Introduction to Optimal Transport

For applications in machine learning

Nicolas Courty
Team Leader Obelix Team
PR UBS / IRISA

Lecture OCEANIX - 29th November 2022

Introduction

Measure and probability distributions are at the core of Machine learning and computational geometry

A point of view on the data

Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}$; $\mathbf{x}_i \in \mathbb{R}^d$ $\longleftrightarrow$ A probability distribution describing the data

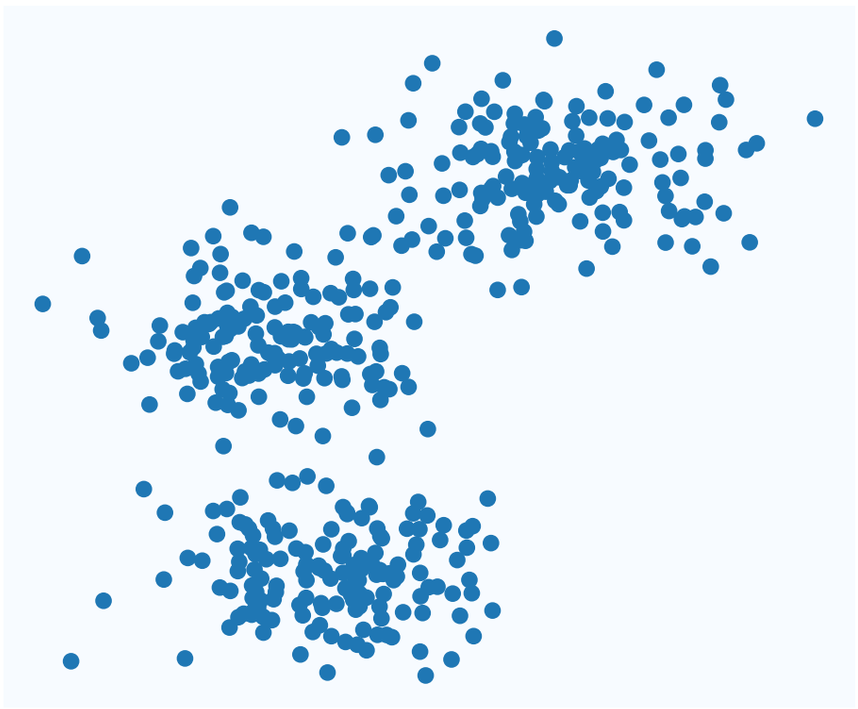
Introduction

Measure and probability distributions are at the core of Machine learning and computational geometry

A point of view on the data

Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}\ ; \mathbf{x}_i \in \mathbb{R}^d \quad \longleftrightarrow \quad$ A probability distribution describing the data

Lagrangian: $\frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}$



(point clouds)

$\delta_{\mathbf{x}_i}(\mathbf{x}) = 1 \text{ if } \mathbf{x} = \mathbf{x}_i \text{ else } 0$

Introduction

Measure and probability distributions are at the core of Machine learning and computational geometry

A point of view on the data

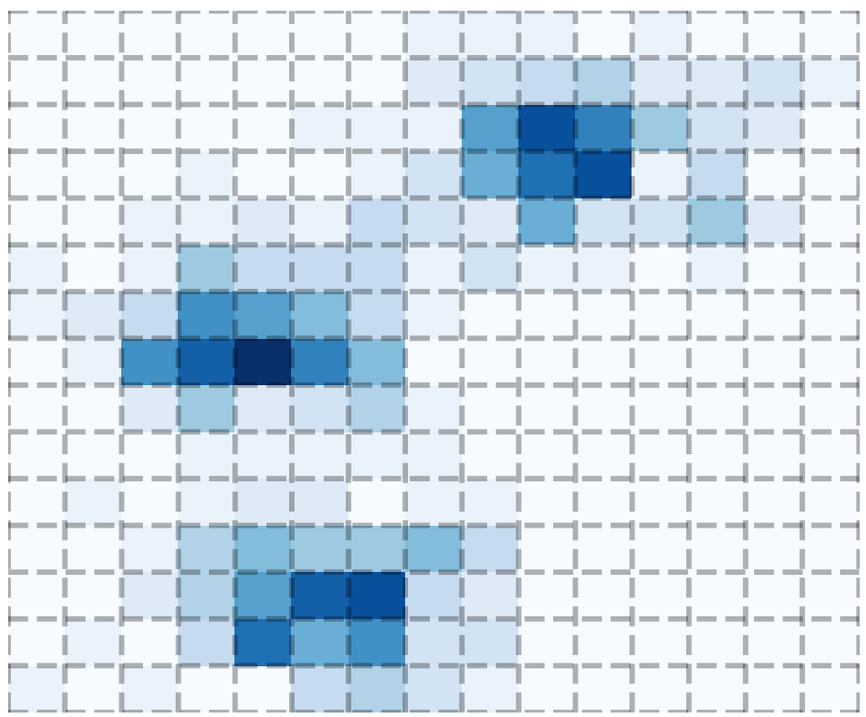
Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}\ ; \mathbf{x}_i \in \mathbb{R}^d \quad \longleftrightarrow \quad$ A probability distribution describing the data

