Optimal Transport for Machine Learning

Aude Genevay

CEREMADE (Université Paris-Dauphine)
DMA (Ecole Normale Supérieure)
MOKAPLAN Team (INRIA Paris)

Imaging in Paris - February 2018

Outline

1. Entropy Regularized OT
2. Applications in Imaging
3. Large Scale "OT" for Machine Learning

Shortcomings of OT

Two main issues when using OT in practice :

- Poor sample complexity : need a lot of samples from μ and ν to get a good approximation of $W(\mu, \nu)$
- Heavy computational cost : solving discrete OT requires solving an LP $\rightarrow$ network simplex solver $O(n^3 log(n))$ [Pele and Werman '09]

Entropy!

- Basically : Adding an entropic regularization smoothes the constraint
- Makes the problem easier :
  - yields an unconstrained dual problem
  - discrete case can be solved efficiently with iterative algorithm (more on that later)
- For ML applications, regularized Wasserstein is better than standard one
- In high dimension, helps avoiding overfitting

Entropic Relaxation of OT [Cuturi '13]

Add entropic penalty to Kantorovitch formulation of OT

$$\min_{\gamma \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)\mathrm{d}\gamma(x,y) + \varepsilon\, \mathsf{KL}(\gamma|\mu\otimes\nu)$$

where

$$\mathsf{KL}(\gamma|\mu\otimes\nu) \overset{\text{def.}}{=} \int_{\mathcal{X}\times\mathcal{Y}} \left(\log\left(\frac{\mathrm{d}\gamma}{\mathrm{d}\mu\mathrm{d}\nu}(x,y)\right) - 1\right)\mathrm{d}\gamma(x,y)$$

Dual Formulation

$$\max_{u \in C(\mathcal{X}) v \in C(\mathcal{Y})} \int_{\mathcal{X}} u(x) d\mu(x) + \int_{\mathcal{Y}} v(y) d\nu(y) - \varepsilon \int_{\mathcal{X} \times \mathcal{Y}} e^{\frac{u(x)+v(y)-c(x,y)}{\varepsilon}} d\mu(x) d\nu(y)$$

Constraint in standard OT $u(x) + v(y) < c(x, y)$ replaced by a smooth penalty term.

Dual Formulation

Dual problem concave in u and v, first order condition for each variable yield :

$$\nabla_u = 0 \Leftrightarrow u(x) = -\varepsilon \log\left(\int_{\mathcal{Y}} e^{\frac{v(y)-c(x,y)}{\varepsilon}} d\nu(y)\right)$$

$$\nabla_v = 0 \Leftrightarrow v(y) = -\varepsilon \log\left(\int_{\mathcal{X}} e^{\frac{u(x)-c(x,y)}{\varepsilon}} d\mu(x)\right)$$

The Discrete Case

Dual problem :

$$\max_{u\in\mathbb{R}^m v\in\mathbb{R}^n} \quad \sum_{i=1}^{n} u_i\mu_i + \sum_{j=1}^{m} v_j\nu_j - \varepsilon \sum_{i,j=1}^{n,m} e^{\frac{u_i+v_j-c(x_i,y_j)}{\varepsilon}}\mu_i\nu_j$$

First order conditions for each variable:

$$\nabla_u = 0 \Leftrightarrow u_i = -\varepsilon \log(\sum_{j=1}^{m} e^{\frac{v_j-c(x_i,y_j)}{\varepsilon}}\nu_j)$$

$$\nabla_v = 0 \Leftrightarrow v_j = -\varepsilon \log(\sum_{i=1}^{n} e^{\frac{u_i-c(x_i,y_j)}{\varepsilon}}\mu_i)$$

$\Rightarrow$ Do alternate maximizations!

Sinkhorn's Algorithm

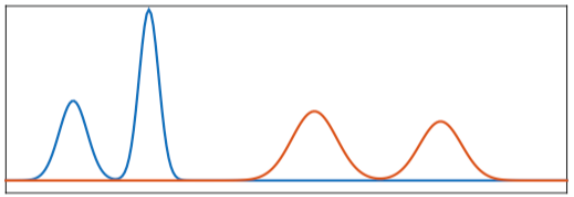
- Iterates $(a, b) := (e^{\frac{u}{\varepsilon}}, e^{\frac{v}{\varepsilon}})$

Sinkhorn algorithm [Cuturi 13]

