On the Gradient Formula for learning Generative Models with Regularized Optimal Transport Costs

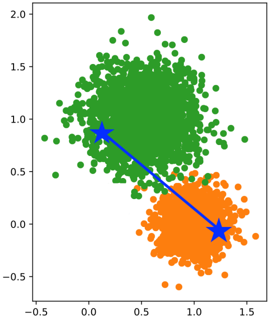
Antoine Houdard, Arthur Leclaire, Nicolas Papadakis, Julien Rabin

Speaker: Nie Yuzhou

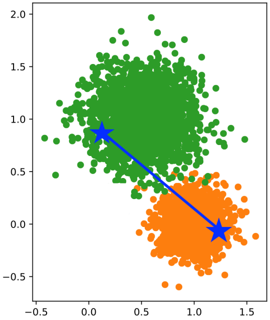
中国人民大學
RENMIN UNIVERSITY OF CHINA

Contents

Optimal Transport Problem



Optimal Transport Problem



Kantorovich-form of optimal transport problem:

$$d_w(\mu,\nu) := \min_{\pi\in\Pi(\mu,\nu)} \mathbb{E}_{x,y\sim\pi}[c(x,y)] = \min_{\pi\in\Pi(\mu_\theta,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)d\pi(x,y), \tag{1}$$

where $\Pi(\mu,\nu) = \left\{\pi > 0 \mid \int_x \pi(x,y)dx = \nu(y), \int_y \pi(x,y)dy = \mu(x)\right\}$, the optimum π^* is called optimal transport (plan).

Regularized Optimal Transport Problem

For $\lambda > 0$, the regularized optimal transport cost is defined by

$$W_\lambda(\mu, \nu) = \min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x, y) d\pi(x, y) + \lambda \mathbb{KL}(\pi \mid \mu \otimes \nu) \tag{2}$$

where $\mathbb{KL}(\pi \mid \mu \otimes \nu) = \int \log\left(\frac{d\pi}{d\mu\otimes\nu}\right) d\pi$ if π admits a density $\frac{d\pi}{d\mu\otimes\nu}$ w.r.t. $\mu \otimes \nu$ and $+\infty$ otherwise.

Regularized Optimal Transport Problem

For $\lambda > 0$, the regularized optimal transport cost is defined by

$$W_\lambda(\mu, \nu) = \min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) d\pi(x, y) + \lambda \mathbb{KL}(\pi \mid \mu \otimes \nu) \tag{2}$$

where $\mathbb{KL}(\pi \mid \mu \otimes \nu) = \int \log\left(\frac{d\pi}{d\mu \otimes \nu}\right) d\pi$ if π admits a density $\frac{d\pi}{d\mu \otimes \nu}$ w.r.t. $\mu \otimes \nu$ and $+\infty$ otherwise.

For discrete setting, regularized optimal transport [Peyre, 2014] (sinkhorn) writes

$$\gamma = \arg\min_{\pi} \quad \langle \pi, \mathbf{C} \rangle_F + \lambda \cdot f(\pi) \tag{3}$$

where $\mathbf{C}$ is cost matrix.

Dual Formulation

OT cost admits a dual formulation [Santambrogio, 2015]

$$\min_{\pi \in \Pi(\mu,\nu)} \mathop{\mathbb{E}}_{(x,y)\sim\pi}[c(x,y)] = \max_{\varphi,\psi}\{\mathop{\mathbb{E}}_{x\sim\mu}[\varphi(x)] + \mathop{\mathbb{E}}_{y\sim\nu}[\psi(y)]\} \tag{4}$$

among all functions $\varphi \in L^1(\mu), \psi \in L^1(\nu)$ such that

$$\varphi(x) + \psi(y) \leq c(x,y), \quad \forall x \in X, y \in Y \tag{5}$$

Dual Formulation

OT cost admits a dual formulation [Santambrogio, 2015]

$$\min_{\pi \in \Pi(\mu,\nu)} \mathop{\mathbb{E}}_{(x,y)\sim\pi}[c(x,y)] = \max_{\varphi,\psi}\{\mathop{\mathbb{E}}_{x\sim\mu}[\varphi(x)] + \mathop{\mathbb{E}}_{y\sim\nu}[\psi(y)]\} \tag{4}$$

