Optimal Transport for Machine Learning

Aude Genevay

CEREMADE - Université Paris Dauphine
INRIA - Mokaplan project-team
DMA - Ecole Normale Supérieure

Rentrée de l'équipe d'analyse - DMA - 3 Octobre 2017

Joint work with F. Bach, M.Cuturi, G. Peyré

Recurrent issue in ML : Comparing probability distributions

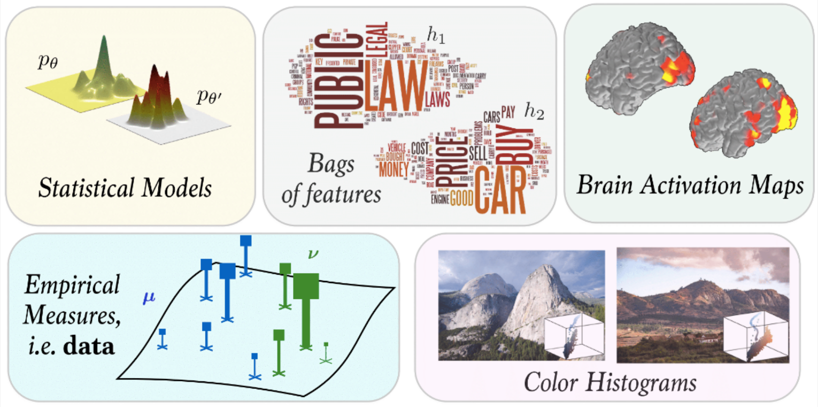
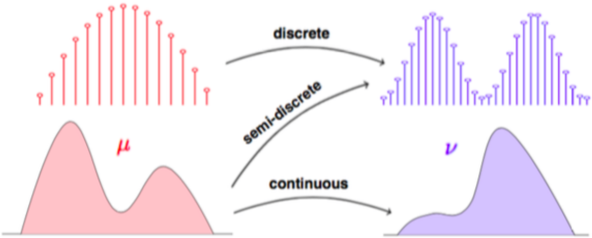


Figure 1: Many objects can be viewed as probability distributions (courtesy of M. Cuturi)

Optimal Transport and the Wasserstein Distance



- Optimal Transport : find coupling that minimizes total cost of moving μ to ν whith unit cost function **c**
- Constrained problem : coupling has fixed marginals
- Minimal cost of moving μ to ν(e.g. solution of the OT problem) is called the **Wasserstein distance** (it's an actual distance!)

OT for ML problems

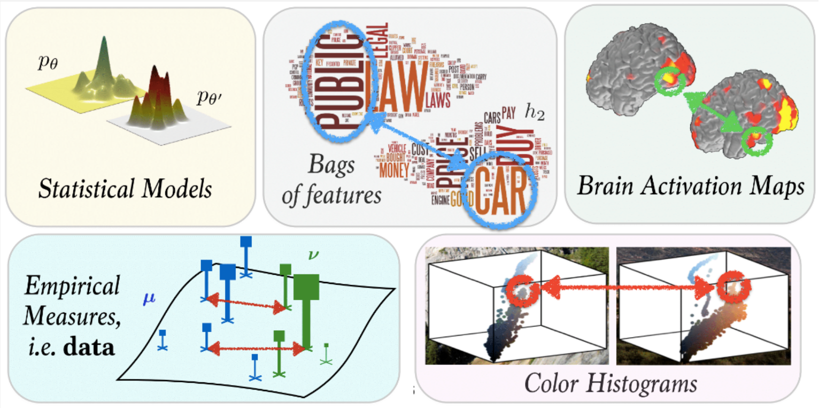


Figure 2: OT gives a natural framework for distances between probability distributions that takes geometry into account (courtesy of M. Cuturi)