Lagrangian: $\frac{1}{n}\sum_{i=1}^n \delta_{x_i}$

(point clouds)

$\delta_{\mathbf{x}_i}(\mathbf{x}) = 1$ if $\mathbf{x} = \mathbf{x}_i$ else 0

Eulerian: $\sum_{i=1}^n a_i \delta_{x_i}$

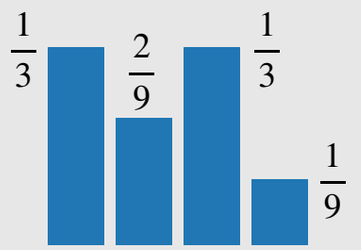


(histograms)

Simplex

$\mathbf{a} = (a_i)_{i\in[\![n]\!]} \in \Sigma_n$

$a_i \geq 0, \sum_{i=1}^n a_i = 1$



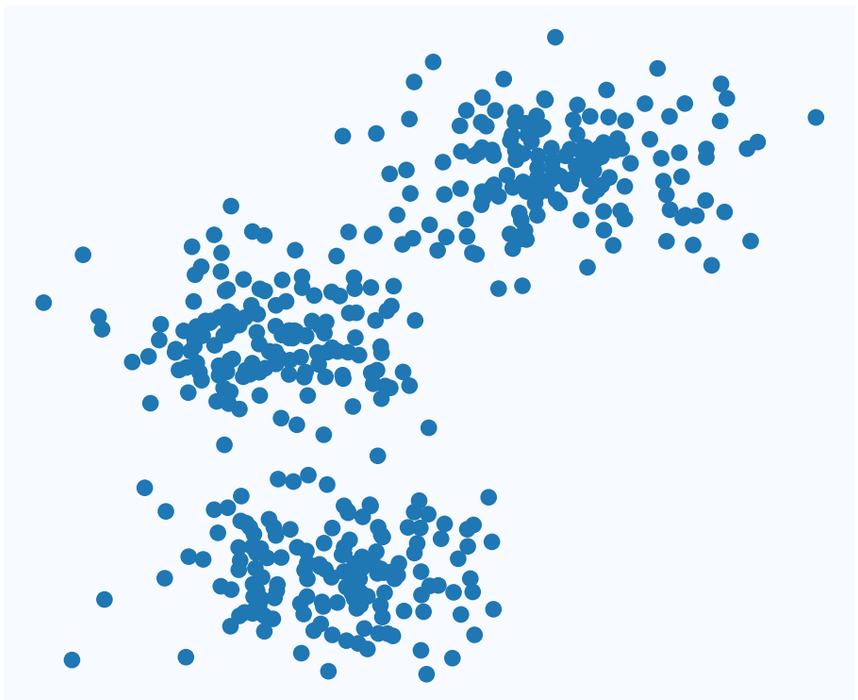
Introduction

Measure and probability distributions are at the core of Machine learning and computational geometry

A point of view on the data

Data: $(\mathbf{x}_i)_{i \in [\![n]\!]} \; ; \mathbf{x}_i \in \mathbb{R}^d \quad \longleftrightarrow \quad$ A probability distribution describing the data

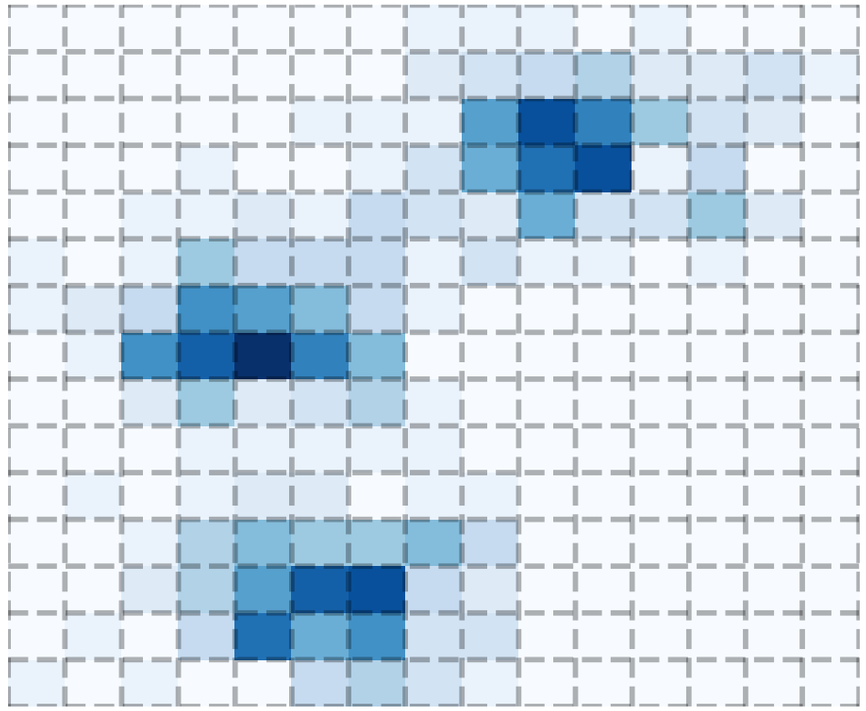
Lagrangian: $\frac{1}{n} \sum_{i=1}^n \delta_{x_i}$



(point clouds)