among all functions $\varphi \in L^1(\mu), \psi \in L^1(\nu)$ such that

$$\varphi(x) + \psi(y) \leq c(x,y), \quad \forall x \in X, y \in Y \tag{5}$$

when $c(x,y) = \|x-y\|$, we can prove that [Basso, 2015]

$$W(p,q) = \min_{\pi \in \Pi(\mu,\nu)} \mathop{\mathbb{E}}_{(x,y)\sim\pi}[\|x-y\|] = \max_{\|f\|_L \leq 1}\{\mathop{\mathbb{E}}_{x\sim\mu}[f(x)] - \mathop{\mathbb{E}}_{y\sim\nu}[f(y)]\}. \tag{6}$$

$W(p,q)$ is used in Wasserstein GAN

Semi-Dual Formulation

For $\psi \in C(\mathcal{Y})$, let us define the regularized c-transform as in [Feydy, 2019]

$$\psi^{c,\lambda}(x) = \operatorname*{softmin}_{y \in \mathcal{Y}} c(x, y) - \psi(y) \tag{7}$$

where the softmin operation is defined as

$$\operatorname*{softmin}_{y \in \mathcal{Y}} u(y) = \begin{cases} \min_{y \in \mathcal{Y}} u(y) & \text{if } \lambda = 0 \\ -\lambda \log \int e^{-\frac{u(y)}{\lambda}} d\nu(y) & \text{if } \lambda > 0 \end{cases} \tag{8}$$

Semi-Dual Formulation

For $\psi \in C(\mathcal{Y})$, let us define the regularized c-transform as in [Feydy, 2019]

$$\psi^{c,\lambda}(x) = \operatorname*{softmin}_{y \in \mathcal{Y}} c(x,y) - \psi(y) \tag{7}$$

where the softmin operation is defined as

$$\operatorname*{softmin}_{y \in \mathcal{Y}} u(y) = \begin{cases} \min_{y \in \mathcal{Y}} u(y) & \text{if } \lambda = 0 \\ -\lambda \log \int e^{-\frac{u(y)}{\lambda}} d\nu(y) & \text{if } \lambda > 0 \end{cases} \tag{8}$$

We have

$$W_\lambda(\mu,\nu) = \max_{\psi \in C(\mathcal{Y})} \int \psi^{c,\lambda}(x) d\mu(x) + \int \psi(y) d\nu(y) \tag{9}$$

Summary of Dual Formulation

- Dual formulation

$$\max_{\varphi,\psi}\{\mathop{\mathbb{E}}_{x\sim\mu}[\varphi(x)] + \mathop{\mathbb{E}}_{y\sim\nu}[\psi(y)]\}, \quad \text{s.t.} \varphi(x) + \psi(y) \leq c(x, y) \tag{10}$$

- Dual formulation for Wasserstein GAN

$$\max_{\|f\|_L \leq 1}\{\mathop{\mathbb{E}}_{x\sim\mu}[f(x)] - \mathop{\mathbb{E}}_{y\sim\nu}[f(x)]\} \tag{11}$$

- Semi dual formulation

$$\max{}_{\psi\in C(\mathcal{Y})} \mathop{\mathbb{E}}_{x\sim\mu}[\psi^{c,\lambda}(x)] + \mathop{\mathbb{E}}_{y\sim\nu}[\psi(y)], \quad \text{s.t.} \psi^{c,\lambda}(x) = \operatorname*{softmin}_{y\in\mathcal{Y}} c(x, y) - \psi(y) \tag{12}$$

Previous Dual Solver

Previous Dual Solver

- The method of [Arjovsky, 2017], being based on the 1-Wasserstein distance, only requires one dual variable that is constrained to be 1-Lipschitz.

$$\max_{\|f\|_L \leq 1} \left\{ \mathbb{E}_{x \sim \mu} [f(x)] - \mathbb{E}_{y \sim \nu} [f(x)] \right\} \tag{13}$$

  - weight clipping
  - gradient penalty

Previous Dual Solver

- The method of [Arjovsky, 2017], being based on the 1-Wasserstein distance, only requires one dual variable that is constrained to be 1-Lipschitz.

$$\max_{\|f\|_L \leq 1} \{ \mathbb{E}_{x \sim \mu} [f(x)] - \mathbb{E}_{y \sim \nu} [f(x)] \} \tag{13}$$

  - weight clipping
  - gradient penalty
- Article [Seguy, 2018] consider regularized OT costs with a generic cost function. An unconstrained dual problem, but with two dual variables.