Optimal Transport

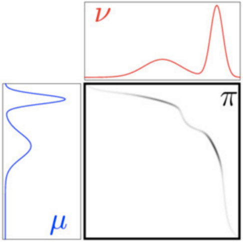
Two positive Radon measures μ on $\mathcal{X}$ and ν on $\mathcal{Y}$ of mass 1

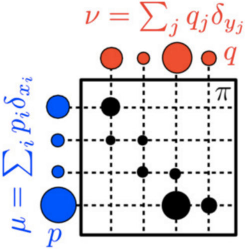
Cost $c(x, y)$ to move a unit of mass from x to y

Set of couplings with marginals μ and ν

$$\Pi(\mu,\nu) \stackrel{\text{def.}}{=} \{\pi \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y}) \mid \pi(A \times \mathcal{Y}) = \mu(A), \pi(\mathcal{X} \times B) = \nu(B)\}$$

What's the coupling that minimizes the total cost?





Kantorovitch Formulation of OT

The optimal overall cost for transporting μ to ν is given by

$$W(\mu, \nu) = \min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x, y)\mathrm{d}\pi(x, y) \qquad (\mathcal{P}_\varepsilon)$$

Kantorovitch Formulation of OT

The optimal overall cost for transporting μ to ν is given by

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

where

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

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 alternate maximizations on the dual variables : Sinkhorn's algorithm (more on that later)
- For ML applications, regularized Wasserstein is better than standard one
- In high dimension, helps avoiding overfitting

Reminder on convex duality

Primal problem:

$$\begin{array}{ll} \min_x & f(x) \\ \text{subject to} & h_i(x) = 0 \quad \text{for i} = 1 \ldots \text{m} \end{array}$$

Lagrange dual function:

$$g(\lambda) = \min_x f(x) + \sum_{i=1}^{m} \lambda_i h_i(x)$$

Dual problem:

$$\max_\lambda g(\lambda)$$

Under good assumptions, both problems are equivalent.

Dual formulation of OT

$$W(\mu, \nu) = \max_{u \in \mathcal{C}(\mathcal{X}), v \in \mathcal{C}(\mathcal{Y})} \int_{\mathcal{X}} u(x) \mathrm{d}\mu(x) + \int_{\mathcal{Y}} v(y) \mathrm{d}\nu(y) - \iota_{U_c}(u, v) \quad (\mathcal{D}_\varepsilon)$$

where the constraint set U_c is defined by

$$U_c \stackrel{\text{def.}}{=} \{(u, v) \in \mathcal{C}(\mathcal{X}) \times \mathcal{C}(\mathcal{Y}) \; ; \; \forall (x, y) \in \mathcal{X} \times \mathcal{Y}, u(x) + v(y) \leq c(x, y)\}$$

Dual formulation of OT (with entropy)

$$W_\varepsilon(\mu, \nu) = \max_{u \in \mathcal{C}(\mathcal{X}), v \in \mathcal{C}(\mathcal{Y})} \int_{\mathcal{X}} u(x) \mathrm{d}\mu(x) + \int_{\mathcal{Y}} v(y) \mathrm{d}\nu(y) - \iota^{\varepsilon}_{U_c}(u, v)$$

and the smoothed indicator is

$$\iota^{\varepsilon}_{U_c}(u, v) \stackrel{\text{def.}}{=} \varepsilon \int_{\mathcal{X} \times \mathcal{Y}} \exp\left(\frac{u(x) + v(y) - c(x, y)}{\varepsilon}\right) \mathrm{d}\mu(x) \mathrm{d}\nu(y)$$

Semi-Dual formulation of OT

The dual problem is convex in u and v. We fix v and minimize over u.