$\delta_{\mathbf{x}_i}(\mathbf{x}) = 1$ if $\mathbf{x} = \mathbf{x}_i$ else 0

Eulerian: $\sum_{i=1}^n a_i \delta_{x_i}$

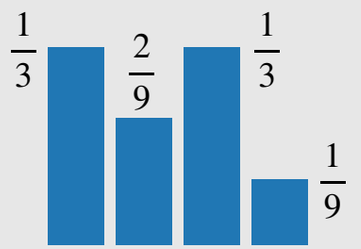


(histograms)

Simplex

$\mathbf{a} = (a_i)_{i \in [\![n]\!]} \in \Sigma_n$

$a_i \geq 0, \sum_{i=1}^n a_i = 1$



Knowing the probability = knowing the data

Introduction

Measure and probability distributions are at the core of Machine learning and computational geometry

A point of view on the data

Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}$; $\mathbf{x}_i \in \mathbb{R}^d$ $\longleftrightarrow$ A probability distribution describing the data

A formalism for many machine learning paradigms

(ERM) $\min_f \mathbb{E}_{(x,y)\sim\mu}[L(f(x),y)]$

Introduction

Measure and probability distributions are at the core of Machine learning and computational geometry

A point of view on the data

Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}$; $\mathbf{x}_i \in \mathbb{R}^d$ $\longleftrightarrow$ A probability distribution describing the data

A formalism for many machine learning paradigms

(ERM) $\min_f \mathbb{E}_{(x,y)\sim\mu}[L(f(x),y)]$ $\xrightarrow{\text{follow the law given by the prob.}}$ $\mu = \frac{1}{n}\sum_{i=1}^n \delta_{(\mathbf{x}_i,y_i)}$

Introduction

Measure and probability distributions are at the core of Machine learning and computational geometry

A point of view on the data

Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}$; $\mathbf{x}_i \in \mathbb{R}^d$ $\longleftrightarrow$ A probability distribution describing the data

A formalism for many machine learning paradigms

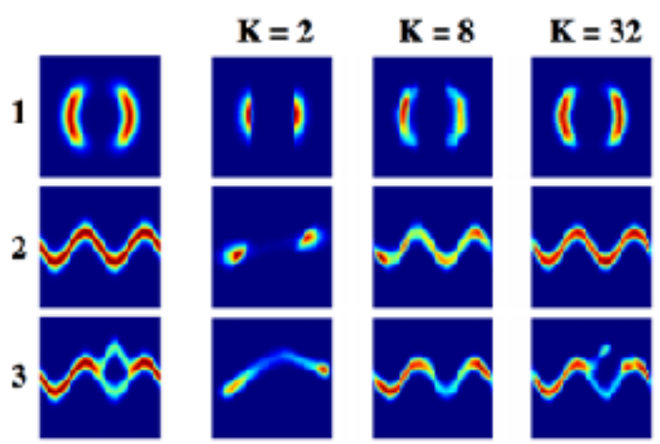
(ERM) $\min_{f} \mathbb{E}_{(x,y)\sim\mu}[L(f(x),y)]$

(Likehood) $\max_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\mathbf{x}\sim\mu}[\log(\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}))]$

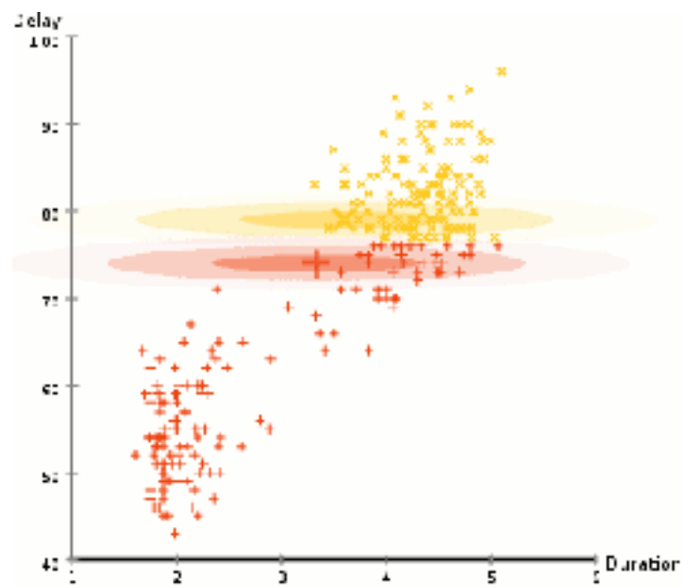
Data = discrete probability measures

$\mu = \frac{1}{n}\sum_{i=1}^{n} \delta_{(\mathbf{x}_i,y_i)}$

$\mu = \frac{1}{n}\sum_{i=1}^{n} \delta_{\mathbf{x}_i}$



[Rezende 2016]



Introduction

Measure and probability distributions are at the core of Machine learning and computational geometry

A point of view on the data

Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}$; $\mathbf{x}_i \in \mathbb{R}^d$ $\longleftrightarrow$ A probability distribution describing the data