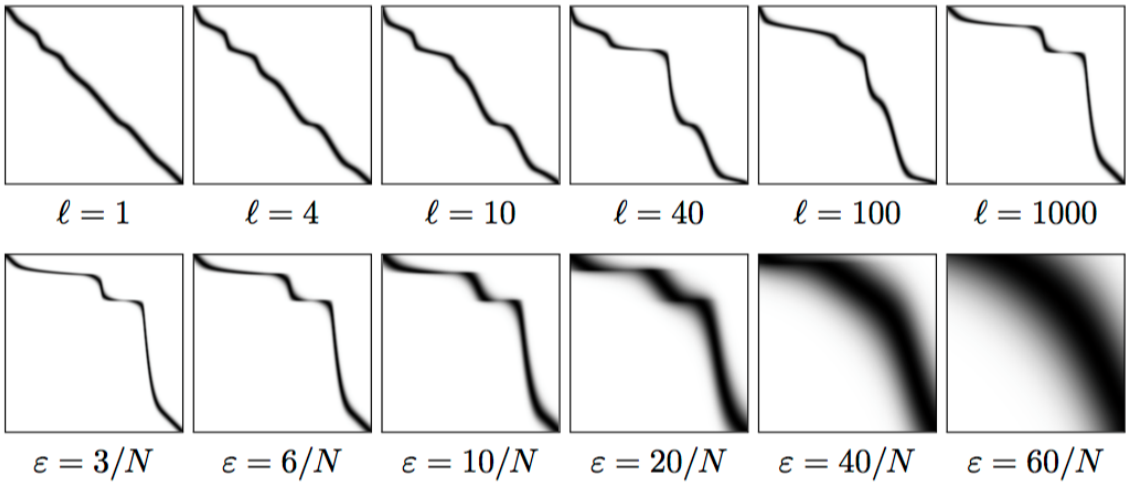
$$\begin{aligned}
&\textbf{initialize} \quad b \leftarrow \mathbb{1}_m \quad K \leftarrow (e^{-c_{ij}/\varepsilon} m_{ij})_{ij} \\
&\textbf{repeat} \\
&\qquad a \leftarrow \mu \oslash Kb \\
&\qquad b \leftarrow \nu \oslash K^T a \\
&\textbf{return} \quad \gamma = \mathrm{diag}(a) K \mathrm{diag}(b)
\end{aligned}$$

- each iteration $O(nm)$ complexity (matrix vector multiplication)
- can be improved to $O(n \log n)$ on gridded space with convolutions [Solomon et al. '15]

Sinkhorn - Toy Example



Marginals μ and ν



top : evolution of γ with number of iterations l

bottom : evolution of γ with regularization parameter ε

Sinkhorn - Convergence

Definition (Hilbert metric)

Projective metric defined for $x, y \in \mathbb{R}^d_{++}$ by

$$d_H(x, y) := \log \frac{\max_i(x_i/y_i)}{\min_i(x_i/y_i)}$$

Theorem

The iterates $(a^{(l)}, b^{(l)})$ converge linearly for the Hilbert metric.

Remark : the contraction coefficient deteriorates quickly when $\varepsilon \to 0$ (exponentially in worst-case)

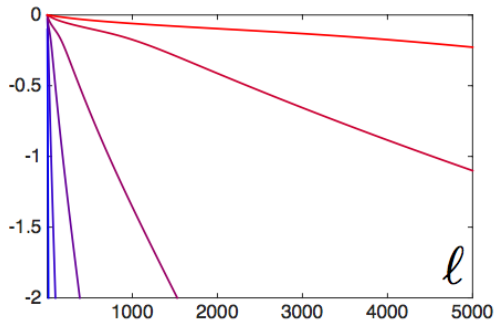
Sinkhorn - Convergence

Constraint violation

We have the following bound on the iterates:

$$d_H(a^{(l)}, a^\star) \leq \kappa d_H(\gamma \mathbb{1}_m, \mu)$$

So monitoring the violation of the marginal constraints is a good way to monitor convergence of Sinkhorn's algorithm