Semi-Dual formulation of OT

The dual problem is convex in u and v. We fix v and minimize over u. This yields :

$$u(x) \stackrel{\text{def.}}{=} \min_{y \in \mathcal{Y}} c(x, y) - v(y)$$

Semi-Dual formulation of OT

The dual problem is convex in u and v. We fix v and minimize over u. This yields :

$$u(x) \stackrel{\text{def.}}{=} \min_{y \in \mathcal{Y}} c(x,y) - v(y)$$

Plugging back in the dual :

$$\begin{aligned} W_\varepsilon(\mu,\nu) &= \max_{v \in \mathcal{C}(\mathcal{Y})} \int_{\mathcal{X}} \min_{y \in \mathcal{Y}} \left(c(x,y) - v(y)\right) \mathrm{d}\mu(x) + \int_{\mathcal{Y}} v(y) \mathrm{d}\nu(y) - \varepsilon \\ &= \max_{v \in \mathcal{C}(\mathcal{Y})} \mathbb{E}_\mu \Big[\min_{y \in \mathcal{Y}} \left(c(x,y) - v(y)\right) + \int_{\mathcal{Y}} v(y) \mathrm{d}\nu(y) - \varepsilon \Big] \end{aligned}$$

Semi-Dual formulation of OT (with entropy)

The dual problem is convex in u and v. We fix v and minimize over u.

Semi-Dual formulation of OT (with entropy)

The dual problem is convex in u and v. We fix v and minimize over u. This yields :

$$u(x) \stackrel{\text{def.}}{=} -\varepsilon \log \left(\int_{\mathcal{Y}} \exp\left(\frac{v(y) - c(x,y)}{\varepsilon}\right) \mathrm{d}\nu(y) \right)$$

Semi-Dual formulation of OT (with entropy)

The dual problem is convex in u and v. We fix v and minimize over u. This yields :

$$u(x) \stackrel{\text{def.}}{=} -\varepsilon \log\left(\int_{\mathcal{Y}} \exp\left(\frac{v(y) - c(x,y)}{\varepsilon}\right) \mathrm{d}\nu(y)\right)$$

Plugging back in the dual :

$$\begin{aligned} W_\varepsilon(\mu,\nu) &= \max_{v \in \mathcal{C}(\mathcal{Y})} \int_{\mathcal{X}} -\varepsilon \log\left(\int_{\mathcal{Y}} \exp\left(\frac{v(y) - c(x,y)}{\varepsilon}\right) \mathrm{d}\nu(y)\right) \mathrm{d}\mu(y) \\ &\quad + \int_{\mathcal{Y}} v(y) \mathrm{d}\nu(y) - \varepsilon \\ &= \max_{v \in \mathcal{C}(\mathcal{Y})} \mathbb{E}_\mu\Big[-\varepsilon \log\left(\int_{\mathcal{Y}} \exp\left(\frac{v(y) - c(x,y)}{\varepsilon}\right)\right) \\ &\quad + \int_{\mathcal{Y}} v(y) \mathrm{d}\nu(y) - \varepsilon \Big] \end{aligned}$$

We consider 2 frameworks :

- Semi-Discrete : μ is continuous and $\nu = \sum_{j=1}^{M} \nu_i \delta y_j$ The optimization problem is

$$\max_{v \in \mathbb{R}^M} \mathbb{E}_\mu \left[-\varepsilon \log \left(\sum_{j=1}^{M} \exp\left(\frac{v(y_j) - c(x, y_j)}{\varepsilon} \right) \right) + \sum_{j=1}^{M} v(y_j)\nu_j - \varepsilon \right]$$

We consider 2 frameworks :

- Semi-Discrete : μ is continuous and $\nu = \sum_{j=1}^{M} \nu_i \delta y_j$ The optimization problem is

$$\max_{v \in \mathbb{R}^M} \mathbb{E}_\mu \left[-\varepsilon \log \left(\sum_{j=1}^{M} \exp(\frac{v(y_j) - c(x, y_j)}{\varepsilon}) \right) + \sum_{j=1}^{M} v(y_j) \nu_j - \varepsilon \right]$$