A formalism for many machine learning paradigms

(ERM)	$\min_{f} \mathop{\mathbb{E}}_{(x,y)\sim\mu}[L(f(x),y)]$
(Likehood)	$\max_{\boldsymbol{\theta}\in\Theta} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mu}[\log(\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}))]$
(GAN)	$\min_{\boldsymbol{\theta}\in\Theta} D(\mu_{\boldsymbol{\theta}},\nu)$

Data = discrete probability measures

$$\mu = \frac{1}{n}\sum_{i=1}^{n} \delta_{(\mathbf{x}_i,y_i)}$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} \delta_{\mathbf{x}_i}$$

[Radford 2015]

Introduction

Measure and probability distributions are at the core of Machine learning and computational geometry

A point of view on the data

Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}$; $\mathbf{x}_i \in \mathbb{R}^d$ $\longleftrightarrow$ A probability distribution describing the data

A formalism for many machine learning paradigms

(ERM) $\min_{f} \mathop{\mathbb{E}}_{(x,y)\sim\mu}[L(f(x),y)]$

(Likehood) $\max_{\boldsymbol{\theta}\in\Theta} \mathop{\mathbb{E}}_{\mathbf{x}\sim\mu}[\log(\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}))]$

(GAN) $\min_{\boldsymbol{\theta}\in\Theta} D(\mu_{\boldsymbol{\theta}}, \nu)$

(Signal processing) $\min_{\mu\in\mathcal{M}(\Theta)} \frac{1}{2}\|\mathbf{y} - \phi * \mu\|_{L^2}^2 + R(\mu)$

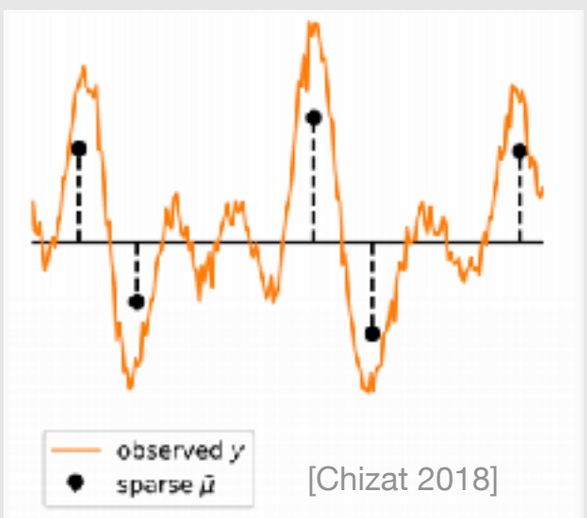
Data = discrete probability measures

$\mu = \frac{1}{n}\sum_{i=1}^n \delta_{(\mathbf{x}_i, y_i)}$

$\mu = \frac{1}{n}\sum_{i=1}^n \delta_{\mathbf{x}_i}$

Recover a sparse signal

$\overline{\mu} = \sum_i w_i \delta_{\boldsymbol{\theta}_i}$



[Chizat 2018]

Introduction

Measure and probability distributions are at the core of Machine learning and computational geometry

A point of view on the data

Data: $(\mathbf{x}_i)_{i\in[\![n]\!]}$; $\mathbf{x}_i \in \mathbb{R}^d$ $\longleftrightarrow$ A probability distribution describing the data

A formalism for many machine learning paradigms

(ERM) $\min_f \mathbb{E}_{(x,y)\sim\mu}[L(f(x),y)]$	Data = discrete probability measures $\mu = \frac{1}{n}\sum_{i=1}^n \delta_{(\mathbf{x}_i,y_i)}$
(Likehood) $\max_{\boldsymbol{\theta}\in\Theta} \mathbb{E}_{\mathbf{x}\sim\mu}[\log(\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}))]$	$\mu = \frac{1}{n}\sum_{i=1}^n \delta_{\mathbf{x}_i}$
(GAN) $\min_{\boldsymbol{\theta}\in\Theta} D(\mu_{\boldsymbol{\theta}},\nu)$	
(Signal processing) $\min_{\mu\in\mathcal{M}(\Theta)} \frac{1}{2}\|\mathbf{y}-\phi*\mu\|_{L^2}^2 + R(\mu)$	

Needs an appropriate way of comparing probability distributions

Overview of the course

- Visual Introduction to OT
- Computational aspects of OT
- Variants of OT
- Gradient flows in probability spaces

Introduction to OT

With images

What is Optimal Transport ?

The natural geometry for probability measures


Monge


Kantorovic


Koopman

Nobel ’75


Dantzi


Brenier


Otto


McCann


Villani

Fields ’10


Figalli

Fields ’18

Origins: Monge Problem (1781)