$\|\gamma \mathbb{1}_m - \mu\|$ for various regularizations

Color Transfer

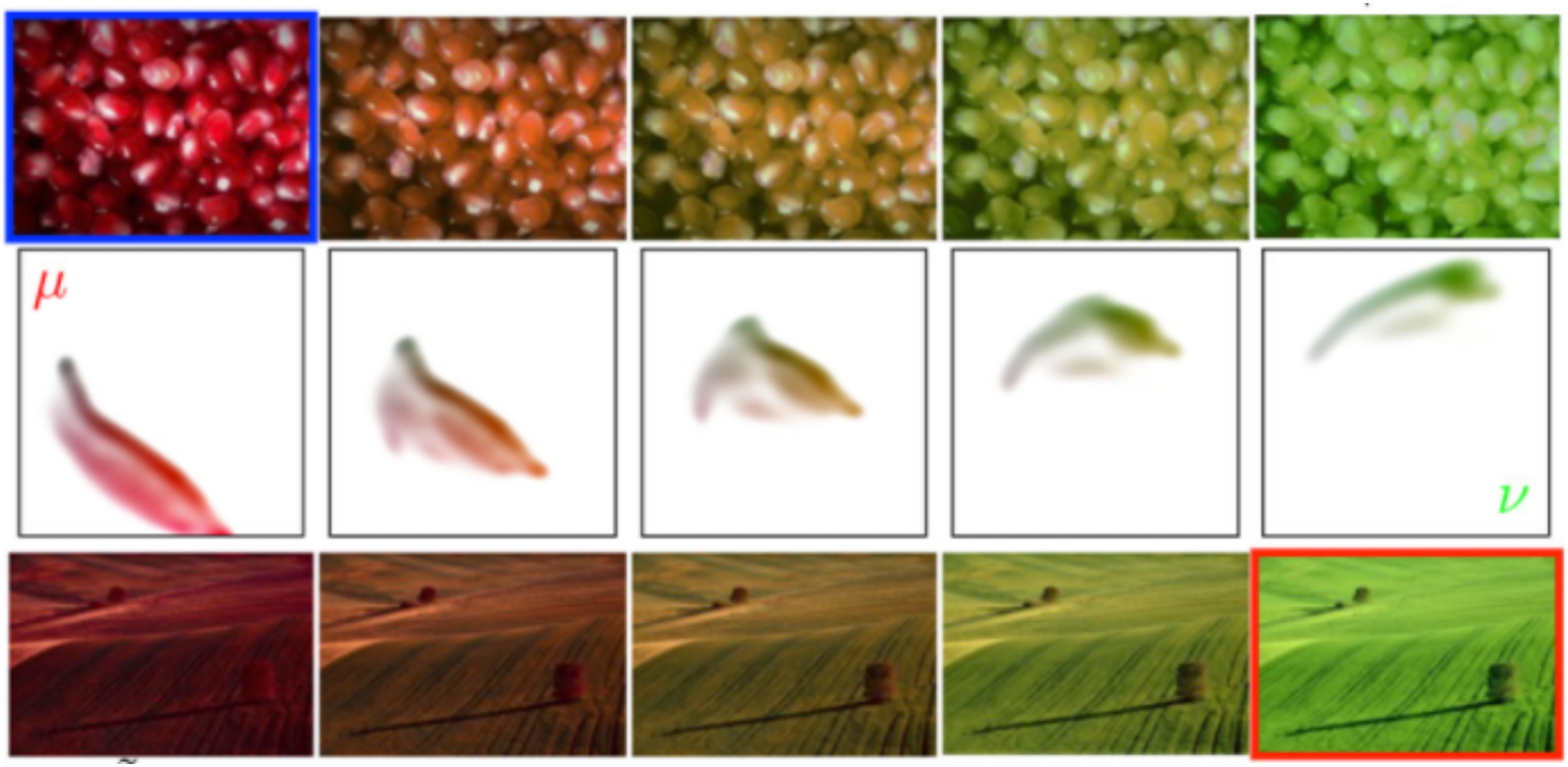
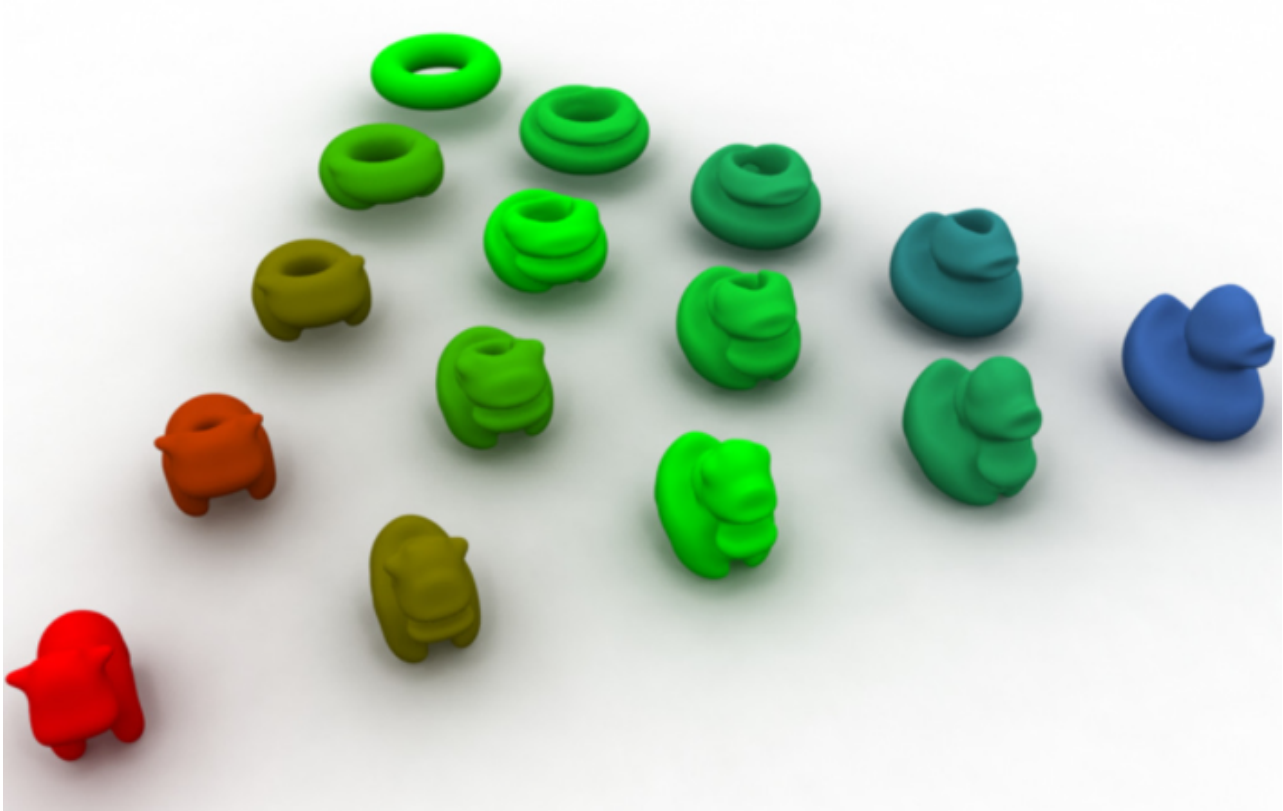


Image courtesy of G. Peyré

Barycenters [Solomon et al.]