Previous Dual Solver

- The method of [Arjovsky, 2017], being based on the 1-Wasserstein distance, only requires one dual variable that is constrained to be 1-Lipschitz.

$$\max_{\|f\|_L \leq 1} \{ \mathbb{E}_{x \sim \mu}[f(x)] - \mathbb{E}_{y \sim \nu}[f(x)] \} \tag{13}$$

  - weight clipping
  - gradient penalty
- Article [Seguy, 2018] consider regularized OT costs with a generic cost function. An unconstrained dual problem, but with two dual variables.
- Article [Chen, 2019] consider the semi-dual formulation of OT, with one dual variable ψ. They do not parameterize the dual variable ψ with a neural network

Previous Dual Solver

- The method of [Arjovsky, 2017], being based on the 1-Wasserstein distance, only requires one dual variable that is constrained to be 1-Lipschitz.

$$\max_{\|f\|_L \leq 1} \{ \underset{x \sim \mu}{\mathbb{E}} [f(x)] - \underset{y \sim \nu}{\mathbb{E}} [f(x)] \} \tag{13}$$

  - weight clipping
  - gradient penalty
- Article [Seguy, 2018] consider regularized OT costs with a generic cost function. An unconstrained dual problem, but with two dual variables.
- Article [Chen, 2019] consider the semi-dual formulation of OT, with one dual variable ψ. They do not parameterize the dual variable ψ with a neural network
  - Article [Mallasto, 2019] propose to parameterize the dual variable ψ by a neural network and to obtain the second one ϕ with an **approximated c-transform** computed on mini-batches.

Differentiability of Dual Formulations

Remind of Wasserstein GAN

$$\begin{aligned} W(\mu_\theta, \nu) &= \max_{\|f\|_L \le 1} \mathbb{E}_{x\sim\nu}[f(x)] - \mathbb{E}_{x\sim\mu_\theta}[f(x)] \\ &= \max_{\|f\|_L \le 1} \mathbb{E}_{x\sim\nu}[f(x)] - \mathbb{E}_{z\sim p(z)}[f(g_\theta(z))] \end{aligned} \tag{14}$$

[Arjovsky, 2017] assumes both sides of the gradient exists,

$$\nabla_\theta(W(\mu_\theta, \nu)) = -\mathbb{E}_{z\sim p(z)}\left[\nabla_\theta f(g_\theta(z))\right] \tag{15}$$

Differentiability of Dual Formulations

Remind of Wasserstein GAN

$$W(\mu_\theta, \nu) = \max_{\|f\|_L \le 1} \mathbb{E}_{x\sim\nu}[f(x)] - \mathbb{E}_{x\sim\mu_\theta}[f(x)] \\ = \max_{\|f\|_L \le 1} \mathbb{E}_{x\sim\nu}[f(x)] - \mathbb{E}_{z\sim p(z)}[f(g_\theta(z))] \tag{14}$$

[Arjovsky, 2017] assumes both sides of the gradient exists,

$$\nabla_\theta(W(\mu_\theta, \nu)) = -\mathbb{E}_{z\sim p(z)}\left[\nabla_\theta f(g_\theta(z))\right] \tag{15}$$

For standard dual formulation,

$$W(\mu_\theta, \nu) = \max_{\varphi, \psi}\{\mathop{\mathbb{E}}_{x\sim\mu_\theta}[\varphi(x)] + \mathop{\mathbb{E}}_{y\sim\nu}[\psi(y)]\} \tag{16}$$

the gradient of θ can also be expressed as

$$\nabla_\theta\left(W(\mu_\theta, \nu)\right) = \nabla_\theta\left(\int \varphi d\mu_\theta\right) \tag{17}$$