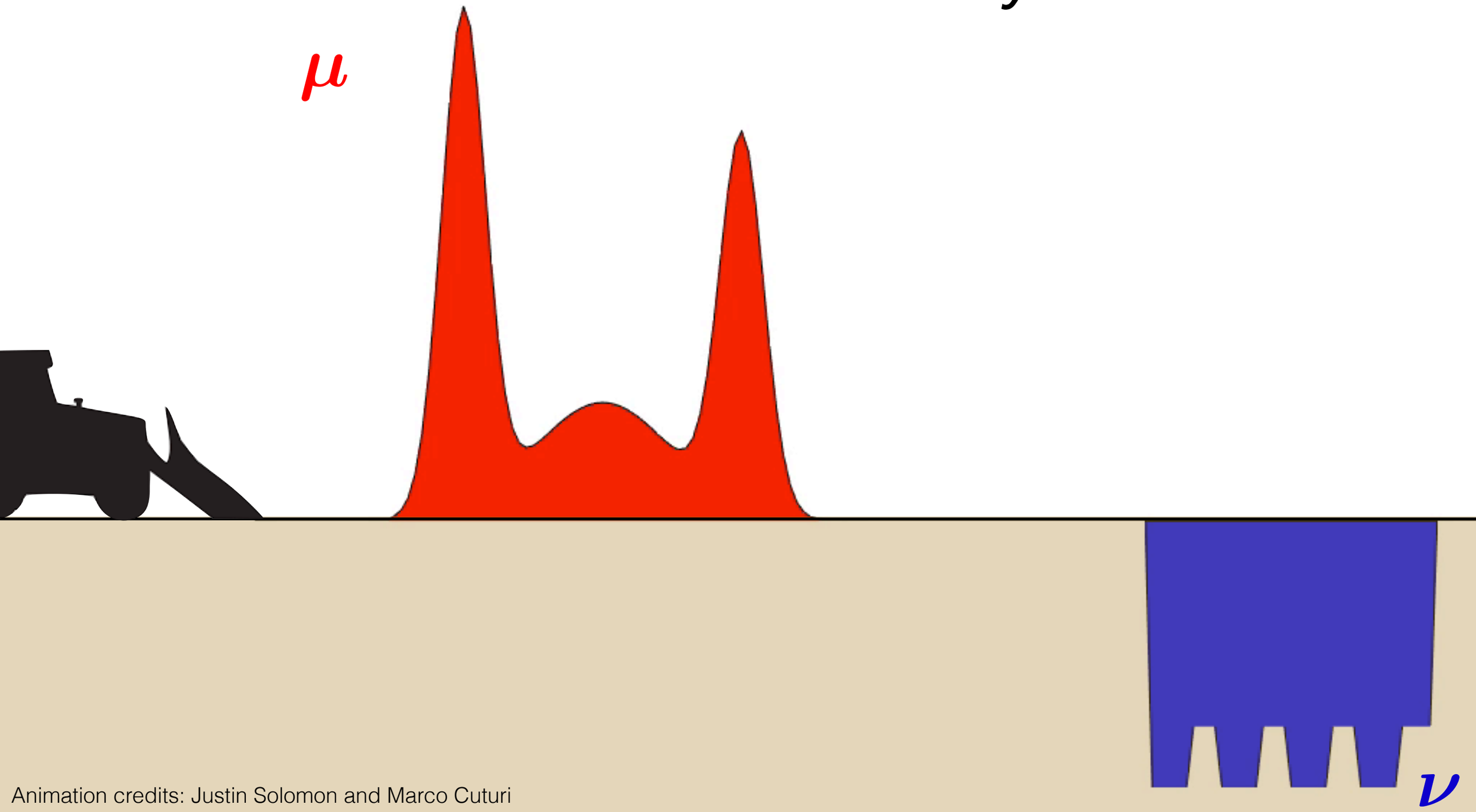
MÉMOIRE
SUR LA
THÉORIE DES DÉBLAIS
ET DES REMBLAIS.

Par M. MONGE.

LORSQU'ON doit transporter des terres d'un lieu dans un autre, on a coutume de donner le nom de *Déblai* au volume des terres que l'on doit transporter, & le nom de *Remblai* à l'espace qu'elles doivent occuper après le transport.