- Discrete : $\mu = \sum_{i=1}^{N} \mu_i \delta x_i$ and $\nu = \sum_{j=1}^{M} \nu_i \delta y_j$ The optimization problem is

$$\max_{v \in \mathbb{R}^M} \sum_{i=1}^{N} \left[-\varepsilon \log \left(\sum_{j=1}^{M} \exp(\frac{v(y_j) - c(x_i, y_j)}{\varepsilon}) \right) + \sum_{j=1}^{M} v(y_j) \nu_j - \varepsilon \right] \mu_i$$

Stochastic Optimization

Computing the full gradient is

- Hard in the semi-discrete setting (even impossible if we don't know μ explicitly)

Stochastic Optimization

Computing the full gradient is

- Hard in the semi-discrete setting (even impossible if we don't know μ explicitly)
- Very costly in the discrete case since we need to compute N gradients and sum them.

The idea of stochastic optimization is to use approximate gradients so that each iteration is inexpensive.

Stochastic Optimization I

- Goal : maximize $H_\varepsilon(v) = \mathbb{E}_\mu\left[h_\varepsilon(X, v)\right]$ over v in $\mathbb{R}^M$.
- Standard gradient ascent :

$$v^{(k)} = v^{(k-1)} + \nabla_v H_\varepsilon(v^{(k-1)})$$

- The whole gradient $\nabla_v H_\varepsilon(v)$ is too costly/complicated to compute
- Idea : Sample x from μ and use $\nabla_v h_\varepsilon(x, v)$ as a proxy for the full gradient in the gradient ascent.

Stochastic Optimization II

Algorithm 1 Averaged SGD

Input: C

Output: v

$v \leftarrow \mathbb{0}_M$, $\bar{v} \leftarrow v$

for $k = 1, 2, \ldots$ **do**

Sample x_k from μ

$v \leftarrow v + \frac{C}{\sqrt{k}} \nabla_v h_\varepsilon(x_k, v)$ (gradient ascent step)

$\bar{v} \leftarrow \frac{1}{k} v + \frac{k-1}{k} \bar{v}$ (averaging)

end for

- cost of each iteration M
- convergence rate $O(1/\sqrt{(}k))$

Discrete OT : Sinkhorn's Algorithm I

State-of-the-art : Sinkhorn's Algorithm

- Two equivalent views
  - Alternate projections on the constraints of the primal
  - Alternate minimizations on the dual
- Iterates $a \overset{\text{def.}}{=} \exp(\frac{u}{\varepsilon})$ and $b \overset{\text{def.}}{=} \exp(\frac{v}{\varepsilon})$: $\begin{cases} a = \frac{1}{K(b \odot \nu)} \\ b = \frac{1}{K^T(a \odot \mu)} \end{cases}$

  where $K \overset{\text{def.}}{=} \exp \frac{-\mathbf{c}}{\varepsilon}$ and $\odot$ is coordinatewise vector multiplication.
- Linear convergence of the iterates to the optimizers

Discrete OT : Sinkhorn's Algorithm II

Algorithm 2 Sinkhorn

Output: $\mathbf{v}$

$\mathbf{b} \leftarrow \mathbb{1}_J$

for $k = 1, 2, \dots$ **do**

$\mathbf{a} \leftarrow \frac{\mathbb{1}_I}{K(\boldsymbol{\nu} \odot \mathbf{b})}$

$\mathbf{b} \leftarrow \frac{\mathbb{1}_J}{K^\top(\boldsymbol{\mu} \odot \mathbf{a})}$

end for

$\mathbf{v} \leftarrow \varepsilon \log(\mathbf{b})$

$\Rightarrow$ Implies matrix vector multiplications at each iteration : cost $I \times J$ per iteration

Stochastic Optimization : Case of a Finite Sum I

When μ is also a discrete measure, we are minimizing a finite sum of N functionals :

$$\max_{v \in \mathbb{R}^M} \sum_{i=1}^{N} \left[-\varepsilon \log \left(\sum_{j=1}^{M} \exp\left(\frac{v(y_j) - c(x_i, y_j)}{\varepsilon}\right) \right) + \sum_{j=1}^{M} v(y_j)\nu_j - \varepsilon \right] \mu_i$$

