input: Real data Y, batch size m, $n_c = 1$, $n_r = 5$, Adam parameters, α, β_1, β_2
Output: G, D
while θ has not converged **do**
  for $t_c = 0$ **to** n_c **do**
    Sample $\{y_i\}_{i \in \mathcal{I}} \sim \mathbb{P}_r$ from real data.
    Sample $\{z_j\}_{j \in \mathcal{J}} \sim \mathbb{P}_z$ random noises.
    Let $x_j = G(z_j), \forall j \in \mathcal{J}$.
    Solve the Linear Programming problem in Eq. (7) using $c_{ij} = ||y_i - x_j||_1$, and obtain T^*.
    $T_t^* \leftarrow T_t^* - (\sum_{k \in \mathcal{I} \cup \mathcal{J}} T_k^*)/(m+n), \forall t \in \mathcal{I} \cup \mathcal{J}$.
    for $t_r = 0$ **to** n_r **do**
      $g_w \leftarrow \nabla_w \frac{1}{m+n} (\sum_{i \in \mathcal{I}} (D_w(y_i) - T_i^*)^2 + \sum_{j \in \mathcal{J}} (D_w(x_j) - T_j^*)^2)$
      $w \leftarrow \text{Adam}(g_w, w, \alpha, \beta_1, \beta_2)$
    end for
    Perform weight scaling on D according to Eq. (15)
  end for
  $g_\theta \leftarrow \nabla_\theta - \frac{1}{n} \sum_{j \in \mathcal{J}} D(G_\theta(z_j))$
  $\theta \leftarrow \text{Adam}(g_\theta, \theta, \alpha, \beta_1, \beta_2)$
end while

Optimize D

Optimize G

目录

Experiments

- **1st part: WGAN-TS is more accurate**
  - metrics
    - ℓ_1 WD
    - ℓ_2 WD
  - models
    - WGAN
    - WGAN-GP (NIPS17)
    - SN-WD (ICLR18)
- **2nd part: WGAN-TS produces better images**
  - dataset
    - MNIST
    - LSUN
    - CIFAR-10
  - Compared models
    - WGAN
    - FisherGAN (NIPS17)
    - WGAN-GP (NIPS17)
    - SN-WD (ICLR18)