Wasserstein Barycenters

$$\bar{\mu} = \arg\min_{\mu} W(\mu_k, \mu)$$



Sinkhorn loss

Consider entropy-regularized OT

$$\min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)\mathrm{d}\pi(x,y) + \varepsilon\,\mathrm{KL}(\pi|\mu\otimes\nu)$$

Regularized loss :

$$W_{c,\varepsilon}(\mu,\nu) \overset{\text{def.}}{=} \int_{\mathcal{X}\mathcal{Y}} c(x,y)d\pi_\varepsilon(x,y)$$

where π_ε solution of (15)

Sinkhorn Divergences : interpolation between OT and MMD

Theorem

The Sinkhorn loss between two measures μ, ν is defined as:

$$\bar{W}_{c,\varepsilon}(\mu,\nu) = 2W_{c,\varepsilon}(\mu,\nu) - W_{c,\varepsilon}(\mu,\mu) - W_{c,\varepsilon}(\nu,\nu)$$

with the following limiting behavior in ε:

1. *as* $\varepsilon \to 0, \quad \bar{W}_{c,\varepsilon}(\mu,\nu) \to 2W_c(\mu,\nu)$
2. *as* $\varepsilon \to +\infty, \quad \bar{W}_{c,\varepsilon}(\mu,\nu) \to \|\mu - \nu\|_{-c}$

where $\|\cdot\|_{-c}$ is the MMD distance whose kernel is minus the cost from OT.

Remark : Some conditions are required on c to get MMD distance when $\varepsilon \to \infty$. In particular, $c = \|\cdot\|_p^p, 0 < p < 2$ is valid.

Sample Complexity

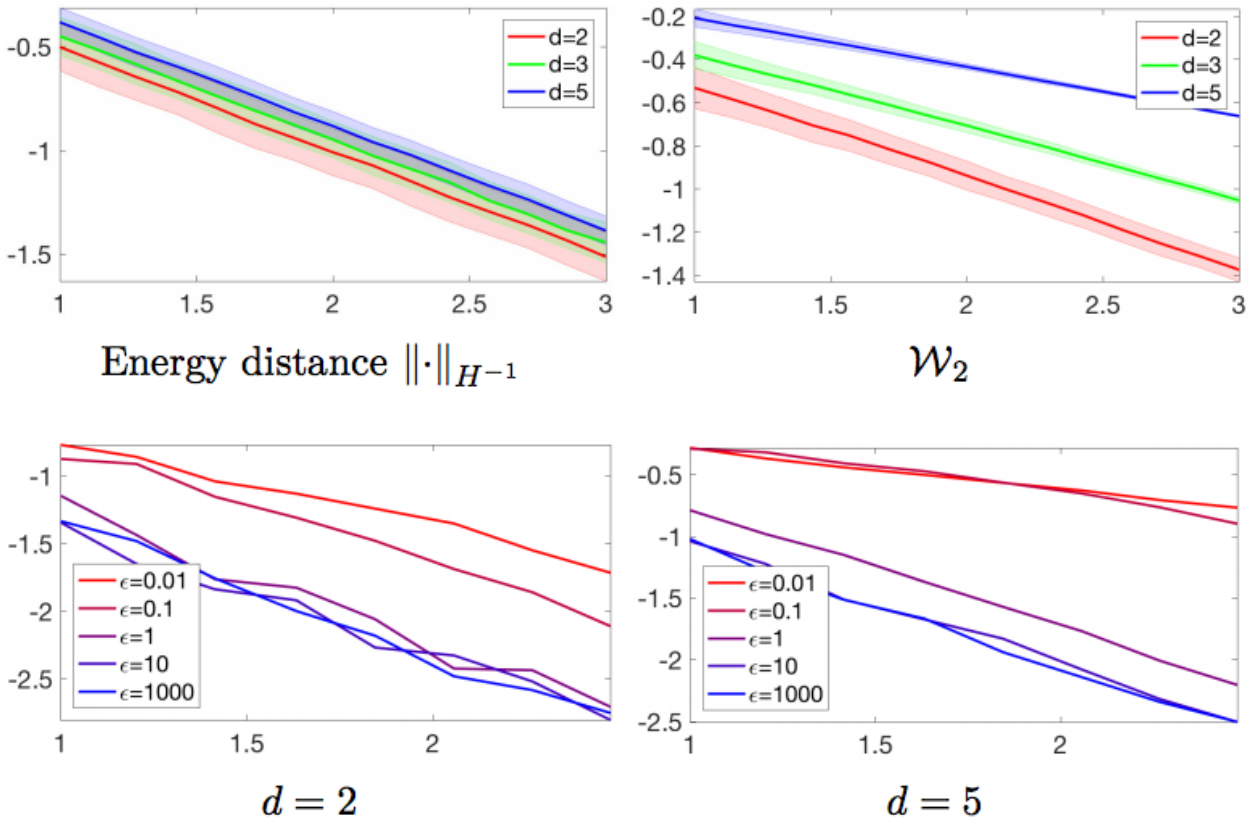
Sample Complexity of OT and MMD

Let μ a probability distribution on $\mathbb{R}^d$, and $\hat{\mu}_n$ an empirical measure from μ

$$\begin{aligned} W(\mu, \hat{\mu}_n) &= O(n^{-1/d}) \\ MMD(\mu, \hat{\mu}_n) &= O(n^{-1/2}) \end{aligned}$$

$\Rightarrow$ the number n of samples you need to get a precision η on the Wassertein distance grows exponentially with the dimension d of the space!