Stochastic Optimization : Case of a Finite Sum II

A more efficient stochastic algorithm consists in using an average of the past gradients as a proxy for the full gradient :

- At iteration k, an index i is drawn. Its gradient $\nabla_v h_\varepsilon(x_i, v^{(k)})$ is updated in the vector of partial gradients (vector with N entries kept in memory).
- The average gradient is updated accordingly, and used in a step of the gradient ascent

Stochastic Optimization : Case of a Finite Sum III

Algorithm 3 SAG for Discrete OT

Input: C

Output: $\mathbf{v}$

$\mathbf{v} \leftarrow \mathbb{0}_M$, $\mathbf{d} \leftarrow \mathbb{0}_J$, $\forall i, \mathbf{g}_i \leftarrow \mathbb{0}_M$

for $k = 1, 2, \ldots$ **do**

  Sample $i \in \{1, 2, \ldots, I\}$ uniform.

  $\mathbf{d} \leftarrow \mathbf{d} - \mathbf{g}_i$

  $\mathbf{g}_i \leftarrow \boldsymbol{\mu}_i \nabla_v \bar{h}_\varepsilon(x_i, \mathbf{v})$

  $\mathbf{d} \leftarrow \mathbf{d} + \mathbf{g}_i$; $\mathbf{v} \leftarrow \mathbf{v} + C\mathbf{d}$

end for

- cost of each iteration M
- convergence rate $O(1/k)$

Stochastic Optimization : Case of a Finite Sum IV

$\Rightarrow$ Slower convergence rate than Sinkhorn but *online* algorithm, better for (very) large-scale problems

Numerical Results for Word Mover's Distance (Discrete OT)

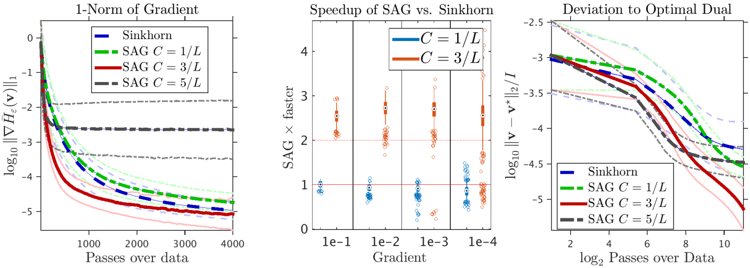
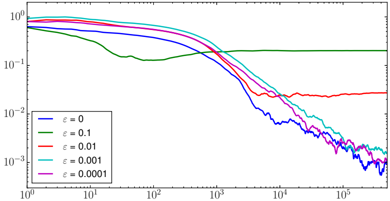
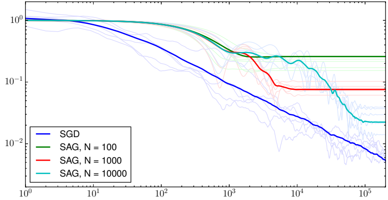


Figure 3: Results for the computation of 595 pairwise word mover's distances between 35 very large corpora of text, each represented as a cloud of $I = 20,000$ word embeddings.

Numerical Results for Density Fitting (Semi-discrete OT)



(a) SGD



(b) SGD vs. SAG

Figure 4: (a) Plot of $\|\mathbf{v}_k - \mathbf{v}_0^\star\|_2 \,/\, \|\mathbf{v}_0^\star\|_2$ as a function of k, for SGD and different values of ε ($\varepsilon = 0$ being un-regularized). (b) Plot of $\|\mathbf{v}_k - \mathbf{v}_\varepsilon^\star\|_2 \,/\, \|\mathbf{v}_\varepsilon^\star\|_2$ averaged over 40 runs as a function of k, for SGD and SAG with different number N of samples, for regularized OT using $\varepsilon = 10^{-2}$.