1st Part

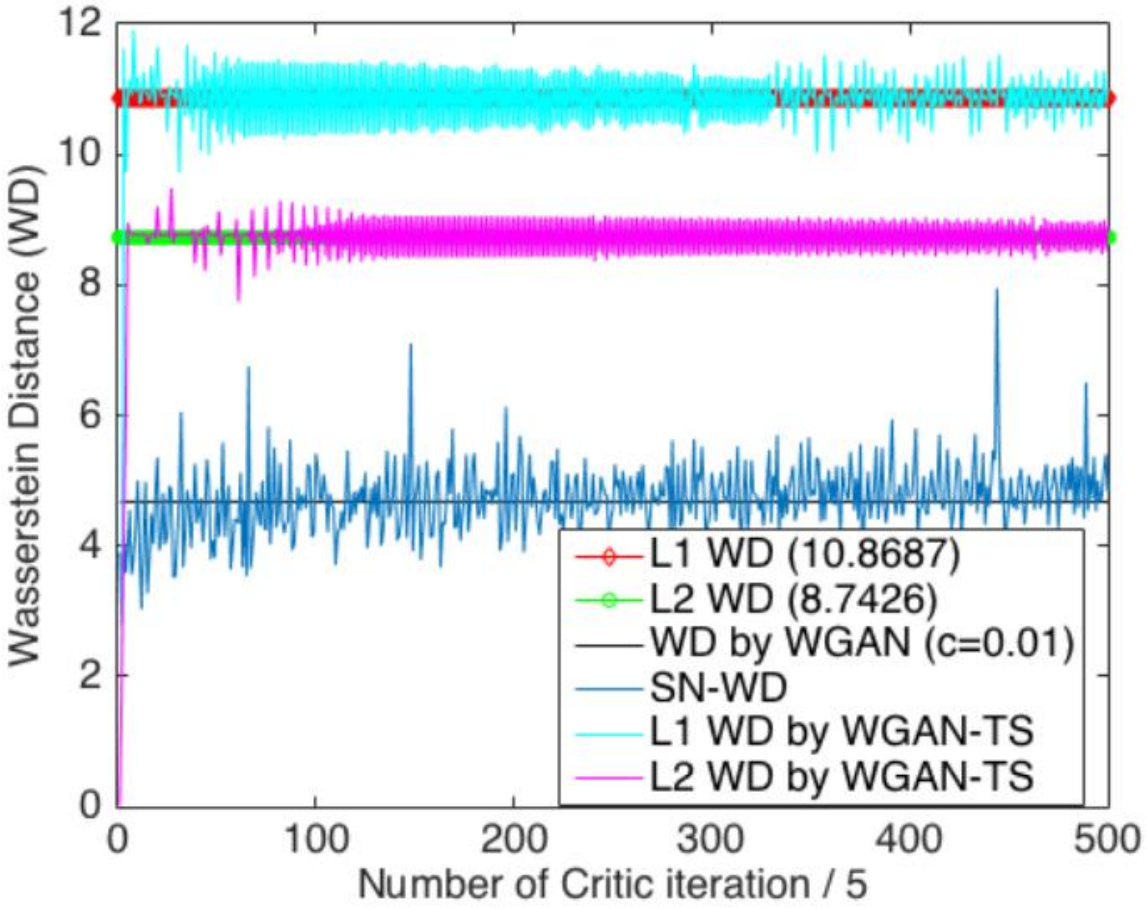
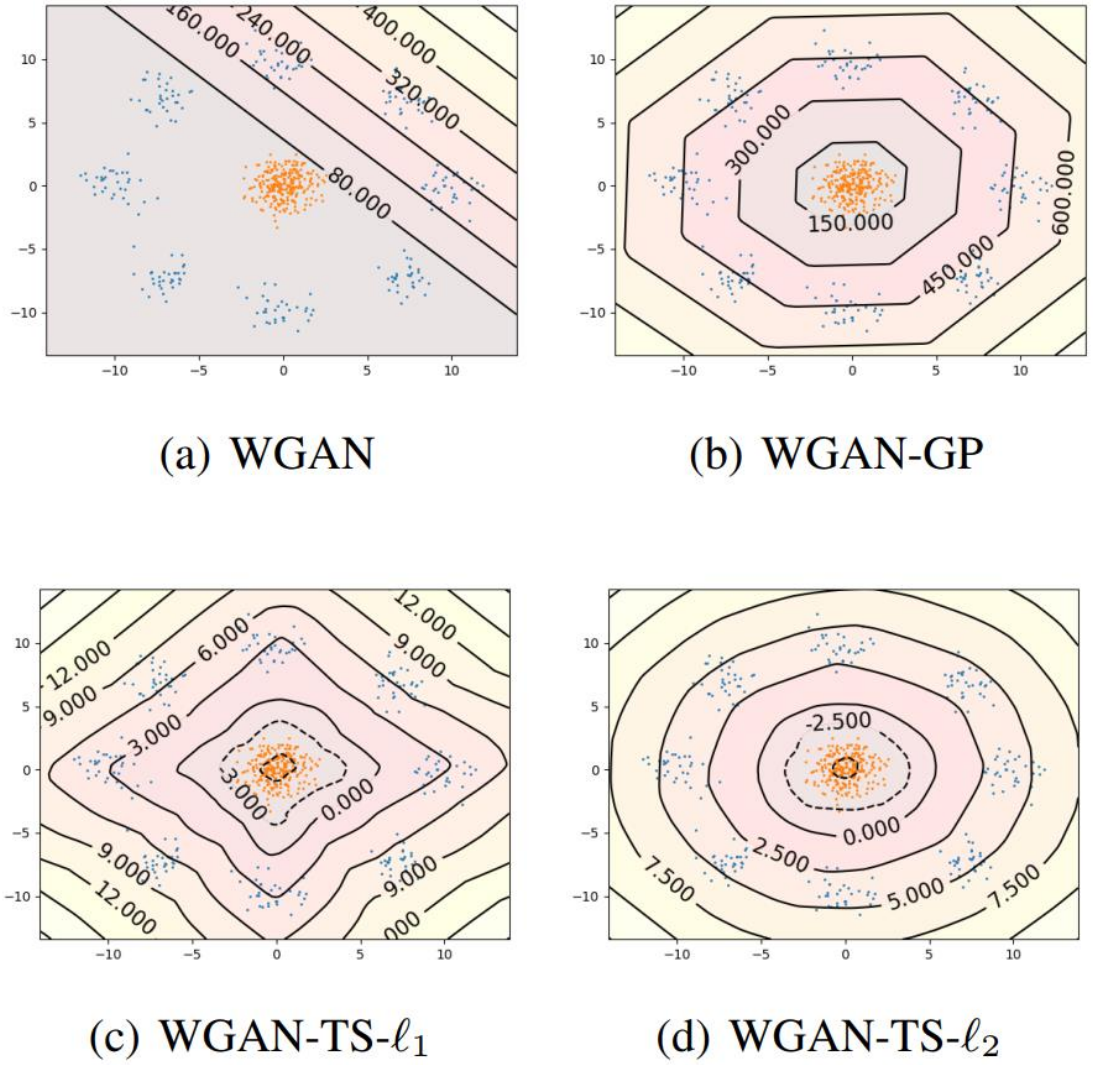