Sample Complexity - Sinkhorn loss



Sample Complexity of Sinkhorn loss seems to improve as ε grows.

Plots courtesy of G. Peyré and M. Cuturi

Generative Models

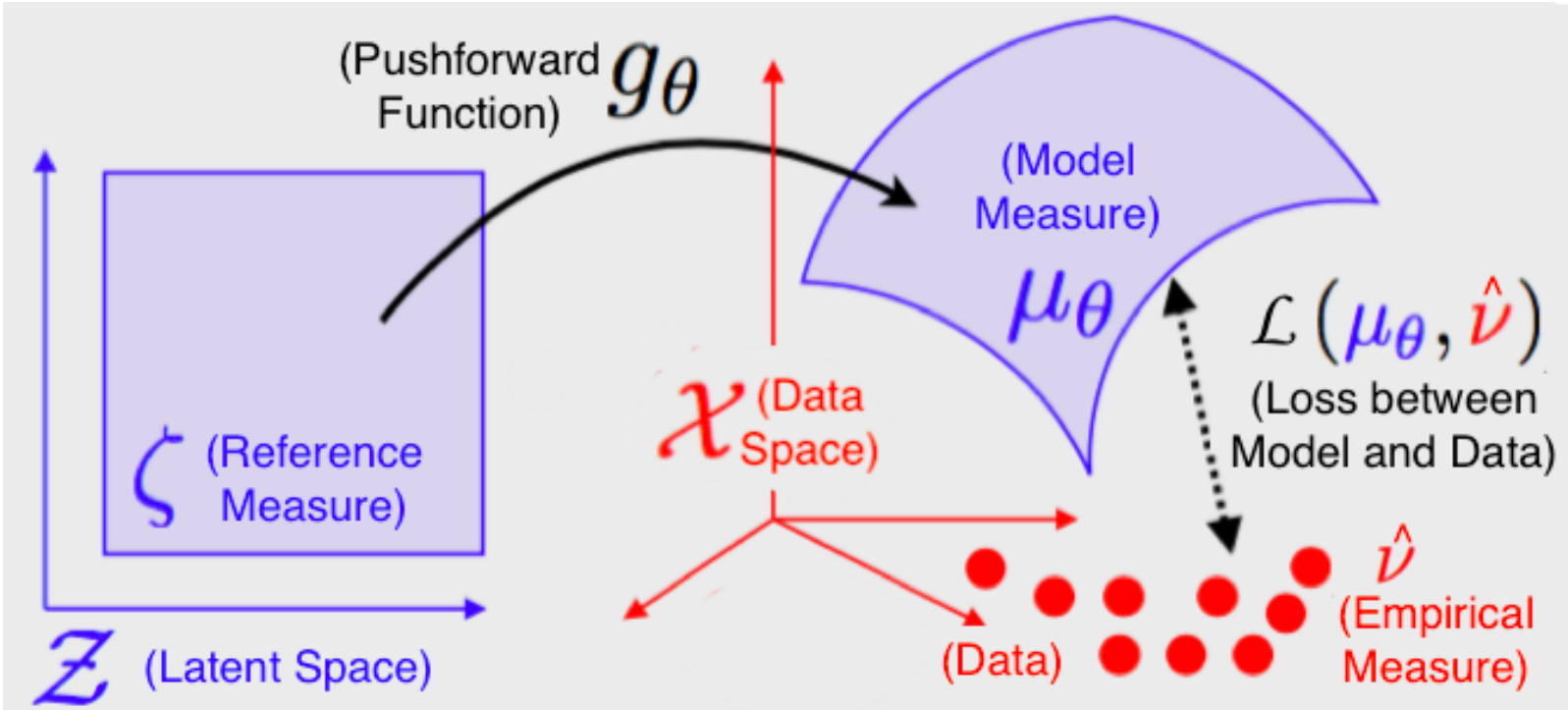


Figure: Illustration of Density Fitting on a Generative Model

Density Fitting with Sinkhorn loss "Formally"

Solve $\min_\theta E(\theta)$

where $E(\theta) \overset{\text{def.}}{=} \bar{W}_{c,\varepsilon}(\mu_\theta, \nu)$

$\Rightarrow$ Issue : untractable gradient

Approximating Sinkhorn loss

- Rather than approximating the gradient approximate the loss itself
- Minibatches : $\hat{E}(\theta)$
  - sample $x_1, \dots, x_m$ from μ_θ
  - use empirical Wasserstein distance $W_{c,\varepsilon}(\hat{\mu}_\theta, \hat{\nu})$ where $\hat{\mu}_\theta = \frac{1}{N}\sum_{i=1}^{m} \delta_{x_i}$
- Use L iterations of Sinkhorn's algorithm : $\hat{E}^{(L)}(\theta)$
  - compute L steps of the algorithm
  - use this as a proxy for $W(\hat{\mu}_\theta, \nu)$