Differentiability of Dual Formulations

Remind of Wasserstein GAN

$$W(\mu_\theta, \nu) = \max_{\|f\|_L \le 1} \mathbb{E}_{x\sim\nu}[f(x)] - \mathbb{E}_{x\sim\mu_\theta}[f(x)] \\ = \max_{\|f\|_L \le 1} \mathbb{E}_{x\sim\nu}[f(x)] - \mathbb{E}_{z\sim p(z)}[f(g_\theta(z))] \tag{14}$$

[Arjovsky, 2017] assumes both sides of the gradient exists,

$$\nabla_\theta(W(\mu_\theta, \nu)) = -\mathbb{E}_{z\sim p(z)}\left[\nabla_\theta f(g_\theta(z))\right] \tag{15}$$

For standard dual formulation,

$$W(\mu_\theta, \nu) = \max_{\varphi,\psi}\{\mathop{\mathbb{E}}_{x\sim\mu_\theta}[\varphi(x)] + \mathop{\mathbb{E}}_{y\sim\nu}[\psi(y)]\} \tag{16}$$

the gradient of θ can also be expressed as

$$\nabla_\theta\left(W(\mu_\theta, \nu)\right) = \nabla_\theta\left(\int \varphi d\mu_\theta\right) \tag{17}$$

But would equation* (15) *and* (17) *exists?

Further Analysis of Equation (17)

$$\nabla_\theta \left(W\left(\mu_\theta, \nu\right)\right) = \nabla_\theta \left(\int \varphi d\mu_\theta \right)$$

Further Analysis of Equation (17)

$$\nabla_\theta \left(W \left(\mu_\theta, \nu \right) \right) = \nabla_\theta \left(\int \varphi d\mu_\theta \right)$$

- Such formula was assumed in [Arjovsky, 2017] and [Liu, 2019] for the 1-Wasserstein cost.

$$\nabla_\theta (W(\mu_\theta, \nu)) = -\mathbb{E}_{z \sim p(z)} \left[\nabla_\theta f(g_\theta(z)) \right]$$

Further Analysis of Equation (17)

$$\nabla_\theta \left(W\left(\mu_\theta, \nu\right)\right) = \nabla_\theta \left(\int \varphi d\mu_\theta \right)$$

- Such formula was assumed in [Arjovsky, 2017] and [Liu, 2019] for the 1-Wasserstein cost.

$$\nabla_\theta(W\left(\mu_\theta, \nu\right)) = -\mathbb{E}_{z\sim p(z)}\left[\nabla_\theta f\left(g_\theta(z)\right)\right]$$

- The equation was proved by [Sanjabi, 2018] to the case of regularized Wasserstein costs.

Further Analysis of Equation (17)

$$\nabla_\theta \left(W\left(\mu_\theta, \nu\right)\right) = \nabla_\theta \left(\int \varphi d\mu_\theta \right)$$

- Such formula was assumed in [Arjovsky, 2017] and [Liu, 2019] for the 1-Wasserstein cost.

$$\nabla_\theta(W(\mu_\theta, \nu)) = -\mathbb{E}_{z \sim p(z)}\left[\nabla_\theta f(g_\theta(z))\right]$$

- The equation was proved by [Sanjabi, 2018] to the case of regularized Wasserstein costs.

these proofs are based on the "envelope theorem" (Danskin's theorem), which requires **some regularity assumptions that should be carefully checked**.

Goal

$$\nabla_\theta \left(W\left(\mu_\theta, \nu\right)\right) = \nabla_\theta \left(\int \varphi d\mu_\theta \right)$$

the main goal of this paper is to

- provide a new set of hypotheses that validates (17)
- show how these results apply to generative models parameterized by neural networks.

Learning a Generative Network

Learning a Generative Network

Estimating a WGAN from an empirical data distribution ν consists in minimizing

$$h_\lambda(\theta) = W_\lambda\left(\mu_\theta, \nu\right) \tag{18}$$

Learning a Generative Network

Estimating a WGAN from an empirical data distribution ν consists in minimizing

$$h_\lambda(\theta) = W_\lambda\left(\mu_\theta, \nu\right) \tag{18}$$

where μ_θ is assumed to be the distribution of $g_\theta(Z)$, with $Z \sim N(0, 1)$.