Origins: Monge Problem

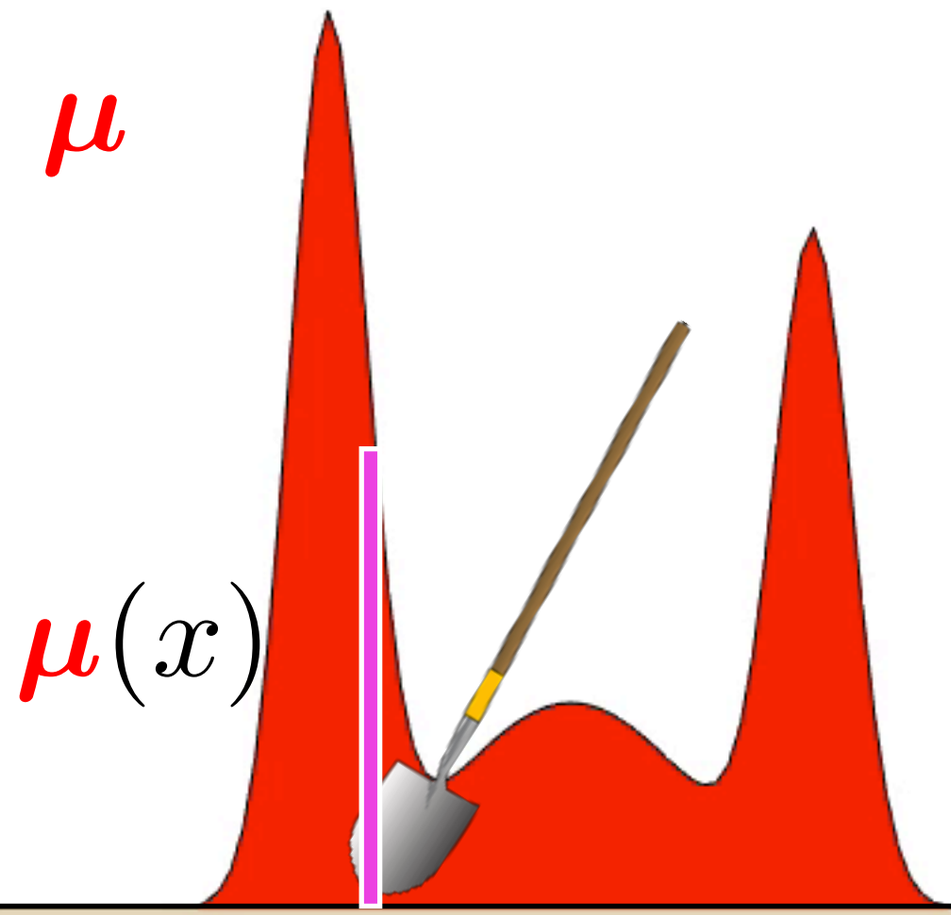
In the 21st Century…



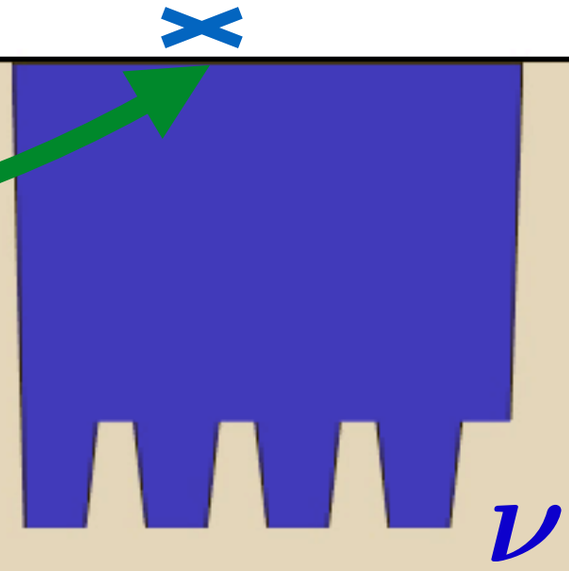
Animation credits: Justin Solomon and Marco Cuturi