Computing the Gradient in Practice

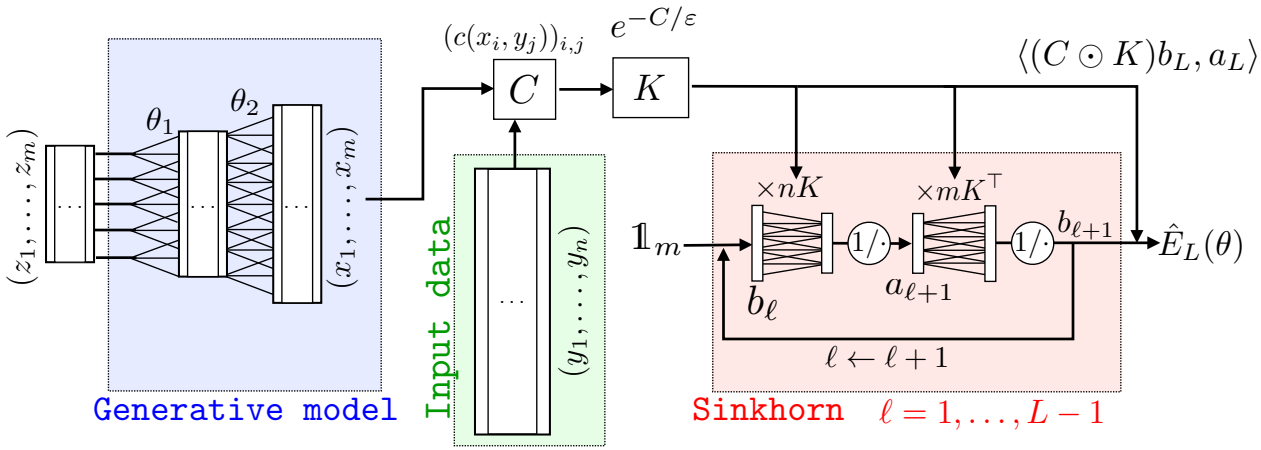


Figure: Scheme of the loss approximation

- Compute *exact* gradient of $\hat{E}^{(L)}(\theta)$ with autodiff
- Backpropagation through above graph
- Same computational cost as evaluation of $\hat{E}^{(L)}(\theta)$

Numerical Results on MNIST (L2 cost)

Figure: Samples from MNIST dataset

Numerical Results on MNIST (L2 cost)

