

(Almost)-Optimal transport

Matthieu Heitz, Agathe Herrou,
Jocelyn Meyron, Loïs Paulin

October 28th, 2019

Outline

Introduction

Continuous and discrete

- Sinkhorn algorithm

- Sinkhorn barycenter

Semi-discrete setting

- Geometric formulation

- Numerical methods

- Applications

Sliced Optimal Transport

- Introduction

- Algorithms

- Applications

Outline

Introduction

Continuous and discrete

- Sinkhorn algorithm

- Sinkhorn barycenter

Semi-discrete setting

- Geometric formulation

- Numerical methods

- Applications

Sliced Optimal Transport

- Introduction

- Algorithms

- Applications

Introduction

- ▶ 18th century: Monge's *Mémoire sur la Théorie des Déblais et des Remblais*

Introduction

- ▶ 18th century: Monge's *Mémoire sur la Théorie des Déblais et des Remblais*
- ▶ 1940's: new formulation by Kantorovich

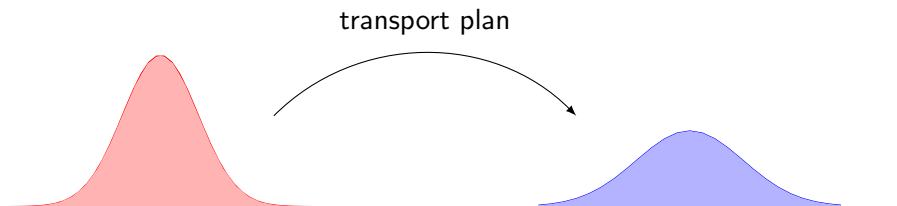
Introduction

- ▶ 18th century: Monge's *Mémoire sur la Théorie des Déblais et des Remblais*
- ▶ 1940's: new formulation by Kantorovich
- ▶ New developments lately: works of Y. Brenier on fluid mechanics

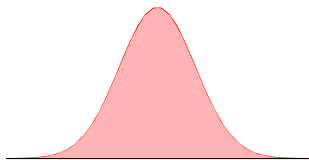
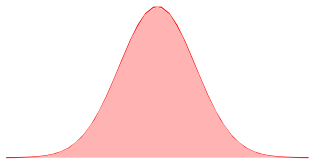
Introduction

- ▶ 18th century: Monge's *Mémoire sur la Théorie des Déblais et des Remblais*
- ▶ 1940's: new formulation by Kantorovich
- ▶ New developments lately: works of Y. Brenier on fluid mechanics
- ▶ Applications in
 - ▶ meteorology (semi-geostrophic equations)
 - ▶ statistical physics
 - ▶ fundamental mathematics (isoperimetric problems, Sobolev inequalities. . .)
 - ▶ computer graphics
 - ▶ . . .

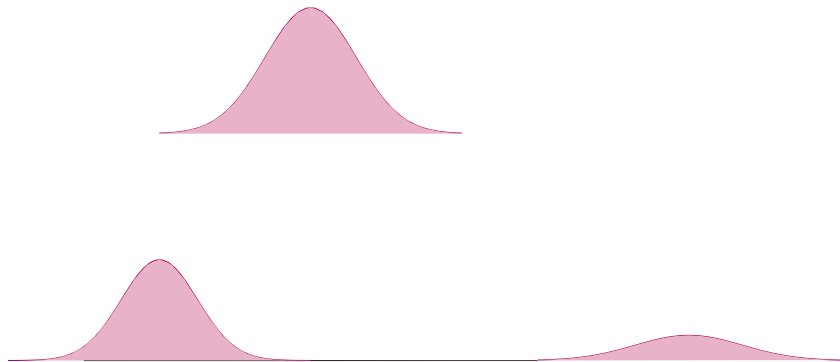
Sand pile



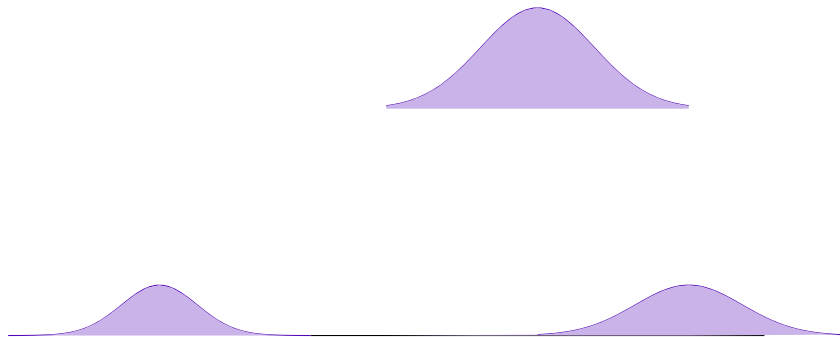
Sand pile



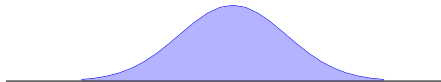
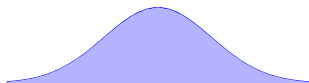
Sand pile



Sand pile



Sand pile



Monge's formulation

Let (X, μ) and (Y, ν) be two measured spaces, $c : X \times Y \rightarrow \mathbb{R}$ a cost function.

We are aiming at finding $T : X \rightarrow Y$ such that

$$\min_{T \text{ s.t. } \nu = T_{\#}\mu} \int_{x \in X} c(x, T(x)) d\mu(x)$$

where $T_{\#}\mu(A) = \mu(T^{-1}(A))$ is the pushforward of μ by T .

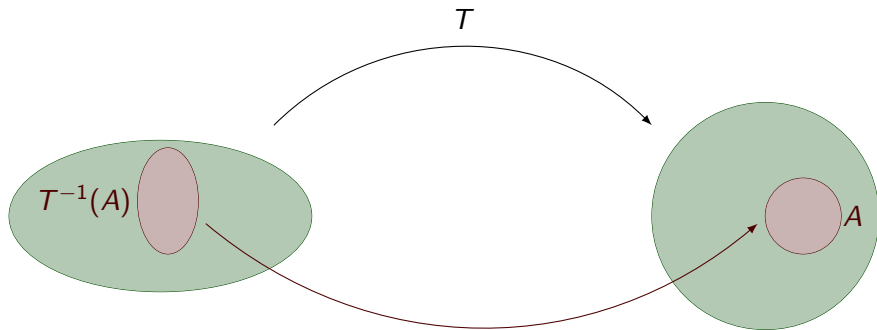
Monge's formulation

Let (X, μ) and (Y, ν) be two measured spaces, $c : X \times Y \rightarrow \mathbb{R}$ a cost function.

We are aiming at finding $T : X \rightarrow Y$ such that

$$\min_{T \text{ s.t. } \nu = T_{\#}\mu} \int_{x \in X} c(x, T(x)) d\mu(x)$$

where $T_{\#}\mu(A) = \mu(T^{-1}(A))$ is the pushforward of μ by T .



Kantorovitch's formulation

Find π realizing

$$\min_{(\mu, \nu) = \text{marginals of } \pi} \int_{X \times Y} c(x, y) d\pi(x, y)$$

where μ is the pushforward of π by $(x, y) \mapsto x$ and ν is the pushforward of π by $(x, y) \mapsto y$, i.e. $\mu(A) = \pi(A, Y)$ and $\nu(B) = \pi(X, B)$

Conveyor's principle

$$\inf_{\pi} \int c(x, y) d\pi(x, y) = \sup_{(\phi, \psi)} \int \phi d\mu + \int \psi d\nu$$

$$\text{s.t } \phi(x) + \psi(y) \leq c(x, y)$$

Fluid mechanics' formulation

- ▶ Introduces the time variable

Fluid mechanics' formulation

- ▶ Introduces the time variable
- ▶ Two density functions g_0 and g_1 describe the state of a fluid at initial and final times

Fluid mechanics' formulation

- ▶ Introduces the time variable
- ▶ Two density functions g_0 and g_1 describe the state of a fluid at initial and final times
- ▶ Unknown: function $g(x, t)$ describing the state of the fluid at all times

Fluid mechanics' formulation

- ▶ Introduces the time variable
- ▶ Two density functions g_0 and g_1 describe the state of a fluid at initial and final times
- ▶ Unknown: function $g(x, t)$ describing the state of the fluid at all times
- ▶ g minimizes the action locally in time $\Rightarrow$ formalized by a minimization problem

Fluid mechanics' formulation

- ▶ Introduces the time variable
- ▶ Two density functions g_0 and g_1 describe the state of a fluid at initial and final times
- ▶ Unknown: function $g(x, t)$ describing the state of the fluid at all times
- ▶ g minimizes the action locally in time $\Rightarrow$ formalized by a minimization problem
- ▶ Discretization: one minimization problem by discretization step

Fluid mechanics' formulation

- ▶ Introduces the time variable
- ▶ Two density functions g_0 and g_1 describe the state of a fluid at initial and final times
- ▶ Unknown: function $g(x, t)$ describing the state of the fluid at all times
- ▶ g minimizes the action locally in time $\Rightarrow$ formalized by a minimization problem
- ▶ Discretization: one minimization problem by discretization step
- ▶ These minimization problems can be expressed as Optimal Transport problems

Outline

Introduction

Continuous and discrete

- Sinkhorn algorithm

- Sinkhorn barycenter

Semi-discrete setting

- Geometric formulation

- Numerical methods

- Applications

Sliced Optimal Transport

- Introduction

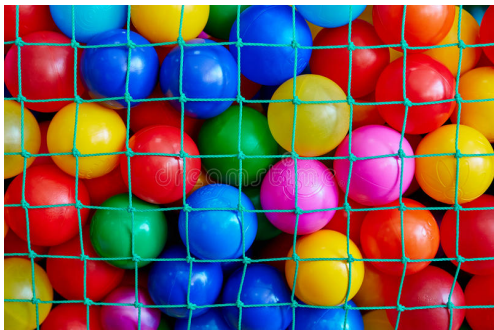
- Algorithms

- Applications

Two approaches

Analogy with fluid mechanics :

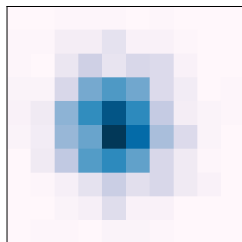
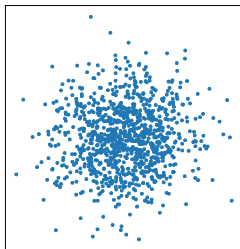
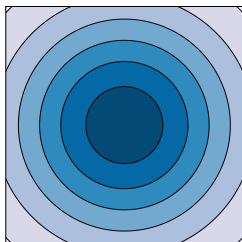
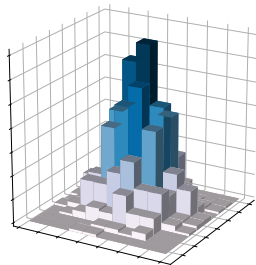
- ▶ **Lagrangian** approach:
Study individual particles.
- ▶ **Eulerian** approach:
Study fixed points of the domain.