Density Fitting : A natural semi-discrete OT problem I

- Data $(y_1, \ldots, y_N) \in \mathcal{X}$
- Empirical measure $\nu = \frac{1}{N} \sum_{i=1}^N \delta_{y_i}$
- Parametric model : $\mu_\theta = g_{\theta\sharp}\xi$
- ξ is a reference measure on the latent space $\mathcal{Z}$
- $g_\theta : \mathcal{Z} \to \mathcal{X}$ from latent space to data space
- i.e. a sample x from the generative model is obtained by $x = g_\theta(z)$ were $z \sim \xi$
- Goal : find $\hat{\theta} = \arg\min_\theta \mathcal{L}(\mu_\theta, \nu)$ where $\mathcal{L}$ is a loss on measures.

Density Fitting : A natural semi-discrete OT problem II

The Wasserstein Distance is a natural candidate for $\mathcal{L}$!

- because of its good geometrical properties
- but also because the usual maximum likelihood framework using $\mathcal{L} = -\sum_j \log \frac{d\mu_\theta}{dx}(y_j)$ can't be used (not defined for singular measures without a density wrt a reference measure)

Density Fitting with the Wasserstein Distance

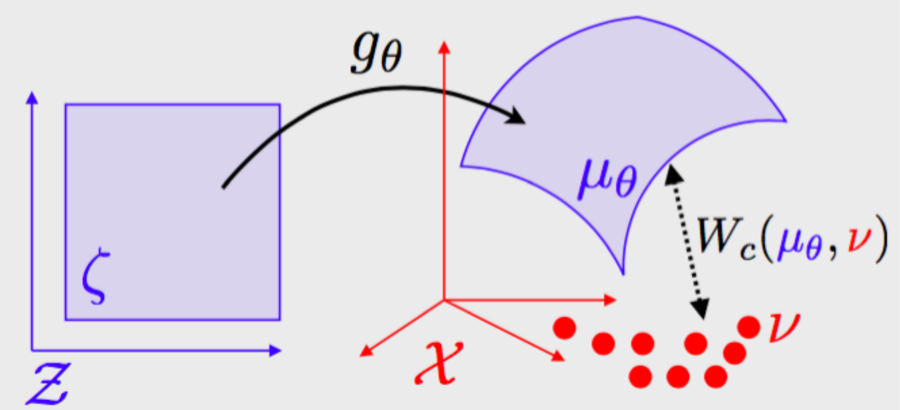


Figure 5: Illustration of the Minimum Kantorovitch Estimation

Density Fitting with the Wasserstein Distance "Formally"

- Solve $\min_\theta E(\theta)$
  where $E(\theta) = \mathcal{W}_c(\mu_\theta, \nu) = \min_{\Pi(\xi,\nu)} \int_{\mathcal{Z},\mathcal{X}} c(g_\theta(z), x) d\pi(z, x)$
- gradient reads $\nabla E(\theta) = \int_{\mathcal{Z}\times\mathcal{X}} (\partial_\theta g_\theta(z))^\top (\nabla_1 c(g_\theta(z), y) \mathrm{d}\pi(z, y)$
  where π is a minimizer of $\mathcal{W}_c$
- intractable in pratice, need an approximation

Approximating the gradient?