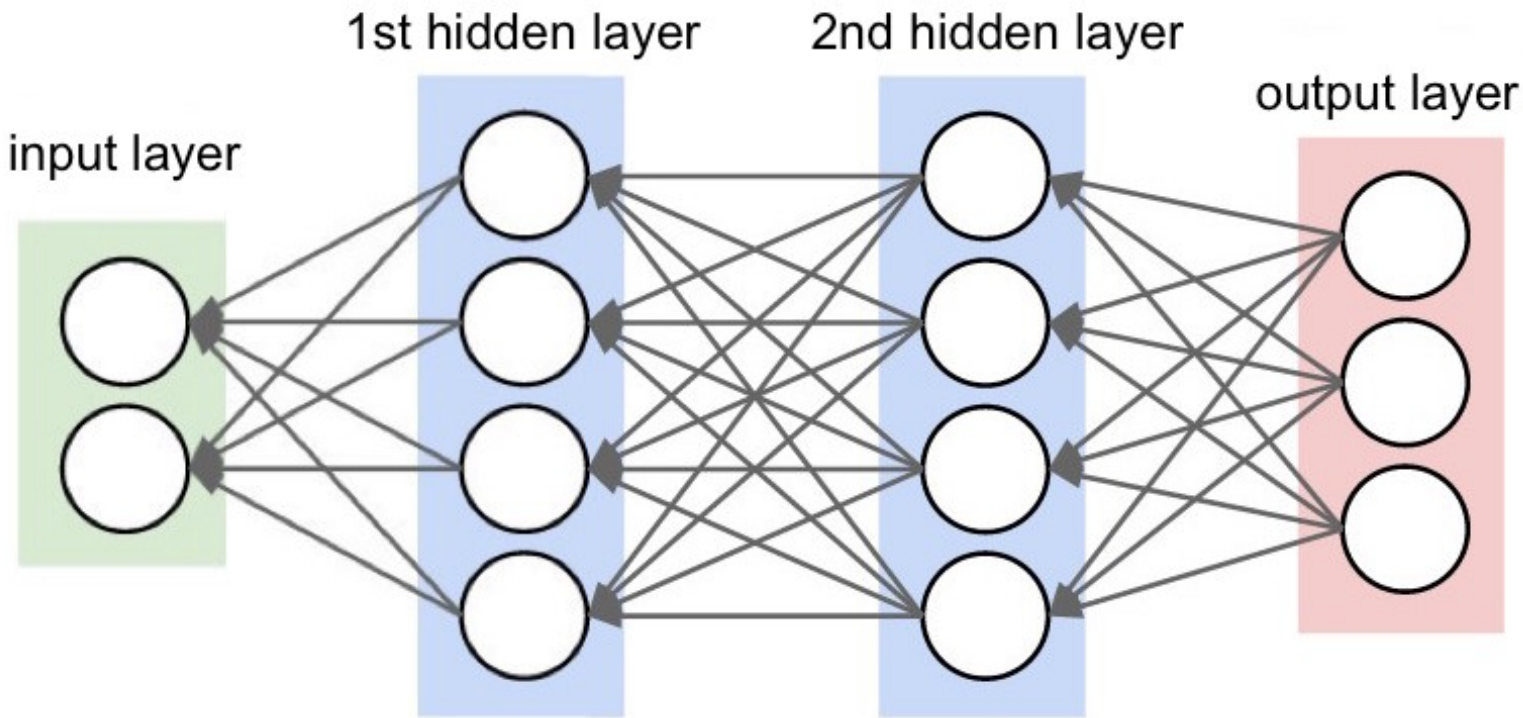
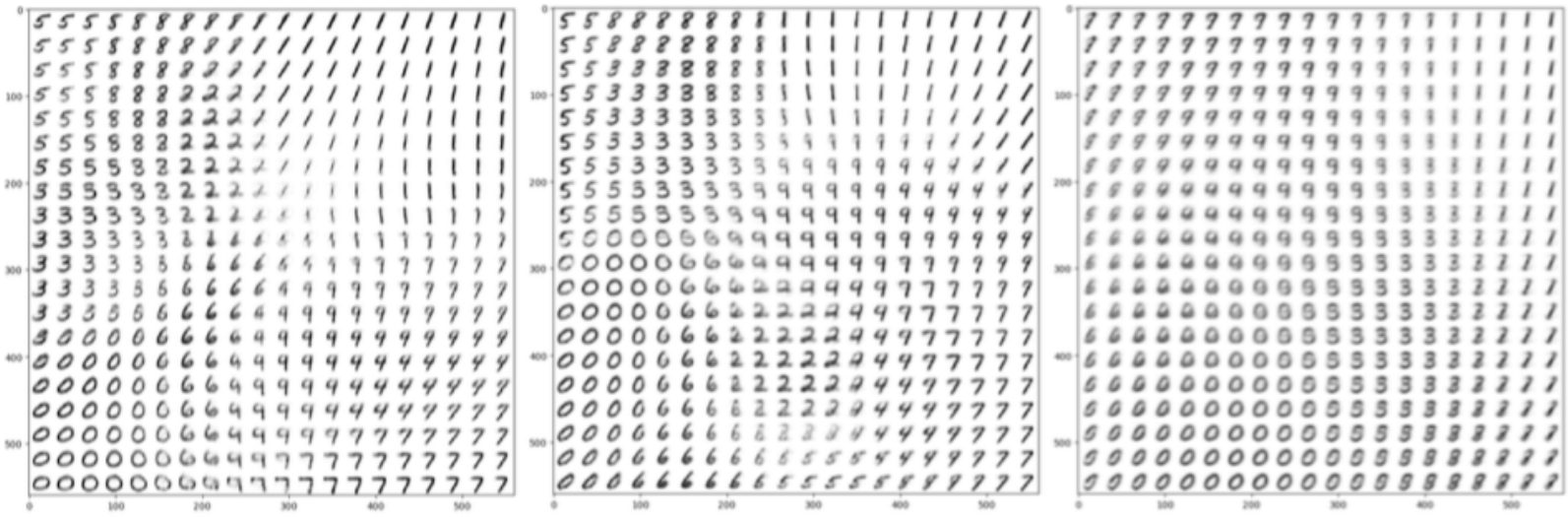


Figure: Fully connected NN with 2 hidden layers

Numerical Results on MNIST (L2 cost)



(a) $\varepsilon = 1, m = 200, L = 10$ (b) $\varepsilon = 10^{-1}, m = 200, L = 100$ (c) $\varepsilon = 10^{-1}, m = 10, L = 300$

Figure: Manifolds in the latent space for various parameters

Learning the cost [Li et al. '17, Bellemare et al. '17]

- On complex data sets, choice of a good ground metric c is not trivial
- Use parametric cost function $c_\phi(x, y) = \|f_\phi(x) - f_\phi(y)\|_2^2$ (where $f_\phi : \mathcal{X} \to \mathbb{R}^d$)
- Optimization problem becomes minmax (like GANs)

$$min_\theta max_\phi \bar{W}_{c_\phi,\varepsilon}(\mu_\theta, \nu)$$

- Same approximations but alternate between updating the cost parameters ϕ and the measure parameters θ

Numerical Results on CIFAR (learning the cost)

Figure: Samples from CIFAR dataset

Numerical Results on CIFAR (learning the cost)