Figure 1. Wasserstein Distance (WD) on the 8 Gaussian toy dataset.

1st Part



(a) WGAN (b) WGAN-GP

(c) WGAN-TS-ℓ_1 (d) WGAN-TS-ℓ_2

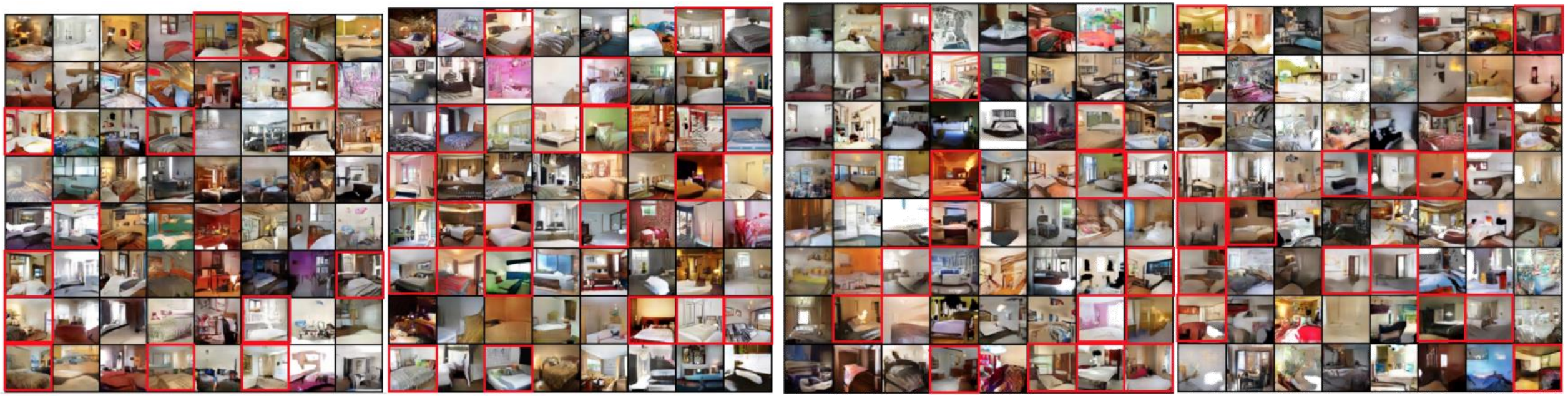
Figure 2. Value surfaces of critics computed by (a) WGAN, (b) WGAN-GP, (c) WGAN-TS-ℓ_1 and (d) WGAN-TS-ℓ_2.

2nd Part – MNIST dataset

(a) WGAN (b) WGAN-GP (c) SN-WD (d) WGAN-TS

WGAN-TS generate more realistic images of digit 8.