Learning a Generative Network

Estimating a WGAN from an empirical data distribution ν consists in minimizing

$$h_\lambda(\theta) = W_\lambda\left(\mu_\theta, \nu\right) \tag{18}$$

where μ_θ is assumed to be the distribution of $g_\theta(Z)$, with $Z \sim N(0,1)$.

Assume we have $\nabla_\theta\left(W\left(\mu_\theta, \nu\right)\right) = \nabla_\theta\left(\int \varphi d\mu_\theta\right)$

Learning a Generative Network

Estimating a WGAN from an empirical data distribution ν consists in minimizing

$$h_\lambda(\theta) = W_\lambda\left(\mu_\theta, \nu\right) \tag{18}$$

where μ_θ is assumed to be the distribution of $g_\theta(Z)$, with $Z \sim N(0,1)$.

Assume we have $\nabla_\theta\left(W\left(\mu_\theta, \nu\right)\right) = \nabla_\theta\left(\int \varphi d\mu_\theta\right)$

Then we define

$$I(\varphi, \theta) = \int_{\mathcal{X}} \varphi d\mu_\theta, \quad \left(\varphi \in C(\mathcal{X}), \theta \in \Theta\right) \tag{19}$$

Learning a Generative Network

With

$$I(\varphi, \theta) = \int_{\mathcal{X}} \varphi d\mu_\theta, \quad (\varphi \in C(\mathcal{X}), \theta \in \Theta)$$

And remind the semi-dual formulation

$$\max_{\psi \in C(\mathcal{Y})} \mathop{\mathbb{E}}_{x \sim \mu} [\psi^{c,\lambda}(x)] + \mathop{\mathbb{E}}_{y \sim \nu} [\psi(y)], \quad \text{s.t.} \psi^{c,\lambda}(x) = \operatorname*{softmin}_{y \in \mathcal{Y}} c(x, y) - \psi(y)$$

the semi-dual expression of optimal transport gives

$$h_\lambda(\theta) = W_\lambda (\mu_\theta, \nu) = \max_{\psi \in C(\mathcal{Y})} I\left(\psi^{c,\lambda}, \theta\right) + \mathop{\mathbb{E}}_{y \sim \nu} \psi(y) \tag{20}$$

define $F_\lambda : C(\mathcal{Y}) \times \Theta \to \mathbb{R}$ with

$$\forall \psi \in C(\mathcal{Y}), \forall \theta \in \Theta, \quad F_\lambda(\psi, \theta) = I\left(\psi^{c,\lambda}, \theta\right) = \int_{\mathcal{X}} \psi^{c,\lambda} d\mu_\theta = \mathbb{E}\left[\psi^{c,\lambda}\left(g_\theta(Z)\right)\right] \tag{21}$$

Learning a Generative Network

For summary, the problem writes

$$W_\lambda\left(\mu_\theta, \nu\right) = h_\lambda(\theta) = \max_{\psi \in C(\mathcal{Y})} H_\lambda(\psi, \theta) \tag{22}$$

with

$$H_\lambda(\psi, \theta) = F_\lambda(\psi, \theta) + \int_{\mathcal{Y}} \psi d\nu \tag{23}$$

Learning a Generative Network

For summary, the problem writes

$$W_\lambda\left(\mu_\theta, \nu\right) = h_\lambda(\theta) = \max_{\psi \in C(\mathcal{Y})} H_\lambda(\psi, \theta) \tag{22}$$

with

$$H_\lambda(\psi, \theta) = F_\lambda(\psi, \theta) + \int_{\mathcal{Y}} \psi d\nu \tag{23}$$

Then equation (17) that needs proof writes

$$\nabla h_\lambda\left(\theta\right) = \nabla_\theta F_\lambda\left(\psi_0, \theta\right) \tag{24}$$