Origins: Monge's Problem

In 1781 however…



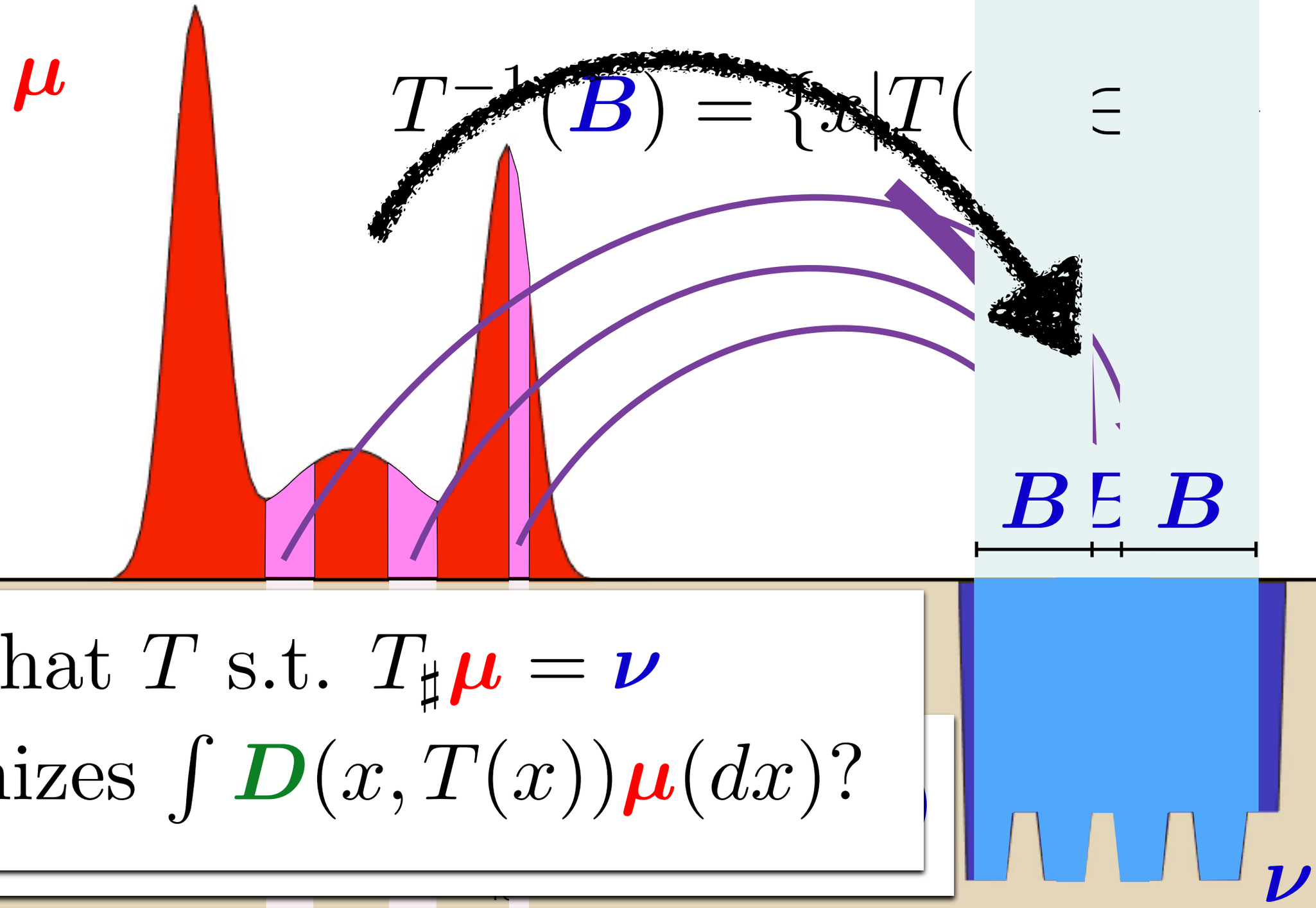
$$y = T(x)$$



work: $\mu(x)D(x, T(x))$

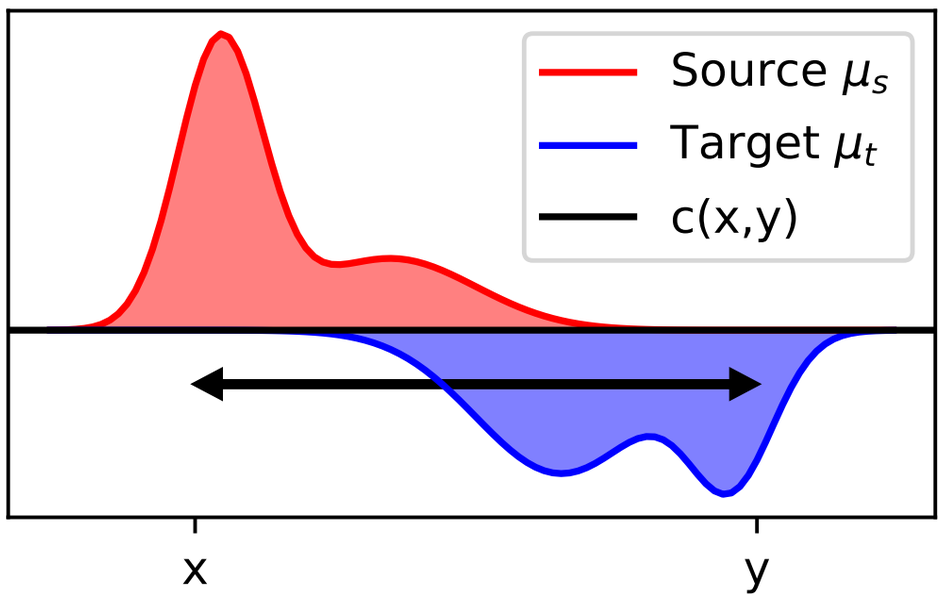
Origins: Monge's Problem

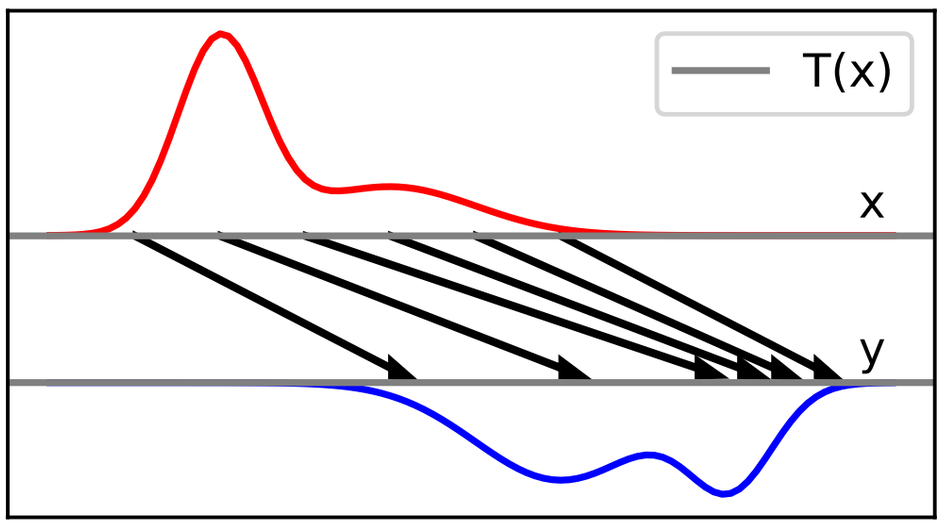
T must **push-forward** *the mass to the blue*



What T s.t. $T_\sharp \boldsymbol{\mu} = \boldsymbol{\nu}$
minimizes $\int \boldsymbol{D}(x, T(x)) \boldsymbol{\mu}(dx)$?