2nd Part – LSUN dataset



(a) WGAN (b) WGAN-GP (c) SN-WD (d) WGAN-TS

Figure 4. On the LSUN dataset, images generated by (a) WGAN, (b) WGAN-GP (c) SN-WD and (d) WGAN-TS. We mark the images that we can recognize as bedrooms with red boxes.

2nd Part

(a) WGAN (b) WGAN-GP (c) SN-WD (d) WGAN-TS

Figure 6. On the CIFAR-10 dataset, images generated by (a) WGAN, (b) WGAN-GP, (c) SN-WD and (d) WGAN-TS. Images that are recognizable are boxed in red. (Higher resolution compar-

Quantitative Analysis

the running time of the critic update per generator update.

Table 1. Inception scores on the MNIST and CIFAR-10 datasets.

METHOD	MNIST	CIFAR-10
WGAN	1.64 ± 0.09	2.77 ± 0.18
WGAN-GP	2.34 ± 0.19	2.99 ± 0.22
FISHERGAN	-	1.00 ± 0.00
SN-WD	2.22 ± 0.23	2.96 ± 0.18
WGAN-TS	$\mathbf{2.35 \pm 0.20}$	$\mathbf{3.13 \pm 0.15}$

Table 2. Critic time consumption per generator iteration.

METHOD	CRITIC TIME (IN SECONDS)
WGAN	0.373 ± 0.171
WGAN-GP	0.881 ± 0.277
FISHERGAN	0.499 ± 0.307
SN-WD	0.435 ± 0.272
WGAN-TS	$\mathbf{0.278 \pm 0.129}$

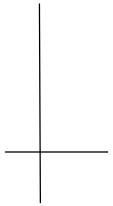
WGAN-TS is more effective and efficient

Conclusion

- A new formulation of the Monge-Kantorovich dual formulation to compute the Wasserstein Distance (WD) is proposed, which can be solved by a combination of linear programming and DNN regression.

- WGAN-TS does not need additional hyper-parameters for weight constraints.

一些想法



AL with Wasserstein Distance

Motivation:

Deep Active Learning: Unified and Principled Method for Query and Training (AISTATS 2020**)**

$$R_{\mathcal{D}}(h) \leq \hat{R}_{\mathcal{Q}}(h) + L(H + \lambda)W_1(\hat{\mathcal{D}}, \hat{\mathcal{Q}}) + L\phi(\lambda) + 2L\mathrm{Rad}_{N_q}(h) + \kappa(\delta, N, N_q),$$

Can we select the representative data by 1-Wasserstein distance?

The Objective Function

$$\max_{T} \quad \frac{1}{m}\sum_{i\in\mathcal{I}} T_i - \frac{1}{n}\sum_{j\in\mathcal{J}} T_j$$

$$\text{s.t.} \quad T_i - T_j \le c_{ij}, \ \forall i \in \mathcal{I}, \forall j \in \mathcal{J}$$

Use a similar trick with MMD

$$\min_{\boldsymbol{w}} \max_{T} \quad \frac{1}{m}\sum_{i\in L} T_i - \frac{1}{b}\sum_{j\in U} w_j T_j$$

$$\text{s.t.} \quad \begin{cases} T_i - T_j \le c_{ij}, & \forall i \in L, \forall j \in U \\ w_j \in \{0,1\}, & \forall j \in U \\ \|w\|_1 = b \end{cases}$$

Optimization

$$\min_{w} \max_{T} \quad \frac{1}{m}\sum_{i \in L} T_i - \frac{1}{b}\sum_{j \in U} w_j T_j$$

$$\text{s.t.} \quad \begin{cases} T_i - T_j \leq c_{ij}, & \forall i \in L, \forall j \in U \\ w_j \in \{0,1\}, & \forall j \in U \\ \|w\|_1 = b \end{cases}$$

A bilinear min-max combinatorial optimization problem*?

$$\min_{x \in X} \max_{y \in Y} y^T C x + Dx + Ey.$$

$$x \in \{0,1\}^n \text{ and } y \in \{0,1\}^m$$

* Pessoa, Artur Alves, et al. "Solving bilevel combinatorial optimization as bilinear min-max optimization via a branch-and-cut algorithm." *Anais do XLV Simpósio Brasileiro de Pesquisa Operacional* (2013).

感谢聆听！