Learning a Generative Network

For summary, the problem writes

$$W_\lambda\left(\mu_\theta,\nu\right)=h_\lambda(\theta)=\max_{\psi\in C(\mathcal{Y})}H_\lambda(\psi,\theta) \tag{22}$$

with

$$H_\lambda(\psi,\theta)=F_\lambda(\psi,\theta)+\int_{\mathcal{Y}}\psi d\nu \tag{23}$$

Then equation (17) that needs proof writes

$$\nabla h_\lambda\left(\theta\right)=\nabla_\theta F_\lambda\left(\psi_0,\theta\right) \tag{24}$$

- **the differentiability of all terms**

Learning a Generative Network

For summary, the problem writes

$$W_\lambda\left(\mu_\theta, \nu\right) = h_\lambda(\theta) = \max_{\psi \in C(\mathcal{Y})} H_\lambda(\psi, \theta) \tag{22}$$

with

$$H_\lambda(\psi, \theta) = F_\lambda(\psi, \theta) + \int_{\mathcal{Y}} \psi d\nu \tag{23}$$

Then equation (17) that needs proof writes

$$\nabla h_\lambda\left(\theta\right) = \nabla_\theta F_\lambda\left(\psi_0, \theta\right) \tag{24}$$

- **the differentiability of all terms**
- the validity of the equation

Previous Results on the Differentiability of OT Costs

Previous Results on the Differentiability of OT Costs

- Article [Arjovsky, 2017] and [Liu, 2019] assumed that the differentiability of all terms exists in WGAN's scenario.

Previous Results on the Differentiability of OT Costs

- Article [Arjovsky, 2017] and [Liu, 2019] assumed that the differentiability of all terms exists in WGAN's scenario.
- Article [Sanjabi et al., 2018] proved the existence limited to discrete regularized OT.

A counter example

Let $\mu_\theta = \delta_\theta$ and $\nu = \frac{1}{2}\delta_{y_1} + \frac{1}{2}\delta_{y_2}$. For $p \geq 1$, consider the cost $c(x, y) = \|x - y\|^p$ where $\|\cdot\|$ is the Euclidean norm on $\mathbb{R}^d$. Then

- $h(\theta) = W(\mu_\theta, \nu)$ is differentiable at any $\theta \notin \{y_1, y_2\}$ for $p = 1$, and at any θ for $p > 1$,
- for any $\psi_{*0} \in \arg\max_\psi H(\psi, \theta)$, the function $\theta \mapsto F(\psi_{*0}, \theta)$ is not differentiable at θ.

It can be proved that relation $\nabla h_\lambda(\theta) = \nabla_\theta F_\lambda(\psi_0, \theta)$ never stands.

Frame Title

- unregularized semi-discrete setting
-
- regularized setting

Alternate Algorithm

The objective funtion writes

$$\begin{aligned}\min_{\theta\in\Theta} W_\lambda\left(\mu_\theta,\nu\right) &= \min h_\lambda(\theta) \\ &= \min_{\theta\in\Theta} \max_{\psi\in C(\mathcal{Y})} H_\lambda(\psi,\theta) \\ \text{where } H_\lambda(\psi,\theta) &= \int_{\mathcal{X}} \psi^{c,\lambda} d\mu_\theta + \int_{\mathcal{Y}} \psi d\nu = F_\lambda(\psi,\theta) + \sum_{y\in\mathcal{Y}} \psi(\boldsymbol{y})\nu(\{\boldsymbol{y}\}).\end{aligned} \tag{25}$$

Alternate Algorithm

The objective funtion writes

$$\begin{aligned}\min_{\theta\in\Theta} W_\lambda\left(\mu_\theta,\nu\right) &= \min h_\lambda(\theta)\\ &= \min_{\theta\in\Theta}\max_{\psi\in C(\mathcal{Y})} H_\lambda(\psi,\theta)\\ \text{where } H_\lambda(\psi,\theta) &= \int_{\mathcal{X}}\psi^{c,\lambda}d\mu_\theta+\int_{\mathcal{Y}}\psi d\nu = F_\lambda(\psi,\theta)+\sum_{y\in\mathcal{Y}}\psi(y)\nu(\{y\}).\end{aligned} \tag{25}$$