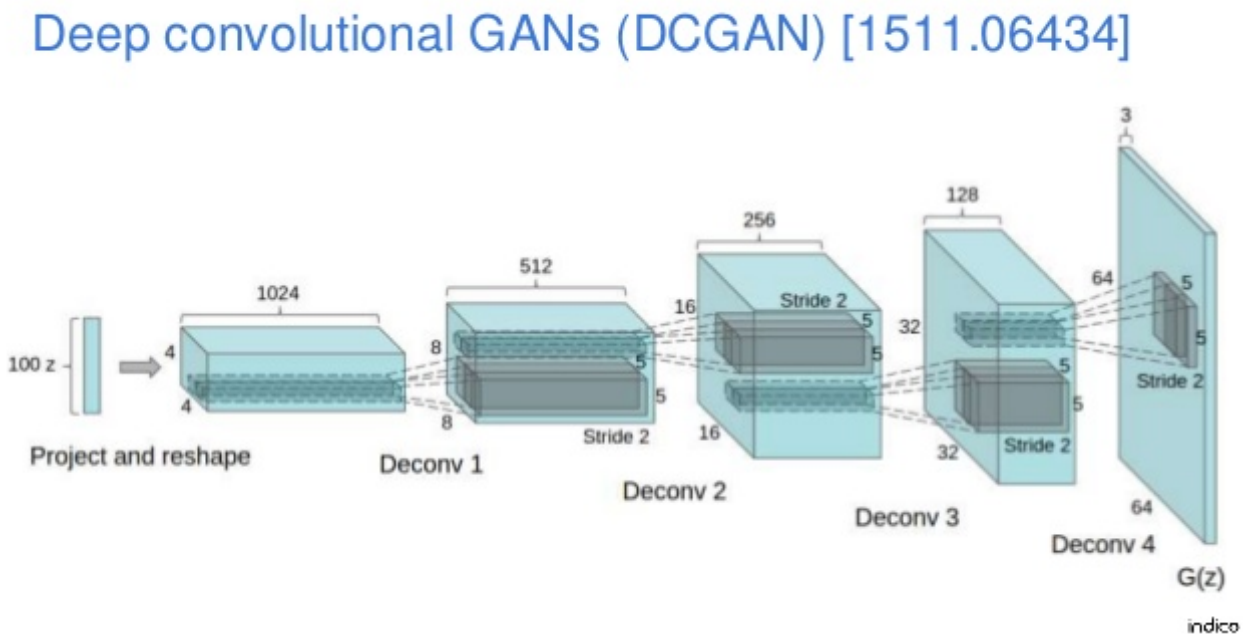
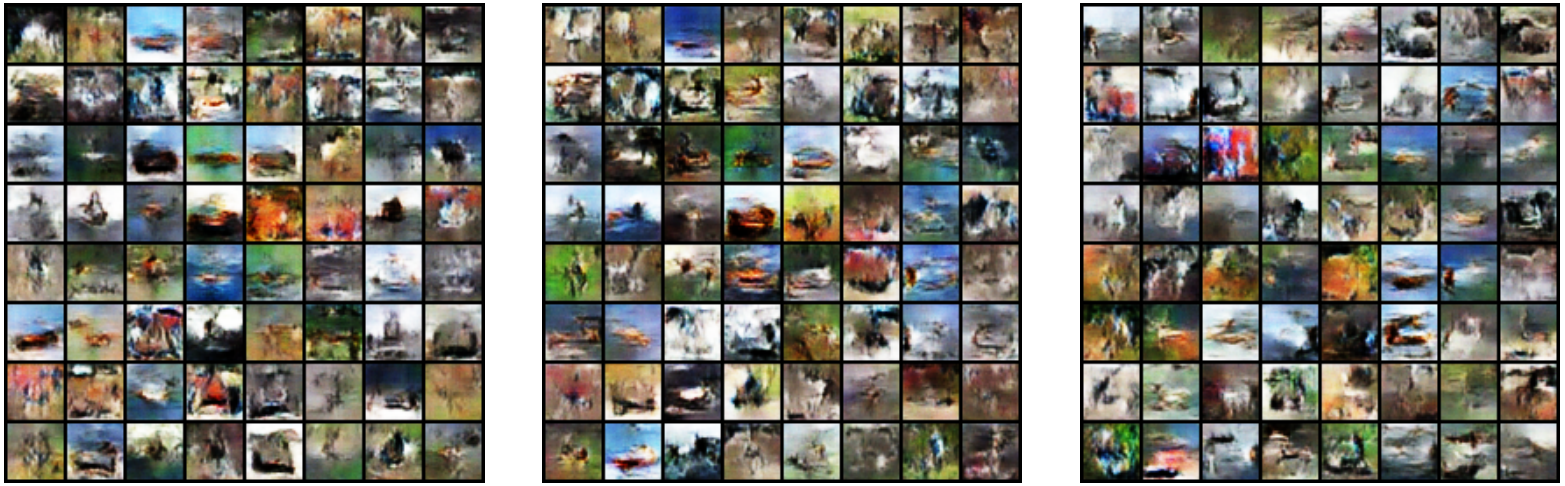


Figure: Fully connected NN with 2 hidden layers

Numerical Results on CIFAR (learning the cost)



(a) MMD (b) $\varepsilon = 1000$ (c) $\varepsilon = 10$

Figure: Samples from the generator trained on CIFAR 10 for MMD and Sinkhorn loss (coming from the same samples in the latent space)

Which is better? Not just about generating nice images, but more about capturing a high dimensional distribution... Hard to evaluate.