Two approaches

Analogy with fluid mechanics :

- ▶ **Lagrangian** approach:
Study individual particles.
- ▶ **Eulerian** approach:
Study fixed points of the domain.



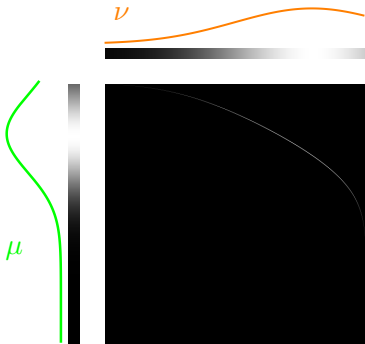
→ Discretization ?

Discretization

The Kantorovich problem

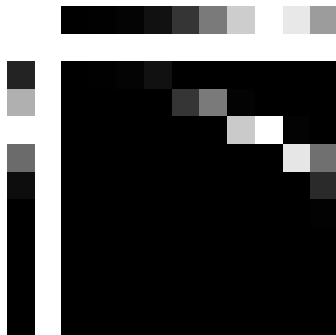
Continuous

$$\mathcal{L}(\mu, \nu) = \min_{\pi \in \Pi(\mu, \nu)} \int_{X \times Y} c(x, y) d\pi(x, y)$$



Discrete

$$L(\mathbf{a}, \mathbf{b}) \stackrel{\text{def.}}{=} \min_{\mathbf{P} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j} \mathbf{C}_{i,j} \mathbf{P}_{i,j}$$



Entropic regularization

Discrete Kantorovich problem

$$L(\mathbf{a}, \mathbf{b}) \stackrel{\text{def.}}{=} \min_{\mathbf{P} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j} \mathbf{C}_{i,j} \mathbf{P}_{i,j}$$

Entropic regularization

Discrete Kantorovich problem

$$L(\mathbf{a}, \mathbf{b}) \stackrel{\text{def.}}{=} \min_{\mathbf{P} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j} \mathbf{C}_{i,j} \mathbf{P}_{i,j}$$

$\Rightarrow$ solve it with a linear program: $O(N^3)$.

Entropy regularization [Cut13]

$$L_{\mathbf{C}}^{\varepsilon}(\mathbf{a}, \mathbf{b}) \stackrel{\text{def.}}{=} \min_{\mathbf{P} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j} \mathbf{C}_{i,j} \mathbf{P}_{i,j} - \varepsilon H(\mathbf{P}),$$

Entropic regularization

Discrete Kantorovich problem

$$L(\mathbf{a}, \mathbf{b}) \stackrel{\text{def.}}{=} \min_{\mathbf{P} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j} \mathbf{C}_{i,j} \mathbf{P}_{i,j}$$

$\Rightarrow$ solve it with a linear program: $O(N^3)$.

Entropy regularization [Cut13]

$$L_{\mathbf{C}}^{\varepsilon}(\mathbf{a}, \mathbf{b}) \stackrel{\text{def.}}{=} \min_{\mathbf{P} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j} \mathbf{C}_{i,j} \mathbf{P}_{i,j} - \varepsilon H(\mathbf{P}),$$

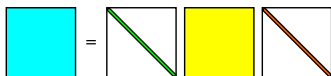
$H(\mathbf{P}) \stackrel{\text{def.}}{=} - \sum_{i,j} \mathbf{P}_{i,j} \log(\mathbf{P}_{i,j})$ is the discrete entropy (“dispersion”)



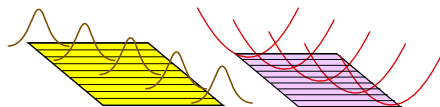
Sinkhorn algorithm

Decomposition of the transport plan $\mathbf{P}$:

$$\mathbf{P} = \text{diag}(\mathbf{u}) \mathbf{K} \text{diag}(\mathbf{v})$$



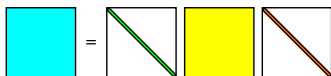
$$\mathbf{K} = \exp(-\mathbf{C}/\varepsilon)$$



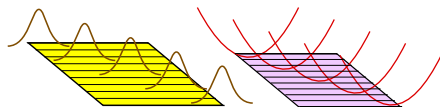
Sinkhorn algorithm

Decomposition of the transport plan $\mathbf{P}$:

$$\mathbf{P} = \text{diag}(\mathbf{u})\mathbf{K}\text{diag}(\mathbf{v})$$



$$\mathbf{K} = \exp(-\mathbf{C}/\varepsilon)$$



Marginal constraints:

$$\mathbf{P}\mathbf{1} = \mathbf{a}$$

$$\Rightarrow \mathbf{u} \odot \mathbf{K}\mathbf{v} = \mathbf{a}$$

$$\Rightarrow \mathbf{u}^{(l+1)} = \frac{\mathbf{a}}{\mathbf{K}\mathbf{v}^{(l)}}$$

$$\mathbf{P}^T\mathbf{1} = \mathbf{b}$$

$$\mathbf{v} \odot \mathbf{K}\mathbf{u} = \mathbf{b}$$

$$\mathbf{v}^{(l+1)} = \frac{\mathbf{b}}{\mathbf{K}^T\mathbf{u}^{(l+1)}}$$

Sinkhorn algorithm

The Sinkhorn algorithm:

Input:

Histograms $\mathbf{a}, \mathbf{b} \in \Sigma_N$,

kernel $\mathbf{K} = \exp(-\mathbf{C}/\varepsilon) \in \mathbb{R}^{N \times N}$,

number of iterations L

$$\mathbf{v} = \mathbf{1}_N$$

for $l = 1 \dots L$ **do**

$$\mathbf{u} = \mathbf{a} / \mathbf{K} \mathbf{v}$$

$$\mathbf{v} = \mathbf{b} / \mathbf{K}^T \mathbf{u}$$

end for

Sinkhorn algorithm

The Sinkhorn algorithm:

Input:

Histograms $\mathbf{a}, \mathbf{b} \in \Sigma_N$,

kernel $\mathbf{K} = \exp(-\mathbf{C}/\varepsilon) \in \mathbb{R}^{N \times N}$,

number of iterations L

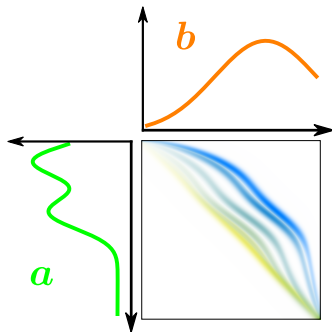
$\mathbf{v} = \mathbf{1}_N$

for $l = 1 \dots L$ **do**

$\mathbf{u} = \mathbf{a}/\mathbf{K}\mathbf{v}$

$\mathbf{v} = \mathbf{b}/\mathbf{K}^T\mathbf{u}$

end for



Sinkhorn algorithm

The Sinkhorn algorithm:

Input:

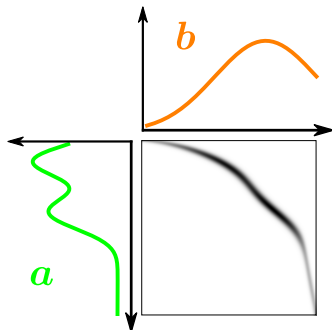
Histograms $\mathbf{a}, \mathbf{b} \in \Sigma_N$,
kernel $\mathbf{K} = \exp(-\mathbf{C}/\varepsilon) \in \mathbb{R}^{N \times N}$,
number of iterations L

```
 $\mathbf{v} = \mathbf{1}_N$   
for  $l = 1 \dots L$  do  
   $\mathbf{u} = \mathbf{a} / \mathbf{K} \mathbf{v}$   
   $\mathbf{v} = \mathbf{b} / \mathbf{K}^T \mathbf{u}$   
end for
```

Output:

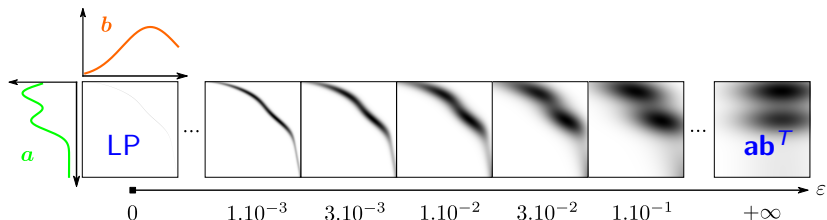
Distance: $L_{\mathbf{C}}^{\varepsilon}(\mathbf{a}, \mathbf{b}) = \varepsilon (\langle \mathbf{a}, \log \mathbf{u} \rangle + \langle \mathbf{b}, \log \mathbf{v} \rangle)$

Fast algorithm ($O(N^2)$ empirically, smooth distance !)