We write

$$F_\lambda(\psi,\theta)=\mathbb{E}\left[f_\lambda(\psi,\theta,Z)\right] \tag{26}$$

where $f_\lambda(\psi,\theta,z)=\psi^{c,\lambda}\left(g_\theta(z)\right)$

Alternate Algorithm

It has been shown in [Genevay, 2016;Houdard, 2022] that $H_\lambda(\cdot, \theta)$ is a concave function whose supergradient $\mathcal{D}(\psi, \theta) = \partial_\psi H_\lambda(\psi, \theta)$ can be written as

$$\mathcal{D}(\psi, \theta) = \mathbb{E}[\mathrm{D}(\psi, \theta, Z)] \tag{27}$$

where $\quad \mathrm{D}(\psi, \theta, z) = \partial_\psi \left(f_\lambda(\psi, \theta, z) - \int \psi d\nu\right)$

$\mathrm{D}(\psi, \theta, z) \in \mathbb{R}^J$ can be computed with an explicit formula given in [Genevay, 2016;Houdard, 2022].

In practice it is implemented by automatic differentiation

Updating ψ and θ

So for a current θ, optimize ψ with

$$\begin{cases} \widetilde{\psi}_k = \widetilde{\psi}_{k-1} + \frac{\gamma}{k^\alpha}\left(\frac{1}{|B_k|}\sum_{z\in B_k} \mathrm{D}\left(\widetilde{\psi}_{k-1},\theta,z\right)\right) \\ \psi_k = \frac{1}{k}\left(\widetilde{\psi}_1 + \cdots + \widetilde{\psi}_k\right) \end{cases} \tag{28}$$

where $\gamma > 0$ is the learning rate, $\alpha \in (0,1)$ a parameter, B_k is a batch containing b independent samples of the distribution of Z.

Updating ψ and θ

So for a current θ, optimize ψ with

$$\begin{cases} \widetilde{\psi}_k = \widetilde{\psi}_{k-1} + \frac{\gamma}{k^\alpha}\left(\frac{1}{|B_k|}\sum_{z\in B_k} \mathrm{D}\left(\widetilde{\psi}_{k-1},\theta,z\right)\right) \\ \psi_k = \frac{1}{k}\left(\widetilde{\psi}_1 + \cdots + \widetilde{\psi}_k\right) \end{cases} \tag{28}$$

where $\gamma > 0$ is the learning rate, $\alpha \in (0,1)$ a parameter, B_k is a batch containing b independent samples of the distribution of Z.

After k inner loop, fix $\underline{\psi}$ and update θ

$$\nabla_\theta h_\lambda(\theta) \approx \nabla_\theta H_\lambda(\underline{\psi},\theta) = \nabla_\theta F_\lambda(\underline{\psi},\theta) = \mathbb{E}\left[D(g\left(\theta,Z\right)^T,\theta)\nabla\underline{\psi}^{c,\lambda}\left(g\left(\theta,Z\right)\right)\right] \tag{29}$$

the expectation cannot be computed in closed form so that an approximation is computed by another batch B' on z

$$\nabla_\theta H_\lambda(\underline{\psi},\theta) \approx \frac{1}{|B'|}\sum_{z\in B'} D_\theta g\left(\theta,z\right)^T \nabla\underline{\psi}^{c,\lambda}\left(g\left(\theta,z\right)\right) \tag{30}$$

Algorithm Illustration

Algorithm 1

Initialization: $\psi_0 = 0$, random initialization of θ

$n = 1$ **to** N

- Approximate $\psi_n \approx \arg\max H_\lambda(\cdot, \theta)$: inner loop with K iterations of ASGD (102) starting from ψ_{n-1}, using batches $B_{n,1}, \ldots, B_{n,K}$ of size b on z
- Update θ with one step of ADAM algorithm on $H_\lambda(\psi_n, \cdot)$ using gradient (104) computed on a batch B'_n of size b on z

