

# 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
- 4 Application to Generative Models

## Shortcomings of OT

Two main issues when using OT in practice :

- Poor sample complexity : need a lot of samples from  $\mu$  and  $\nu$  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

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

where

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

## 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 :

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

First order conditions for each variable:

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

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

 $\Rightarrow$  Do alternate maximizations!



## Sinkhorn's Algorithm

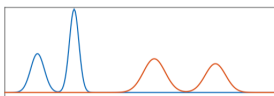
- Iterates  $(a, b) := (e^{\frac{u}{\epsilon}}, e^{\frac{v}{\epsilon}})$

## Sinkhorn algorithm [Cuturi '13]

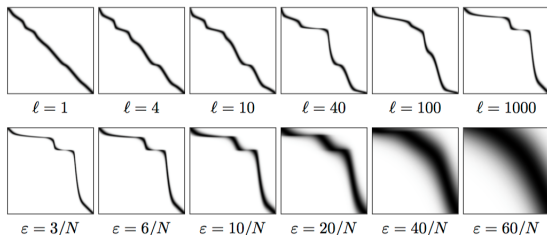
```
initialize  $b \leftarrow \mathbb{1}_m$     $K \leftarrow (e^{-c_{ij}/\epsilon} m_{ij})_{ij}$   
repeat  
     $a \leftarrow \mu \oslash K b$   
     $b \leftarrow \nu \oslash K^T a$   
return  $\gamma = \text{diag}(a) K \text{diag}(b)$ 
```

- 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  $\mu$  and  $\nu$



top : evolution of  $\gamma$  with number of iterations  $l$   
 bottom : evolution of  $\gamma$  with regularization parameter  $\varepsilon$

## 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  $\varepsilon \rightarrow 0$  (exponentially in worst-case)

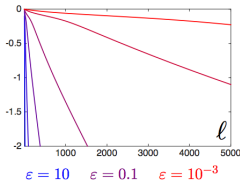
## Sinkhorn - Convergence

## Constraint violation

We have the following bound on the iterates:

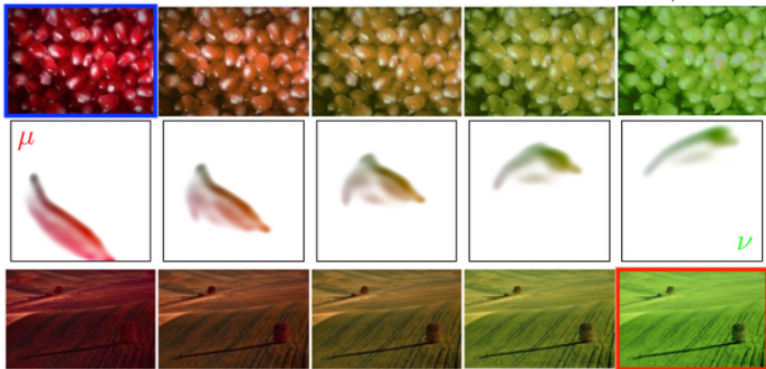
$$d_H(a^{(l)}, a^*) \leq \kappa d_H(\gamma \mathbb{1}_m, \mu)$$

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

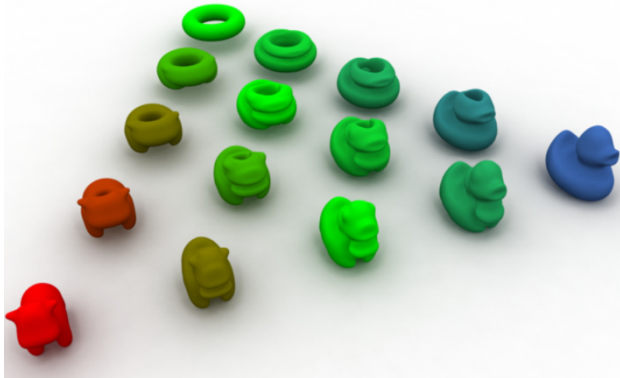


*Image courtesy of G. Peyré*

# Shape / Image Barycenters

Regularized Wasserstein Barycenters [Nenna et al. '15]

$$\bar{\mu} = \arg \min_{\mu \in \Sigma_n} W_\varepsilon(\mu_k, \mu)$$



Consider entropy-regularized OT

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

Regularized loss :

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

where  $\pi_{\varepsilon}$  solution of (15)

# Sinkhorn Divergences : interpolation between OT and MMD

## Theorem

The Sinkhorn loss between two measures  $\mu, \nu$  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  $\varepsilon$ :

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

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  $\varepsilon \rightarrow \infty$ . In particular,  $c = \|\cdot\|_p^p$ ,  $0 < p < 2$  is valid.



# Sample Complexity

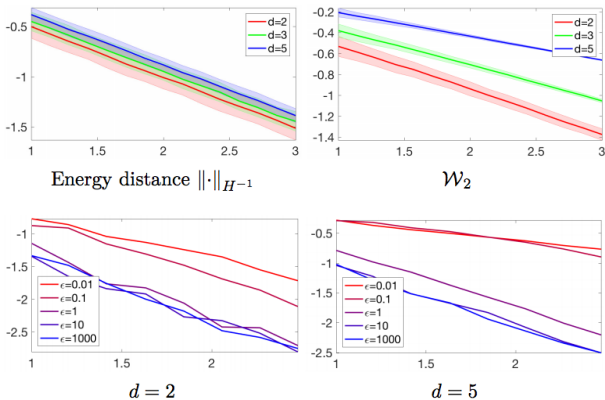
## Sample Complexity of OT and MMD

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

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

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

# Sample Complexity - Sinkhorn loss



Sample Complexity of Sinkhorn loss seems to improve as  $\epsilon$  grows.