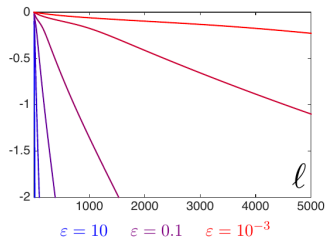
Sinkhorn algorithm

Influence of the regularization ε :



As $\varepsilon \rightarrow 0$:

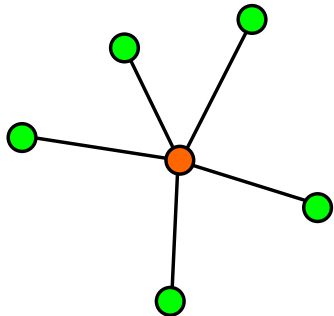
- ▶ slower convergence
- ▶ numerical instabilities



Sinkhorn barycenter

Euclidean barycenter:

$$\mathbf{p}(A, \lambda) \stackrel{\text{def.}}{=} \operatorname{argmin}_{\mathbf{b} \in \Sigma_N} \sum_{s=1}^S \lambda_s \|\mathbf{a}_s - \mathbf{b}\|^2$$



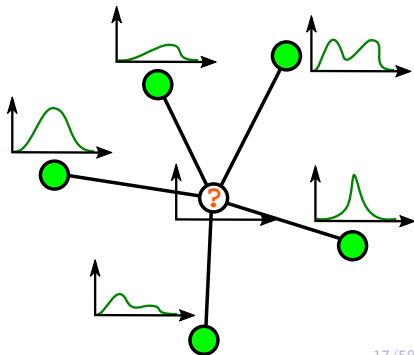
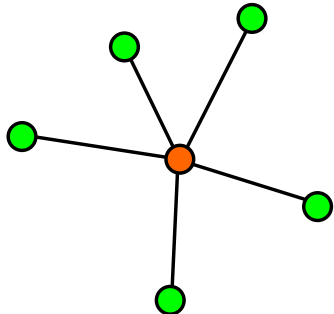
Sinkhorn barycenter

Euclidean barycenter:

$$\mathbf{p}(A, \lambda) \stackrel{\text{def.}}{=} \operatorname{argmin}_{\mathbf{b} \in \Sigma_N} \sum_{s=1}^S \lambda_s \|\mathbf{a}_s - \mathbf{b}\|^2$$

Wasserstein barycenter:

$$\mathbf{p}(A, \lambda) \stackrel{\text{def.}}{=} \operatorname{argmin}_{\mathbf{b} \in \Sigma_N} \sum_{s=1}^S \lambda_s L^2(\mathbf{a}_s, \mathbf{b})$$



Sinkhorn barycenter

Euclidean barycenter:

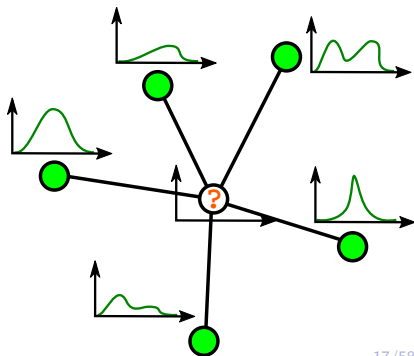
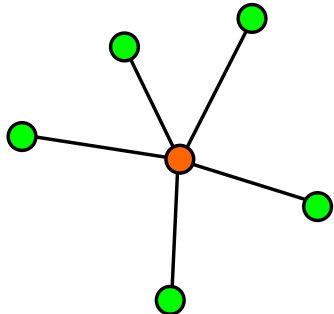
$$\mathbf{p}(A, \lambda) \stackrel{\text{def.}}{=} \operatorname{argmin}_{\mathbf{b} \in \Sigma_N} \sum_{s=1}^S \lambda_s \|\mathbf{a}_s - \mathbf{b}\|^2$$

Wasserstein barycenter:

$$\mathbf{p}(A, \lambda) \stackrel{\text{def.}}{=} \operatorname{argmin}_{\mathbf{b} \in \Sigma_N} \sum_{s=1}^S \lambda_s L^2(\mathbf{a}_s, \mathbf{b})$$

Regularized Wasserstein barycenter:

$$\mathbf{p}_\epsilon(A, \lambda) \stackrel{\text{def.}}{=} \operatorname{argmin}_{\mathbf{b} \in \Sigma_N} \sum_{s=1}^S \lambda_s L_\epsilon^2(\mathbf{a}_s, \mathbf{b})$$



Sinkhorn barycenter

Input:

Histograms $A = \mathbf{a}_1, \dots, \mathbf{a}_S \in \Sigma_N$,

weights $\lambda \in \Sigma_S$,

kernel $\mathbf{K} = \exp(-\mathbf{C}/\varepsilon) \in \mathbb{R}^{N \times N}$,

number of iterations L

$\mathbf{v} = \mathbf{1}_N$

for $l = 1 \dots L$ **do**

$\mathbf{u} = \mathbf{a} / \mathbf{K} \mathbf{v}$

$\mathbf{v} = \mathbf{b} / \mathbf{K}^T \mathbf{u}$

end for



Sinkhorn barycenter

Input:

Histograms $A = \mathbf{a}_1, \dots, \mathbf{a}_S \in \Sigma_N$,

weights $\lambda \in \Sigma_S$,

kernel $\mathbf{K} = \exp(-\mathbf{C}/\varepsilon) \in \mathbb{R}^{N \times N}$,

number of iterations L

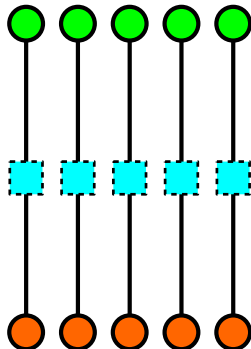
$$\mathbf{v}_s = \mathbf{1}_N \quad \forall s$$

for $l = 1 \dots L$ do

$$\mathbf{u}_s = \mathbf{a}_s / \mathbf{K} \mathbf{v}_s \quad \forall s$$

$$\mathbf{v}_s = \mathbf{b}_s / \mathbf{K}^T \mathbf{u}_s \quad \forall s$$

end for



Sinkhorn barycenter

Input:

Histograms $A = \mathbf{a}_1, \dots, \mathbf{a}_S \in \Sigma_N$,

weights $\lambda \in \Sigma_S$,

kernel $\mathbf{K} = \exp(-\mathbf{C}/\varepsilon) \in \mathbb{R}^{N \times N}$,

number of iterations L

$$\mathbf{v}_s = \mathbf{1}_N \quad \forall s$$

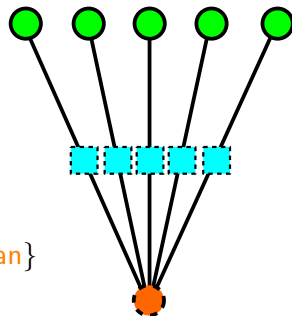
for $l = 1 \dots L$ **do**

$$\mathbf{u}_s = \mathbf{a}_s / \mathbf{K} \mathbf{v}_s \quad \forall s$$

$$\mathbf{b} = \prod_s (\mathbf{K}^T \mathbf{u}_s)^{\lambda_s} \quad \{\text{geometric mean}\}$$

$$\mathbf{v}_s = \mathbf{b} / \mathbf{K}^T \mathbf{u}_s \quad \forall s$$

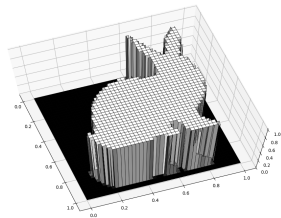
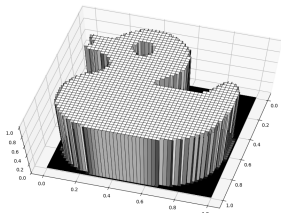
end for



Result: Fast algorithm, smooth quantities !

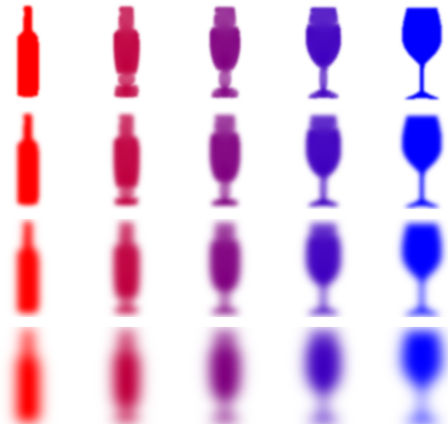
Sinkhorn barycenter

Shape interpolation:



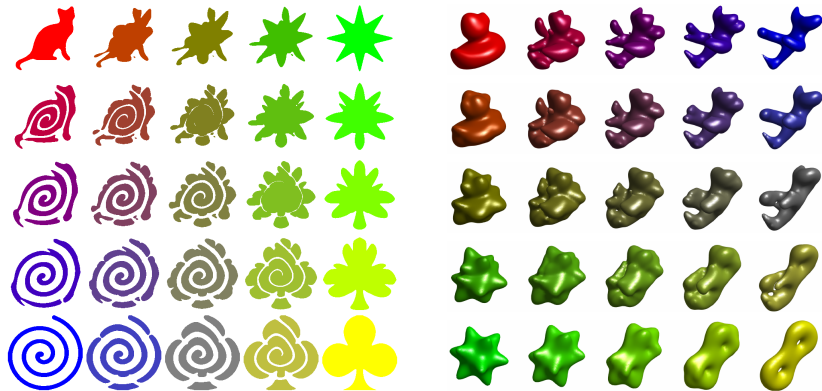
Sinkhorn barycenter

Regularization on barycenters:



Sinkhorn barycenter

Eye candy:



Sinkhorn barycenter

The Wasserstein space can be surprising sometimes...