end for

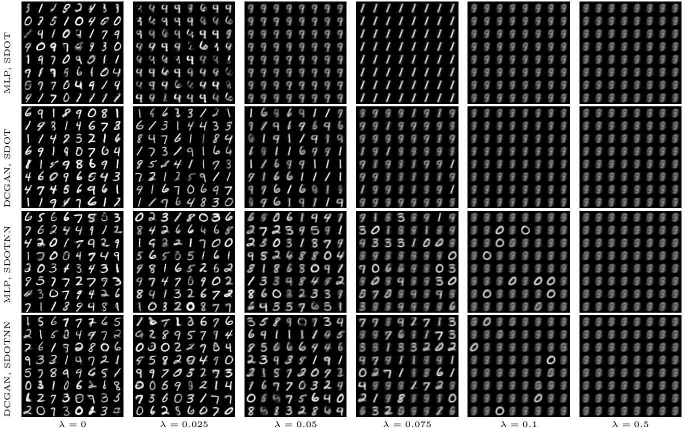
Output: estimated generative model parameter θ

- for $\alpha = 0.5$, ASGD algorithm has a convergence guarantee in $\mathcal{O}\left(\frac{\log k}{\sqrt{k}}\right)$
- The exact computation of $\psi^{c,\lambda}$ requires to visit all the dataset $\mathcal{Y}$, which is prohibitive for a very large database.

Implementation Details

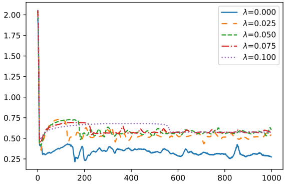
- The cost $c(x, y)$ is the quadratic cost $\alpha^{-1}\|x - y\|^2$ normalized by $\alpha = \frac{1}{J}\sum_{y\in\mathcal{Y}} \|y\|^2$
- For the generator g_θ, two different architectures are considered:
  - a multilayer perceptron (MLP) with four fully-connected layers; the number of channels for the successive hidden layers is $256, 512, 1024$.
  - a Deep Convolutional Adversarial Network (DCGAN) [Radford, 2015] adapted for the dimension 28×28 of MNIST images, with four deconvolution layers; the number of channels for the successive hidden layers is $256, 128, 64$.
- The dual variable ψ also have two alternative architectures:
  - directly modeled by a vector $\psi \in \mathbb{R}^J$: SDOT (for semi-dual OT)
  - parameterized by a multilayer perceptron with four fully-connected layers: SDOTNN (for semi-dual OT with neural network)

Sampled Digits

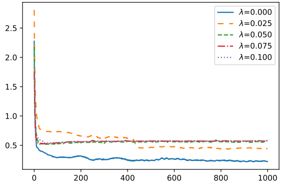


- The generators learned with unregularized OT ($\lambda = 0$) produce mostly convincing samples.
- DCGAN provides cleaner samples that better cover the whole database.

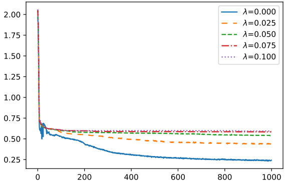
Impact of the Regularization Parameter



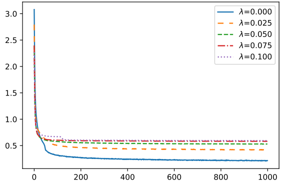
MLP, SDOT



DCGAN, SDOT



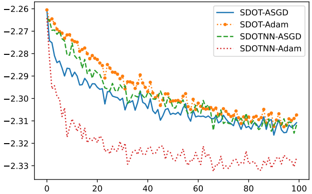
MLP, SDOTNN



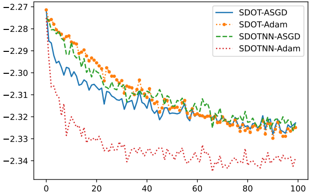
DCGAN, SDOTNN

- the convergence speed does not improve drastically when using a larger regularization parameter λ

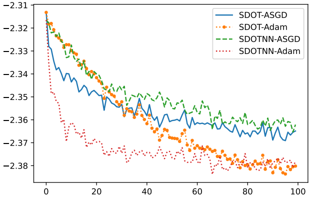
Impact of the Regularization Parameter For ASGD



$\lambda = 0$



$\lambda = 0.001$



$\lambda = 0.005$

- using the ADAM algorithm with the SDOTNN parameterization seems beneficial for all tested regularization parameters, and the convergence is much faster.

Thanks!