## Generative Models

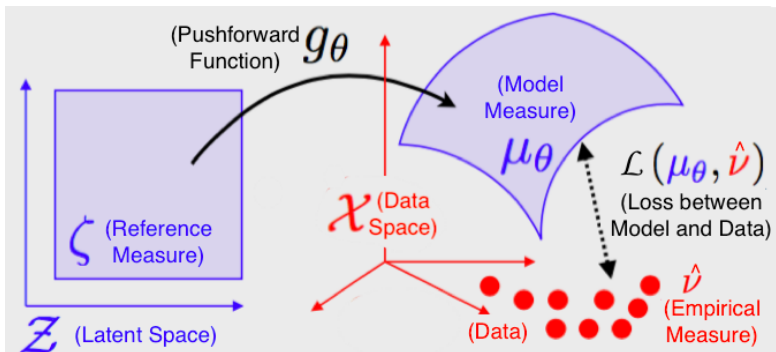


Figure: Illustration of Density Fitting on a Generative Model

# Density Fitting with Sinkhorn loss "Formally"

$$\text{Solve } \min_{\theta} E(\theta)$$

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

⇒ 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  $\mu_\theta$
  - 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

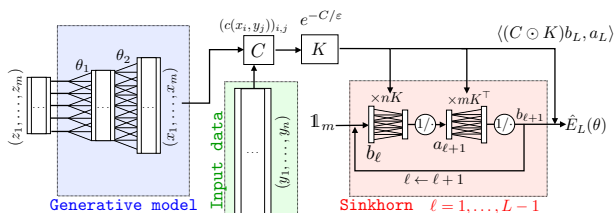


Figure: Scheme of the loss approximation

- 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)

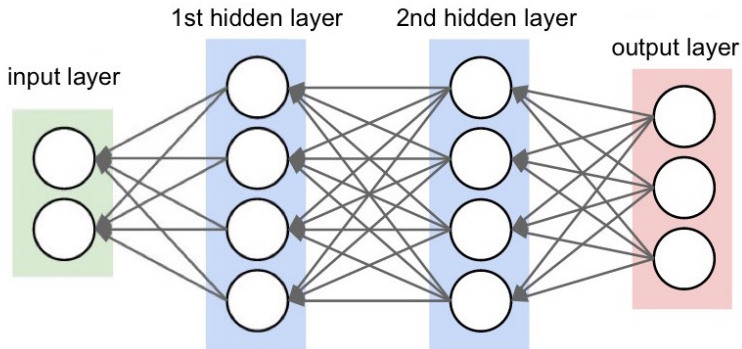


Figure: Fully connected NN with 2 hidden layers



Optimal  
transport

Aude  
Genevay

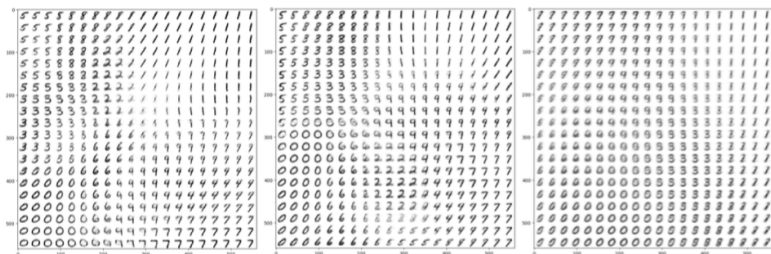
Entropy  
Regularized  
OT

Applications  
in Imaging

Large Scale  
"OT" for  
Machine  
Learning

Application  
to Generative  
Models

# 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  $f_\phi : \mathcal{X} \rightarrow \mathbb{R}^d$ )
- 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  $\phi$  and the measure parameters  $\theta$

# Numerical Results on CIFAR (learning the cost)



Figure: Samples from CIFAR dataset

# Numerical Results on CIFAR (learning the cost)

## Deep convolutional GANs (DCGAN) [1511.06434]

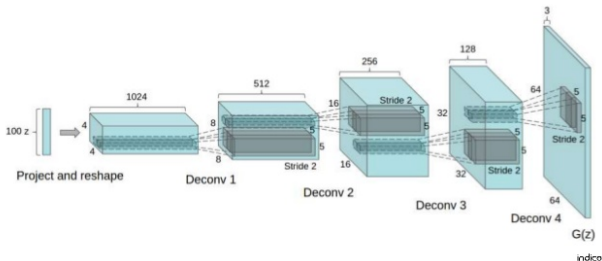
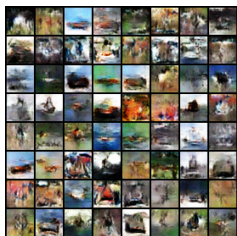


Figure: Fully connected NN with 2 hidden layers

Numerical Results on CIFAR  
(learning the cost)

(a) MMD

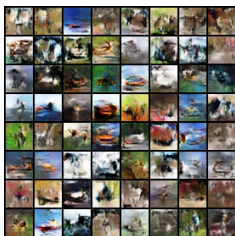
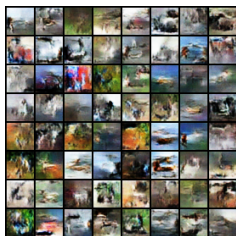
(b)  $\varepsilon = 1000$ (c)  $\varepsilon = 10$ 

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.

Shape Registration [Feydy et al.  
'17]

Register a shape  $A$  to a shape  $B$   
through a (rigid, diffeomorphic, etc.)  
transformation  $\varphi$ :

$$A \longrightarrow \varphi(A) \simeq B.$$

Variational setting : minimize

$$E(\varphi) = \underbrace{\text{Reg}(\varphi)}_{\text{Regularization}} + \underbrace{d(\varphi(A) \rightarrow B)}_{\text{fidelity term}}.$$