M2DISCO	M2DISCO	G2DISCO	GEOMOD	GEOMOD
M2DISCO	M2DISCO	G2DISCO	GEOMOD	GEOMOD
M2DISCO	M2DISCO		GEOMOD	GEOMOD
SAARA	SAARA	SAARA	R3AM	R3AM
SAARA	SAARA	SAARA	R3AM	R3AM

Sinkhorn barycenter

The Wasserstein space can be surprising sometimes...

M2DISCO	M2DISCO	GEOMOD	GEOMOD	GEOMOD
M2DISCO	M2DISCO	GEOMOD	GEOMOD	GEOMOD
M2DISCO	M2DISCO	ORIGAMI	GEOMOD	GEOMOD
SAARA	SAARA	SAARA	R3AM	R3AM
SAARA	SAARA	SAARA	R3AM	R3AM

Sinkhorn barycenter

The Wasserstein space can be surprising sometimes...

M2DISCO	M2DISCO	GEOMOD	GEOMOD	GEOMOD
M2DISCO	M2DISCO	GEOMOD	GEOMOD	GEOMOD
M2DISCO	M2DISCO	GEOMOD	GEOMOD	GEOMOD
SAARA	SAARA	R3AM	R3AM	R3AM
SAARA	SAARA	R3AM	R3AM	R3AM

Outline

Introduction

Continuous and discrete

- Sinkhorn algorithm

- Sinkhorn barycenter

Semi-discrete setting

- Geometric formulation

- Numerical methods

- Applications

Sliced Optimal Transport

- Introduction

- Algorithms

- Applications

Semi-discrete problem

We specialize the initial formulation with:

- ▶ X a continuous Euclidian space
- ▶ $Y = \{y_1, \dots, y_n\}$ a finite set of discrete sites

Semi-discrete problem

We specialize the initial formulation with:

- ▶ X a continuous Euclidian space
- ▶ $Y = \{y_1, \dots, y_n\}$ a finite set of discrete sites

$$\min_{T, \nu = T_{\#} \mu} \int_{x \in X} c(x, T(x)) d\mu(x)$$

Semi-discrete problem

We specialize the initial formulation with:

- ▶ X a continuous Euclidian space
- ▶ $Y = \{y_1, \dots, y_n\}$ a finite set of discrete sites

$$\min_{T, \nu = T_{\#} \mu} \int_{x \in \bigcup_{i=1}^N T^{-1}(y_i)} c(x, T(x)) d\mu(x)$$

Semi-discrete problem

We specialize the initial formulation with:

- ▶ X a continuous Euclidian space
- ▶ $Y = \{y_1, \dots, y_n\}$ a finite set of discrete sites

$$\min_{T, \nu = T_{\#} \mu} \sum_{i=1}^N \int_{x \in T^{-1}(y_i)} c(x, y_i) d\mu(x)$$

Least-squares assignment

Definition (Least-squares assignment)

Given a domain X and a set of sites Y , a least-squares assignment of X to Y is a function $T : X \rightarrow Y$ that minimizes

$$\int_{x \in X} \|x - T(x)\|^2 dx$$

It is said to be *constrained* if we require that each $T^{-1}(y_i)$ has a given measure, then called its capacity.

Least-squares assignment

Definition (Least-squares assignment)

Given a domain X and a set of sites Y , a least-squares assignment of X to Y is a function $T : X \rightarrow Y$ that minimizes

$$\int_{x \in X} \|x - T(x)\|^2 dx$$

It is said to be *constrained* if we require that each $T^{-1}(y_i)$ has a given measure, then called its capacity.

Definition (Laguerre diagram)

A Laguerre diagram is a generalization of a Voronoi diagram, where each site is assigned a weight.

$$\text{Lag}^\phi(y_i) = \{x \in X \mid c(x, y_i) - \phi_i \leq c(x, y_j) - \phi_j, \forall j \neq i\}$$

Least-square assignment computation

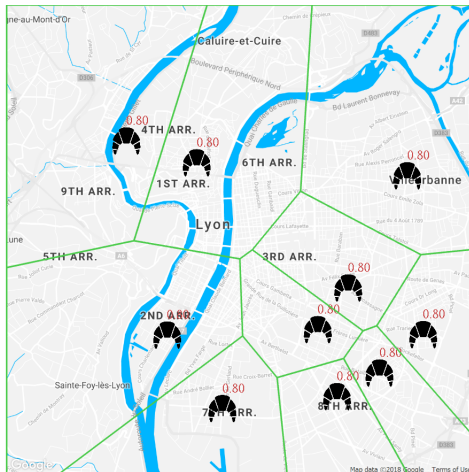


Figure: Laguerre diagram

Least-square assignment computation

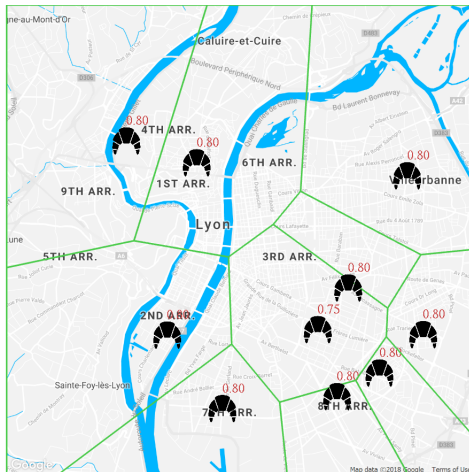


Figure: Laguerre diagram

Equivalence theorems

Theorem (Aurenhammer et al., 1998)

Any assignment from X to Y induced by a Laguerre diagram is a constrained least-squares assignment for the corresponding prescribed capacities.

Equivalence theorems

Theorem (Aurenhammer et al., 1998)

Any assignment from X to Y induced by a Laguerre diagram is a constrained least-squares assignment for the corresponding prescribed capacities.

Theorem (Aurenhammer et al., 1998)

Conversely, any constrained least-squares assignment from X to Y for given capacities can be realized by a certain Laguerre diagram.

Equivalence theorems

Theorem (Aurenhammer et al., 1998)

Any assignment from X to Y induced by a Laguerre diagram is a constrained least-squares assignment for the corresponding prescribed capacities.

Theorem (Aurenhammer et al., 1998)

Conversely, any constrained least-squares assignment from X to Y for given capacities can be realized by a certain Laguerre diagram.

We can build a semi-discrete optimal transport plan just by tracing a power diagram.

Unconstrained concave optimization

- Recall the dual Kantorovich formulation:

$$\max_{\substack{\phi, \psi \\ \phi + \psi \leq c}} \int_X \phi(x) d\mu(x) + \sum_{i=1}^N \psi_i \nu_i.$$

Unconstrained concave optimization

- Recall the dual Kantorovich formulation:

$$\max_{\substack{\phi, \psi \\ \phi + \psi \leq c}} \int_X \phi(x) d\mu(x) + \sum_{i=1}^N \psi_i \nu_i.$$

- Introducing the c -transform

$$\psi^c(x) = \inf_{1 \leq i \leq N} [c(x, y_i) - \psi_i] = \text{distance to the closest } y_i,$$

we get an unconstrained optimization problem:

$$\max_{\psi \in \mathbb{R}^N} \int_X \psi^c(x) d\mu(x) + \sum_{i=1}^N \psi_i \nu_i.$$

Unconstrained concave optimization

- ▶ Recall the dual Kantorovich formulation:

$$\max_{\substack{\phi, \psi \\ \phi + \psi \leq c}} \int_X \phi(x) d\mu(x) + \sum_{i=1}^N \psi_i \nu_i.$$

- ▶ Introducing the c -transform

$$\psi^c(x) = \inf_{1 \leq i \leq N} [c(x, y_i) - \psi_i] = \text{distance to the closest } y_i,$$

we get an unconstrained optimization problem:

$$\max_{\psi \in \mathbb{R}^N} \int_X \psi^c(x) d\mu(x) + \sum_{i=1}^N \psi_i \nu_i.$$

- ▶ Because $X = \bigsqcup_{i=1}^N \text{Lag}^\psi(y_i)$ and under some assumptions:

$$\max_{\psi \in \mathbb{R}^N} \sum_{i=1}^N \int_{\text{Lag}^\psi(y_i)} (c(x, y_i) + \psi_i) d\mu(x) + \sum_{i=1}^N \psi_i \nu_i.$$

Unconstrained concave optimization (cont.)

- ▶ Continuing the previous calculations, we get:

$$\max_{\psi \in \mathbb{R}^N} \Phi(\psi) := \sum_{i=1}^N \int_{\text{Lag}^\psi(y_i)} c(x, y_i) d\mu(x) + \sum_{i=1}^N \psi_i (G_i(\psi) - \nu_i)$$

where $G_i(\psi) = \int_{\text{Lag}^\psi(y_i)} d\mu(x)$ = weighted area of $\text{Lag}^\psi(y_i)$.

Unconstrained concave optimization (cont.)

- ▶ Continuing the previous calculations, we get:

$$\max_{\psi \in \mathbb{R}^N} \Phi(\psi) := \sum_{i=1}^N \int_{\text{Lag}^\psi(y_i)} c(x, y_i) d\mu(x) + \sum_{i=1}^N \psi_i (G_i(\psi) - \nu_i)$$

where $G_i(\psi) = \int_{\text{Lag}^\psi(y_i)} d\mu(x)$ = weighted area of $\text{Lag}^\psi(y_i)$.

- ▶ One can show that Φ is concave and its gradient is given by:

$$\nabla \Phi(\psi) = G(\psi) - \nu.$$

Unconstrained concave optimization (cont.)

- ▶ Continuing the previous calculations, we get:

$$\max_{\psi \in \mathbb{R}^N} \Phi(\psi) := \sum_{i=1}^N \int_{\text{Lag}^\psi(y_i)} c(x, y_i) d\mu(x) + \sum_{i=1}^N \psi_i (G_i(\psi) - \nu_i)$$

where $G_i(\psi) = \int_{\text{Lag}^\psi(y_i)} d\mu(x)$ = weighted area of $\text{Lag}^\psi(y_i)$.