- Actually, rather than approximating the gradient approximate the distance
- Minibatches : $\hat{E}(\theta)$
  - sample $x_1, \dots, x_m$ from μ_θ
  - use empirical Wasserstein distance $\mathcal{W}(\hat{\mu}_\theta, \nu)$ where $\hat{\mu}_\theta = \frac{1}{N}\sum_{i=1}^m \delta_{x_i}$
- Regularize (with entropy) and use Sinkhorn's algorithm : $\hat{E}_L(\theta)$
  - state of the art approximation algorithm for discrete OT
  - each step simple matrix/vector multiplication
  - compute L steps of the algorithm
  - use this as a proxy for $\mathcal{W}(\hat{\mu}_\theta, \nu)$
- Compute *exact* gradient of $\hat{E}_L(\theta)$ with autodiff (in tensorflow)

The Inference Algorithm in Practice

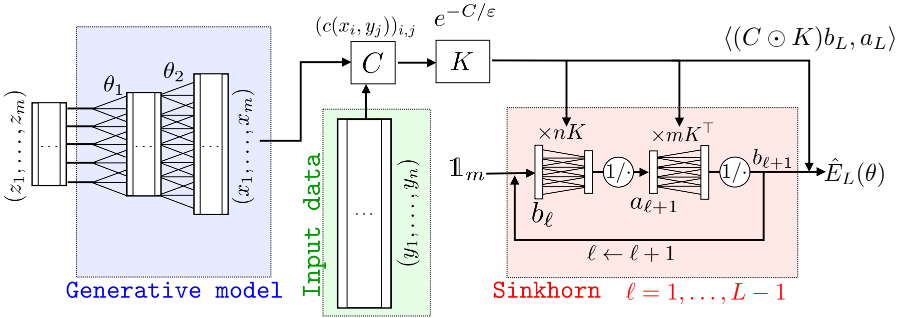
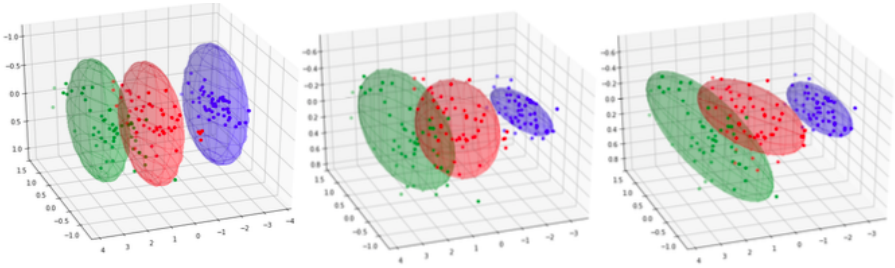


Figure 6: Scheme of the inference prodecure



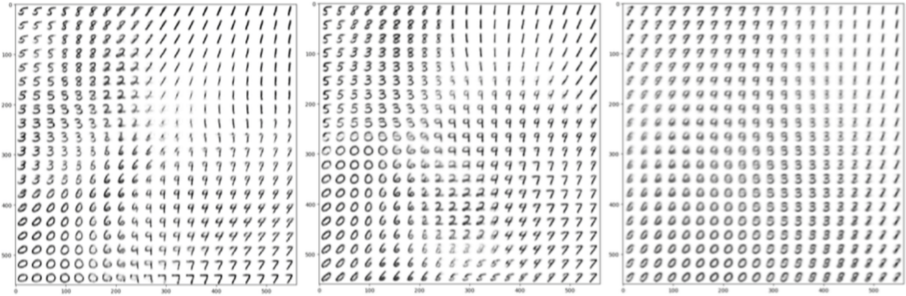
(a) Initialization (unit balls, kmeans centers)

(b) After 3 gradient steps

(c) At convergence (15 steps)

Figure 7: Fitting ellipses (centers and covariance matrices) to the IRIS dataset

Numerical Results on MNIST



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

Figure 8: Manifolds in the latent space for various parameters

Conclusion

- Similar frameworks : WGAN (Arjovsky et al.) / VEGAN (Bousquet et al.)
- Competing concept : Cramer-GAN (Bellemare et al.) using MMD (Maximum Mean Discrepency) kernel-based distance instead of Wasserstein distance
- Actually, when using regularized Wasserstein
  - $\varepsilon \to 0$: standard OT
  - $\varepsilon \to \infty$: MMD
- Ongoing work to explore the effects of interpolating between both with ε in-between