- ▶ One can show that Φ is concave and its gradient is given by:

$$\nabla \Phi(\psi) = G(\psi) - \nu.$$

Theorem

Semi-discrete optimal transport $\iff$ finding $\psi \in \mathbb{R}^N$ such that:

$$\forall i, \quad G_i(\psi) = \nu_i.$$

The optimal transport map T_ψ is given by the assignment

$$T_\psi : x \in X \mapsto y_i \text{ s.t. } x \in \text{Lag}^\psi(y_i).$$

Illustrations of semi-discrete optimal transport

Setting: quadratic cost $c(x, y) = \|x - y\|^2$

- ▶ Source = uniform density on a square,
- ▶ Target = uniform, points sampled in a **small region**,

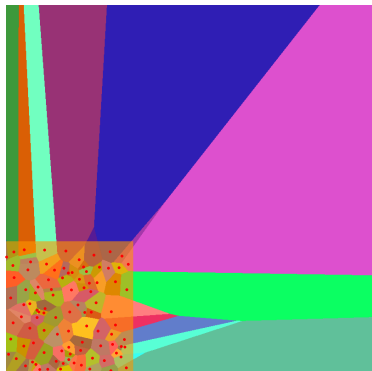


Figure: Left: initial Laguerre diagram. Right: final Laguerre diagram.

Illustrations of semi-discrete optimal transport

Setting: quadratic cost $c(x, y) = \|x - y\|^2$

- ▶ Source = uniform density on a square,
- ▶ Target = uniform, points sampled in a **small region**,

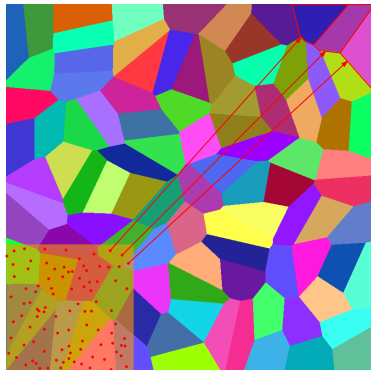


Figure: Left: initial Laguerre diagram. Right: final Laguerre diagram.

Illustrations of semi-discrete optimal transport

Setting: quadratic cost $c(x, y) = \|x - y\|^2$

- ▶ Source = uniform density on a square,
- ▶ Target = uniform, points sampled in a **small region**,
- ▶ Points do **NOT** move.

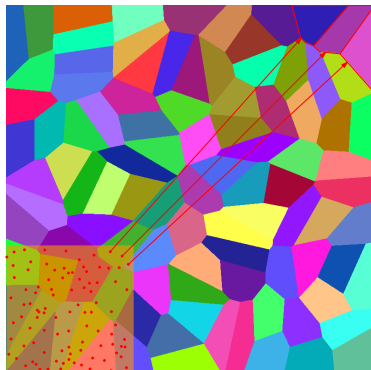
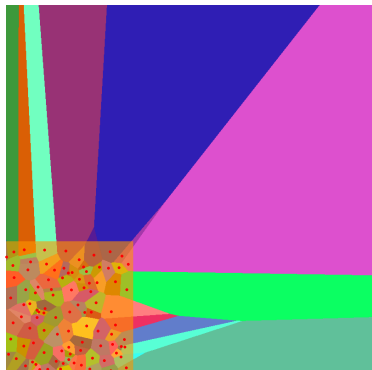


Figure: Left: initial Laguerre diagram. Right: final Laguerre diagram.

Numerical methods

Different methods:

- ▶ Olliger-Prussner type algorithms [CKO99]
 - ▶ coordinatewise decrement;
 - ▶ complexity = $O(N^3/\eta)$ where η = numerical precision.

Numerical methods

Different methods:

- ▶ Olier-Prussner type algorithms [CKO99]
 - ▶ coordinatewise decrement;
 - ▶ complexity = $O(N^3/\eta)$ where η = numerical precision.
- ▶ Unconstrained concave optimization
 - ▶ one can use any classical optimization method;
 - ▶ Newton or quasi-Newton methods (order 2) [Mér11, Mir15];
 - ▶ better practical complexity and convergence guarantees.

Numerical methods

Different methods:

- ▶ Ollker-Prussner type algorithms [CKO99]
 - ▶ coordinatewise decrement;
 - ▶ complexity = $O(N^3/\eta)$ where η = numerical precision.
- ▶ Unconstrained concave optimization
 - ▶ one can use any classical optimization method;
 - ▶ Newton or quasi-Newton methods (order 2) [Mér11, Mir15];
 - ▶ better practical complexity and convergence guarantees.

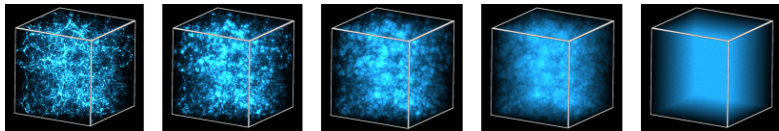


Figure: Early universe reconstruction using semi-discrete optimal transport (Brenier et al, pictures by B. Lévy).

Damped Newton's method

Input A density μ , a discrete measure $\nu = \sum_{i=1}^N \nu_i \delta_{y_i}$,
A numerical error $\eta > 0$,
Weights $\psi^0 \in \mathbb{R}^N$ such that
 $\varepsilon_0 := \min [\min_i G_i(\psi^0), \min_i \nu_i] > 0$.

Damped Newton's method

Input A density μ , a discrete measure $\nu = \sum_{i=1}^N \nu_i \delta_{y_i}$,
A numerical error $\eta > 0$,
Weights $\psi^0 \in \mathbb{R}^N$ such that
 $\varepsilon_0 := \min [\min_i G_i(\psi^0), \min_i \nu_i] > 0$.

While $\|G(\psi^k) - \nu\|_2 \geq \eta$

- ▶ Compute $\mathbf{v}^k = -DG(\psi^k)^+(G(\psi^k) - \nu)$,
- ▶ Find the smallest $\ell \in \mathbb{N}$ such that
 $\psi^{k,\ell} := \psi^k + 2^{-\ell} \mathbf{v}^k$ and

$$\begin{cases} \min_{1 \leq i \leq N} G_i(\psi^{k,\ell}) \geq \varepsilon_0 \\ \left\| G(\psi^{k,\ell}) - \nu \right\|_2 \leq (1 - 2^{-(\ell+1)}) \left\| G(\psi^k) - \nu \right\|_2. \end{cases}$$

- ▶ Set $\psi^{k+1} = \psi^k + 2^{-\ell} \mathbf{v}^k$ et $k \leftarrow k + 1$.

Damped Newton's method

Input A density μ , a discrete measure $\nu = \sum_{i=1}^N \nu_i \delta_{y_i}$,
A numerical error $\eta > 0$,
Weights $\psi^0 \in \mathbb{R}^N$ such that
 $\varepsilon_0 := \min [\min_i G_i(\psi^0), \min_i \nu_i] > 0$.

While $\|G(\psi^k) - \nu\|_2 \geq \eta$

- ▶ Compute $v^k = -DG(\psi^k)^+(G(\psi^k) - \nu)$,
- ▶ Find the smallest $\ell \in \mathbb{N}$ such that
 $\psi^{k,\ell} := \psi^k + 2^{-\ell} v^k$ and

$$\begin{cases} \min_{1 \leq i \leq N} G_i(\psi^{k,\ell}) \geq \varepsilon_0 \\ \|G(\psi^{k,\ell}) - \nu\|_2 \leq (1 - 2^{-(\ell+1)}) \|G(\psi^k) - \nu\|_2. \end{cases}$$

- ▶ Set $\psi^{k+1} = \psi^k + 2^{-\ell} v^k$ et $k \leftarrow k + 1$.

Damped Newton's method

Input A density μ , a discrete measure $\nu = \sum_{i=1}^N \nu_i \delta_{y_i}$,

A numerical error $\eta > 0$,

Weights $\psi^0 \in \mathbb{R}^N$ such that

$\varepsilon_0 := \min [\min_i G_i(\psi^0), \min_i \nu_i] > 0$.

While $\|G(\psi^k) - \nu\|_2 \geq \eta$

► Compute $\mathbf{v}^k = -DG(\psi^k)^+(G(\psi^k) - \nu)$,

► Find the smallest $\ell \in \mathbb{N}$ such that

$\psi^{k,\ell} := \psi^k + 2^{-\ell} \mathbf{v}^k$ and

$$\begin{cases} \min_{1 \leq i \leq N} G_i(\psi^{k,\ell}) \geq \varepsilon_0 \\ \left\| G(\psi^{k,\ell}) - \nu \right\|_2 \leq (1 - 2^{-(\ell+1)}) \left\| G(\psi^k) - \nu \right\|_2. \end{cases}$$

► Set $\psi^{k+1} = \psi^k + 2^{-\ell} \mathbf{v}^k$ et $k \leftarrow k + 1$.

Damped Newton's method

Input A density μ , a discrete measure $\nu = \sum_{i=1}^N \nu_i \delta_{y_i}$,

A numerical error $\eta > 0$,

Weights $\psi^0 \in \mathbb{R}^N$ such that

$\varepsilon_0 := \min [\min_i G_i(\psi^0), \min_i \nu_i] > 0$.

While $\|G(\psi^k) - \nu\|_2 \geq \eta$

► Compute $v^k = -DG(\psi^k)^+(G(\psi^k) - \nu)$,

► Find the smallest $\ell \in \mathbb{N}$ such that

$\psi^{k,\ell} := \psi^k + 2^{-\ell} v^k$ and

$$\begin{cases} \min_{1 \leq i \leq N} G_i(\psi^{k,\ell}) \geq \varepsilon_0 \\ \|G(\psi^{k,\ell}) - \nu\|_2 \leq (1 - 2^{-(\ell+1)}) \|G(\psi^k) - \nu\|_2. \end{cases}$$

► Set $\psi^{k+1} = \psi^k + 2^{-\ell} v^k$ et $k \leftarrow k + 1$.

Output ψ^k such that $\|G(\psi^k) - \nu\|_2 \leq \eta$.

Guarantees of Newton's method

Theorem (Convergence)

We set $E_\varepsilon = \{\psi \in \mathbb{R}^N \mid \forall i, G_i(\psi) > \varepsilon\}$. The algorithm converges globally with a linear rate and is locally quadratic if:

Guarantees of Newton's method

Theorem (Convergence)

We set $E_\varepsilon = \{\psi \in \mathbb{R}^N \mid \forall i, G_i(\psi) > \varepsilon\}$. The algorithm converges globally with a linear rate and is locally quadratic if:

- ▶ *(Regularity) G is $\mathcal{C}^1$ on E_ε ,*

Guarantees of Newton's method

Theorem (Convergence)

We set $E_\varepsilon = \{\psi \in \mathbb{R}^N \mid \forall i, G_i(\psi) > \varepsilon\}$. The algorithm converges globally with a linear rate and is locally quadratic if:

- ▶ *(Regularity) G is $\mathcal{C}^1$ on E_ε ,*
- ▶ *(Strict monotonicity) $\forall \psi \in E_\varepsilon$, DG is negative definite on $\{\text{cst}\}^\perp$.*

Guarantees of Newton's method

Theorem (Convergence)

We set $E_\varepsilon = \{\psi \in \mathbb{R}^N \mid \forall i, G_i(\psi) > \varepsilon\}$. The algorithm converges globally with a linear rate and is locally quadratic if:

- ▶ *(Regularity) G is $\mathcal{C}^1$ on E_ε ,*
- ▶ *(Strict monotonicity) $\forall \psi \in E_\varepsilon$, DG is negative definite on $\{\text{cst}\}^\perp$.*

True for :

Guarantees of Newton's method

Theorem (Convergence)

We set $E_\varepsilon = \{\psi \in \mathbb{R}^N \mid \forall i, G_i(\psi) > \varepsilon\}$. The algorithm converges globally with a linear rate and is locally quadratic if:

- ▶ *(Regularity) G is $\mathcal{C}^1$ on E_ε ,*
- ▶ *(Strict monotonicity) $\forall \psi \in E_\varepsilon$, DG is negative definite on $\{\text{cst}\}^\perp$.*

True for :

- ▶ [Mir15] quadratic cost for a density in $\mathbb{R}^2$ and $\mathbb{R}^3$,

Guarantees of Newton's method

Theorem (Convergence)

We set $E_\varepsilon = \{\psi \in \mathbb{R}^N \mid \forall i, G_i(\psi) > \varepsilon\}$. The algorithm converges globally with a linear rate and is locally quadratic if:

- ▶ (Regularity) G is $\mathcal{C}^1$ on E_ε ,
- ▶ (Strict monotonicity) $\forall \psi \in E_\varepsilon$, DG is negative definite on $\{\text{cst}\}^\perp$.

True for :

- ▶ [Mir15] quadratic cost for a density in $\mathbb{R}^2$ and $\mathbb{R}^3$,
- ▶ [KMT19] cost satisfying the so-called *Ma-Trudinger-Wang* condition (verified for $c(x, y) = \|x - y\|^2$) and measures supported on subsets of $\mathbb{R}^d$,

Guarantees of Newton's method

Theorem (Convergence)

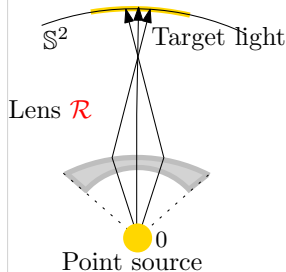
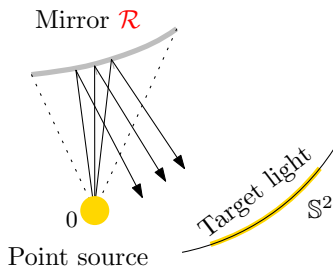
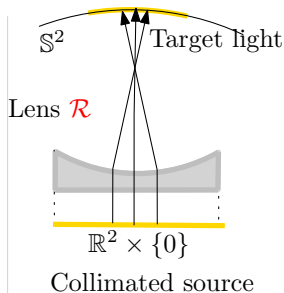
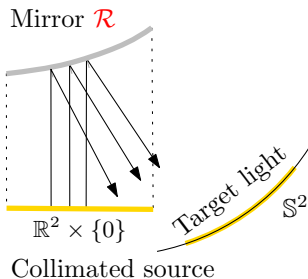
We set $E_\varepsilon = \{\psi \in \mathbb{R}^N \mid \forall i, G_i(\psi) > \varepsilon\}$. The algorithm converges globally with a linear rate and is locally quadratic if:

- ▶ (Regularity) G is $\mathcal{C}^1$ on E_ε ,
- ▶ (Strict monotonicity) $\forall \psi \in E_\varepsilon$, DG is negative definite on $\{\text{cst}\}^\perp$.

True for :

- ▶ [Mir15] quadratic cost for a density in $\mathbb{R}^2$ and $\mathbb{R}^3$,
- ▶ [KMT19] cost satisfying the so-called *Ma-Trudinger-Wang* condition (verified for $c(x, y) = \|x - y\|^2$) and measures supported on subsets of $\mathbb{R}^d$,
- ▶ [MMT18] quadratic cost and measure supported on a *simplex soup* (think triangulations in $\mathbb{R}^3$).

Application: inverse problem in optics



Optimal transport and optics

Ideas:

- ▶ discretize target light $\nu = \sum_{i=1}^N \nu_i \delta_{y_i}$,

Optimal transport and optics

Ideas:

- ▶ discretize target light $\nu = \sum_{i=1}^N \nu_i \delta_{y_i}$,
- ▶ parametrize the mirror or lens by $x \mapsto \mathcal{R}_\psi(x)$ where $\psi \in \mathbb{R}^N$,

Optimal transport and optics

Ideas:

- ▶ discretize target light $\nu = \sum_{i=1}^N \nu_i \delta_{y_i}$,
- ▶ parametrize the mirror or lens by $x \mapsto \mathcal{R}_\psi(x)$ where $\psi \in \mathbb{R}^N$,
- ▶ introduce the *visibility cell* for direction y_i :

$$V^\psi(y_i) = \{x \in X \mid x \text{ reflected/refracted towards } y_i\},$$

Optimal transport and optics

Ideas:

- ▶ discretize target light $\nu = \sum_{i=1}^N \nu_i \delta_{y_i}$,
- ▶ parametrize the mirror or lens by $x \mapsto \mathcal{R}_\psi(x)$ where $\psi \in \mathbb{R}^N$,
- ▶ introduce the *visibility cell* for direction y_i :

$$V^\psi(y_i) = \{x \in X \mid x \text{ reflected/refracted towards } y_i\},$$

- ▶ we can show that $V^\psi(y_i) = \text{Lag}^\psi(y_i)$ for a certain cost c ,

Optimal transport and optics

Ideas:

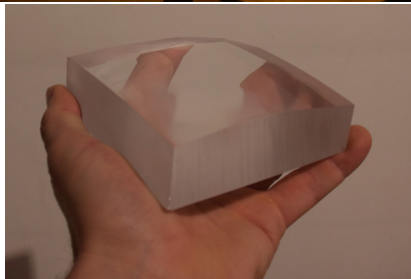
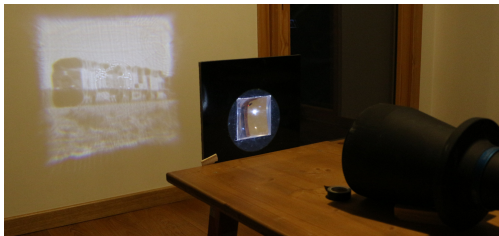
- ▶ discretize target light $\nu = \sum_{i=1}^N \nu_i \delta_{y_i}$,
- ▶ parametrize the mirror or lens by $x \mapsto \mathcal{R}_\psi(x)$ where $\psi \in \mathbb{R}^N$,
- ▶ introduce the *visibility cell* for direction y_i :

$$V^\psi(y_i) = \{x \in X \mid x \text{ reflected/refracted towards } y_i\},$$

- ▶ we can show that $V^\psi(y_i) = \text{Lag}^\psi(y_i)$ for a certain cost c ,
- ▶ finally $\text{Lag}^\psi(y_i) = X \cap \text{Pow}^\omega(p_i) \implies$ efficient computation of $\text{Lag}^\psi(y_i)$.

Inverse problems in optics: physical prototypes

Example of a lens for a collimated light source



Inverse problems in optics: physical prototypes

Example of a mirror for a collimated light source



Inverse problems in optics: simulation

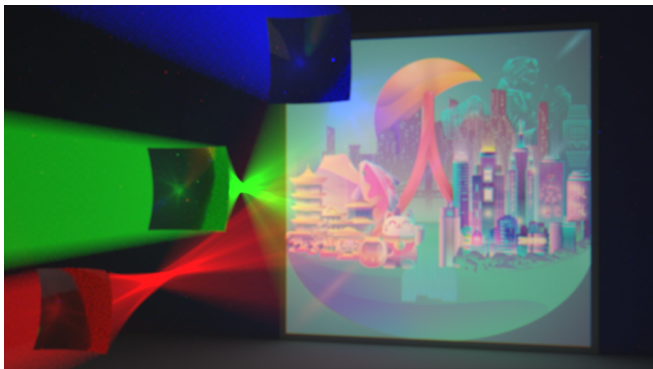


Figure: Three lenses where each one of them targets a color channel of an image.

Application: symmetrized semi-discrete optimal transport

$(x_i) := \text{rand}()$

$(y_i) := \text{rand}()$

$\phi_i := 1/N$

$\psi_i := 1/N$

while ad lib **do**

$(\phi_i) :=$ optimal transport weights from (Ω, μ) to (x_i)

$(y_i) :=$ centroids of $\text{Lag}^\phi(x_i)$

$(\psi_i) :=$ optimal transport weights from (Ω, ν) to (y_i)

$(x_i) :=$ centroids of $\text{Lag}^\psi(y_i)$

end while

Algorithm 1: Alternating algorithm for computing symmetrized optimal transport plans

Direct algorithm

$$\begin{aligned}\Phi((x_i), (y_i), (\phi_i), (\psi_i)) = & \sum_{i=1}^n \phi_i \left(\int_{x \in \text{Lag}^\phi(x_i)} \sigma(x) dx - p_i \right) + \sum_{i=1}^n \psi_i \left(\int_{y \in \text{Lag}^\psi(y_i)} \rho(y) dy - q_i \right) \\ & - \sum_{i=1}^n \int_{x \in \text{Lag}^\phi(x_i)} \|x - x_i\|^2 \sigma(x) dx - \sum_{i=1}^n \int_{y \in \text{Lag}^\psi(y_i)} \|y - y_i\|^2 \rho(y) dy \\ & + \sum_{i=1}^n A \left\| x_i \cdot \int_{y \in \text{Lag}^\psi(y_i)} \sigma(y) dy - \int_{y \in \text{Lag}^\psi(y_i)} y \sigma(y) dy \right\|^2 \\ & + \sum_{i=1}^n A \left\| y_i \cdot \int_{x \in \text{Lag}^\phi(x_i)} \rho(x) dx - \int_{x \in \text{Lag}^\phi(x_i)} x \rho(x) dx \right\|^2\end{aligned}$$

Direct algorithm

$$\begin{aligned}\Phi((x_i), (y_i), (\phi_i), (\psi_i)) = & \sum_{i=1}^n \phi_i \left(\int_{x \in \text{Lag}^\phi(x_i)} \sigma(x) dx - p_i \right) + \sum_{i=1}^n \psi_i \left(\int_{y \in \text{Lag}^\psi(y_i)} \rho(y) dy - q_i \right) \\ & - \sum_{i=1}^n \int_{x \in \text{Lag}^\phi(x_i)} \|x - x_i\|^2 \sigma(x) dx - \sum_{i=1}^n \int_{y \in \text{Lag}^\psi(y_i)} \|y - y_i\|^2 \rho(y) dy \\ & + \sum_{i=1}^n A \left\| x_i \cdot \int_{y \in \text{Lag}^\psi(y_i)} \sigma(y) dy - \int_{y \in \text{Lag}^\psi(y_i)} y \sigma(y) dy \right\|^2 \\ & + \sum_{i=1}^n A \left\| y_i \cdot \int_{x \in \text{Lag}^\phi(x_i)} \rho(x) dx - \int_{x \in \text{Lag}^\phi(x_i)} x \rho(x) dx \right\|^2\end{aligned}$$

Breaking down the objective function

$$\begin{aligned} & \sum_{i=1}^n \phi_i \left(\int_{x \in \text{Lag}^\phi(x_i)} \sigma(x) dx - p_i \right) \\ & - \sum_{i=1}^n \int_{x \in \text{Lag}^\phi(x_i)} \|x - x_i\|^2 \sigma(x) dx \end{aligned}$$

to compare with

$$\sum_{y \in Y} \phi_y \left(\int_{x \in T_\phi^{-1}} d\mu(x) - \nu(y) \right) - \int_{x \in T_\phi^{-1}} c(x, y) d\mu(x)$$

Breaking down the objective function

Centroid of $Lag^\phi(x_i)$:

$$\frac{\int_{x \in Lag^\phi(x_i)} x \rho(x) dx}{\int_{x \in Lag^\phi(x_i)} \rho(x) dx}$$

Thus

$$\sum_{i=1}^n A \left\| y_i \cdot \int_{x \in Lag^\phi(x_i)} \rho(x) dx - \int_{x \in Lag^\phi(x_i)} x \rho(x) dx \right\|^2 = 0$$
$$\Rightarrow \forall i, y_i \text{ is at the barycenter of } Lag^\phi(x_i)$$

Outline

Introduction

Continuous and discrete

- Sinkhorn algorithm

- Sinkhorn barycenter

Semi-discrete setting

- Geometric formulation

- Numerical methods

- Applications

Sliced Optimal Transport

- Introduction

- Algorithms

- Applications

Sliced Optimal Transport - Introduction

Motivation

- ▶ OT is a useful tool
- ▶ Computationally costly
- ▶ No algorithm in high dimensions
- ▶ Need for an approximation

Sliced Wasserstein Distance

Definition [RPDB11]

$$\tilde{W}_p(X, Y) = \left(\int_{\theta \in \Omega} W_p(X_\theta, Y_\theta)^p d\theta \right)^{\frac{1}{p}}$$

where

$$X_\theta = \{\langle X_i, \theta \rangle\},$$

$$\Omega = \{\theta \in \mathbb{R}^d \mid \|\theta\| = 1\},$$

$W_p(X_\theta, Y_\theta)$ the 1D Wasserstein-p distance between X_θ and Y_θ

A Link to OT

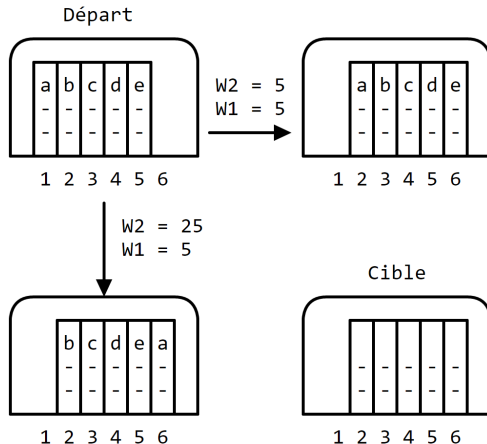
Which Properties for Sliced Wasserstein Distance ? [Bon13]

- ▶ Sliced Wasserstein Distance is a Distance
 - ▶ Triangular inequality
 - ▶ $\tilde{W}(\mu, \nu) = 0 \Leftrightarrow \mu = \nu$
- ▶ "Equivalent" to standard Wasserstein Distance for distribution with finite support

$$c_{d,p} \tilde{W}_p(\mu, \nu)^p \leq W_p(\mu, \nu)^p \leq C_{d,p} \tilde{W}_p(\mu, \nu)^{1/(d+1)}$$

1D Optimal Transport

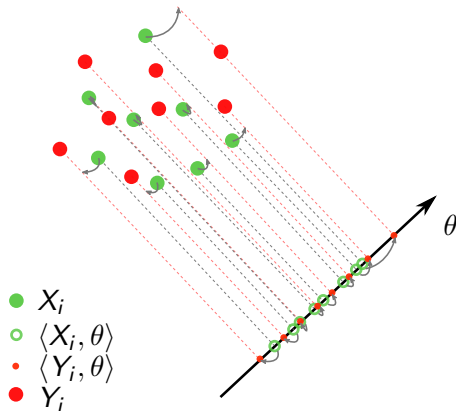
- Brenier's Theorem no flux crossing for $p > 1$ [Bre91]



1D Optimal Transport

Discrete Sliced Algorithm

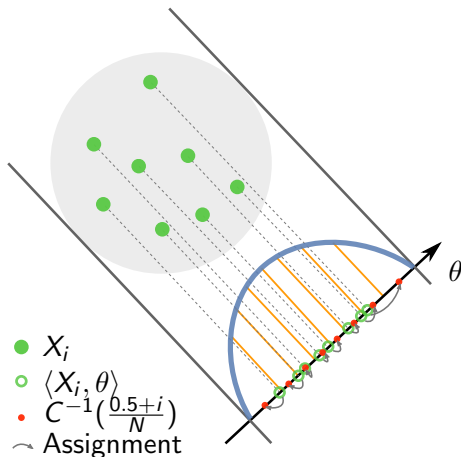
- ▶ Sort both point sets along θ
- ▶ Assign points greedily
- ▶ Compute Transport Cost



1D Optimal Transport

Semi-Discrete Sliced Algorithm

- ▶ Compute Distribution
Radon transform along θ
and corresponding CDF
and inverse CDF
- ▶ Sort points along θ
- ▶ Assign each point to a cell
of same



A Bit of Monte Carlo

$$\tilde{W}_p(X, Y) = \left(\int_{\theta \in \Omega} W_p(X_\theta, Y_\theta)^2 d\theta \right)^{\frac{1}{2}}$$

A Bit of Monte Carlo

$$\tilde{W}_p(X, Y) \approx \left(\frac{1}{K} \sum_{i=1}^K W_p(X_{\theta_i}, Y_{\theta_i})^2 \right)^{\frac{1}{2}}$$

with $\{\theta_1, \dots, \theta_K\}$ uniformly picked directions

- ▶ $W_p(X_{\theta_i}, Y_{\theta_i})$ computed in $\mathcal{O}(n \log n)$
- ▶ Approximation in $\mathcal{O}(Kn \log n)$

A Bit of Monte Carlo

Data: X and Y two distributions, $\{\theta_1, \dots, \theta_K\}$ K uniformly distributed directions

Result: $\tilde{W}$: The sliced Wasserstein- p Distance between X and Y
 $\tilde{W} = 0$;

for $i = 1..K$ **do**

$\tilde{W} += W_p(X_{\theta_i}, Y_{\theta_i})^p$;

end

$\tilde{W} /= K$;

$\tilde{W} = \tilde{W}^{\frac{1}{p}}$;

Algorithm 2: Sliced Wasserstein- p Distance Approximation Computation

A Bit of Monte Carlo

Data: X and Y two sets of N reals

Result: W : The 1D Wasserstein- p Distance between X and Y

$\text{SORT}(X);$

$\text{SORT}(Y);$

$W = 0;$

for $i = 1..N$ **do**

$W += |X_i - Y_i|^p;$

end

$W = W^{\frac{1}{p}};$

Algorithm 3: 1D Discrete Wasserstein- p Distance Computation

A Bit of Monte Carlo

Data: X a set of N reals, C the CDF of target distribution

Result: W : The 1D Wasserstein- p Distance between X and Y

$\text{SORT}(X)$;

$W = 0$;

for $i = 0..N-1$ **do**

$W += |X_i - C^{-1}(\frac{0.5+i}{N})|^p$;

end

$W = W^{\frac{1}{p}}$;

Algorithm 4: 1D Semi-Discrete Wasserstein- p Distance Computation

Sliced Optimal Transport Minimization (Semi-Discrete)

Problem Statement

- ▶ Given a distribution Y on $\mathbb{R}^d$ find a point set $X \subset \mathbb{R}^d$ of size N that minimizes the $\tilde{W}_p(X, Y)$

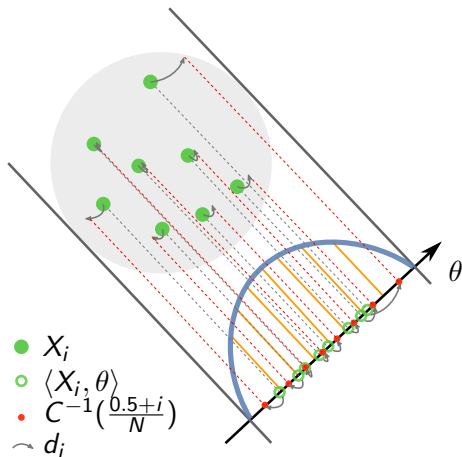
Iterative Method

- ▶ Inspired from Stochastic Gradient Descent
- ▶ Minimize each point contribution to the cost and iterate

Sliced Optimal Transport Minimization (Semi-Discrete)

Iterative Method

- ▶ Agregate d_i^θ for multiple θ to approximate the gradient for point X_i
- ▶ Displace X_i according to estimated gradient
- ▶ Iterate until convergence



Sliced Optimal Transport Minimization (Semi-Discrete)

Data: X : N random points, Y : a distribution, M : number of slices for gradient approximation

Result: X : N points minimizing $\tilde{W}_p(X, Y)$

while *Not Converged* **do**

for $j = 1..M$ **do**

θ = Random Direction;

d_i^θ displacement of X_i along θ to minimize $W_p(X_\theta, Y_\theta)$;

for $i = 1..N$ **do**

$d_i \ += \frac{d_i^\theta}{M} \theta$;

end

end

for $i = 1..N$ **do**

$X_i \ += \ d_i$;

$d_i = 0$;

end

end

Algorithm 5: Iterative Optimization Method

Texture Interpolation [RPDB11]

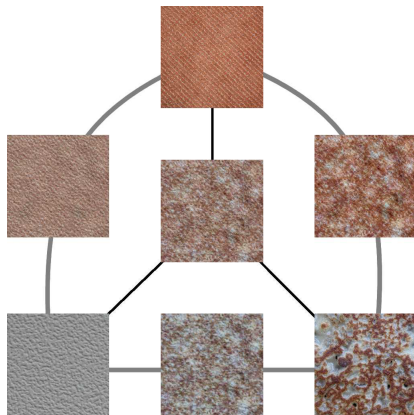


Figure: Texture interpolation using an iterative optimization method to find the Wasserstein Barycenter X minimizing $\sum_{Y_i \in J} p_i \tilde{W}_2(X, Y_i)$ with J a set of distributions

Point Cloud Registration [BC19]

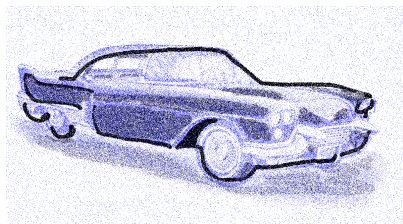
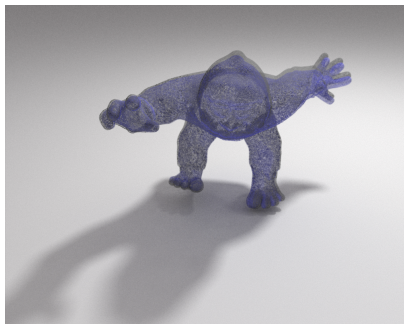


Figure: Pointset Registration using Sliced Partial Optimal Transport

Sampling

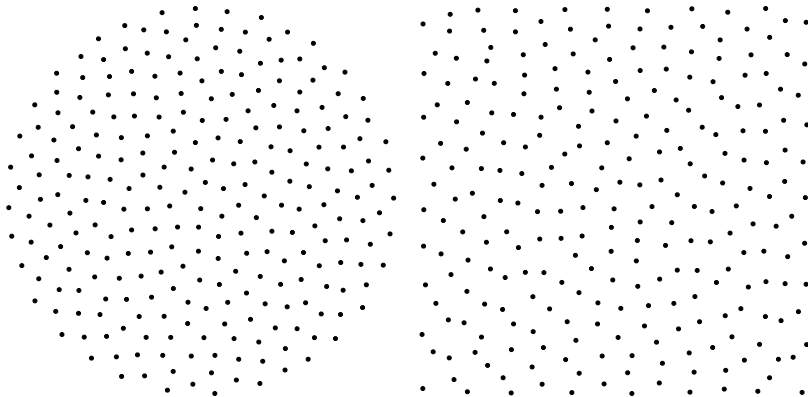


Figure: Uniform Sampling using sliced optimal transport minimization

High Dimensional Point Assignment Problem

Video

Conclusion

Usage of optimal transport:

- ▶ such matching
- ▶ wow, meaningful distances
- ▶ very loss function for optimization
- ▶ much interpolation (barycenters)

Other advantages:

- ▶ works in different settings;
- ▶ can be computed efficiently;



Conclusion

Usage of optimal transport:

- ▶ such matching
- ▶ wow, meaningful distances
- ▶ very loss function for optimization
- ▶ much interpolation (barycenters)

Other advantages:

- ▶ works in different settings;
- ▶ can be computed efficiently;



Thank you for your attention.



References I

-  Martin Arjovsky, Soumith Chintala, and Léon Bottou, *Wasserstein GAN*, arXiv:1701.07875 [cs, stat] (2017) (en), ZSCC: 0002026.
-  Nicolas Bonneel and David Coeurjolly, *SPOT: Sliced Partial Optimal Transport*, ACM Transactions on Graphics (SIGGRAPH) **38** (2019), no. 4.
-  Nicolas Bonnotte, *Unidimensional and evolution methods for optimal transportation*, Ph.D. thesis, Paris 11, 2013.
-  Nicolas Bonneel, Gabriel Peyré, and Marco Cuturi, *Wasserstein barycentric coordinates: Histogram regression using optimal transport*, ACM Transactions on Graphics **35** (2016), no. 4, 1–10 (en), ZSCC: 0000043.
-  Yann Brenier, *Polar factorization and monotone rearrangement of vector-valued functions*, Communications on pure and applied mathematics **44** (1991), no. 4, 375–417.
-  Luis A Caffarelli, Sergey A Kochengin, and Vladimir I Oliker, *Problem of reflector design with given far-field scattering data*, Monge Ampère Equation: Applications to Geometry and Optimization: NSF-CBMS Conference on the Monge Ampère Equation, Applications to Geometry and Optimization, July 9-13, 1997, Florida Atlantic University, vol. 226, American Mathematical Soc., 1999, p. 13.

References II

-  Marco Cuturi, *Sinkhorn distances: Lightspeed computation of optimal transport*, Advances in Neural Information Processing Systems, 2013, ZSCC: 0000598, pp. 2292–2300.
-  Jean Feydy, Benjamin Charlier, François-Xavier Vialard, and Gabriel Peyré, *Optimal Transport for Diffeomorphic Registration*, arXiv:1706.05218 [math] (2017) (en), ZSCC: 0000020 arXiv: 1706.05218.
-  Jun Kitagawa, Quentin Mérigot, and Boris Thibert, *Convergence of a newton algorithm for semi-discrete optimal transport*, Journal of the European Mathematical Society (2019).
-  Quentin Mérigot, *A multiscale approach to optimal transport*, Computer Graphics Forum, vol. 30, Wiley Online Library, 2011, pp. 1583–1592.
-  Jean-Marie Mirebeau, *Discretization of the 3D Monge-Ampere operator, between Wide Stencils and Power Diagrams*, arXiv preprint arXiv:1503.00947 (2015).
-  Quentin Mérigot, Jocelyn Meyron, and Boris Thibert, *An algorithm for optimal transport between a simplex soup and a point cloud*, SIAM Journal on Imaging Sciences **11** (2018), no. 2, 1363–1389.

References III



Antoine Rolet, Marco Cuturi, and Gabriel Peyré, *Fast Dictionary Learning with a Smoothed Wasserstein Loss*, Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (Cadiz, Spain) (Arthur Gretton and Christian C. Robert, eds.), Proceedings of Machine Learning Research, vol. 51, PMLR, May 2016, ZSCC: 0000059[s3], pp. 630–638.



Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot, *Wasserstein barycenter and its application to texture mixing*, International Conference on Scale Space and Variational Methods in Computer Vision, Springer, 2011, pp. 435–446.



Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas, *Convolutional Wasserstein Distances: Efficient Optimal Transportation on Geometric Domains*, ACM Trans. Graph. **34** (2015), no. 4, 66:1–66:11, ZSCC: 0000217.



Morgan A Schmitz, Matthieu Heitz, Nicolas Bonneel, Fred Maurice Ngolè Mboula, David Coeurjolly, Marco Cuturi, Gabriel Peyré, and Jean-Luc Starck, *Wasserstein Dictionary Learning: Optimal Transport-based unsupervised non-linear dictionary learning*, SIAM Journal on Imaging Sciences (2018), ZSCC: 0000000[s4].