The origins of optimal transport

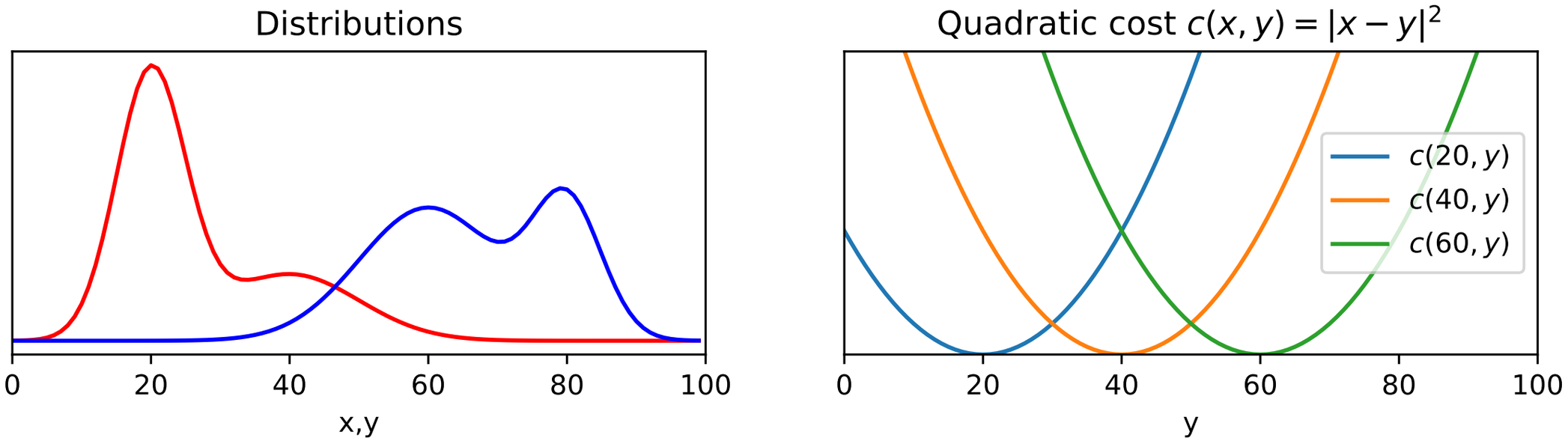




Problem [Monge, 1781]

- How to move dirt from one place (déblais) to another (remblais) while minimizing the effort ?
- Find a mapping T between the two distributions of mass (transport).
- Optimize with respect to a displacement cost $c(x, y)$ (optimal).

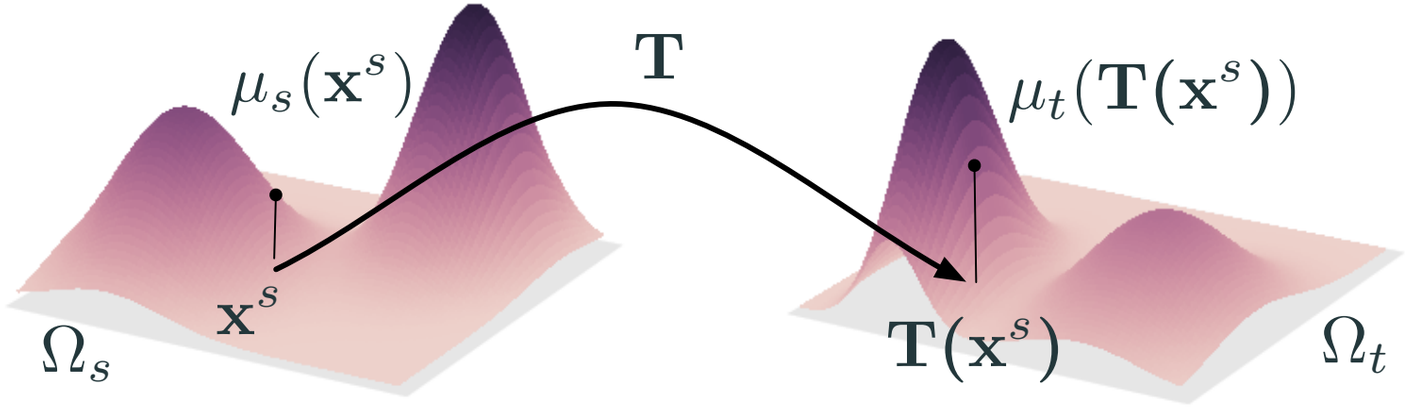
Optimal transport (Monge formulation)



- Probability measures μ_s and μ_t on and a cost function $c : \Omega_s \times \Omega_t \rightarrow \mathbb{R}^+$.
- The Monge formulation [Monge, 1781] aim at finding a mapping $T : \Omega_s \rightarrow \Omega_t$

$$\inf_{T\#\mu_s = \mu_t} \quad \int_{\Omega_s} c(\mathbf{x}, T(\mathbf{x}))\mu_s(\mathbf{x})d\mathbf{x} \tag{1}$$

What is $T\#\mu_s = \mu_t$?



- $T\#$ is the so called push forward operator
- it transfers measures from one space Ω_s to another space Ω_t
- it is equivalent to:

$$\mu_t(A) = \mu_s(T^{-1}(A))$$

$$\int_{\Omega_t} g(y) d\mu_t(y) = \int_{\Omega_s} g(T(x)) d\mu_s(x)$$

- for smooth measures $\mu_s = \rho(x)dx$ and $\mu_t = \eta(x)dx$

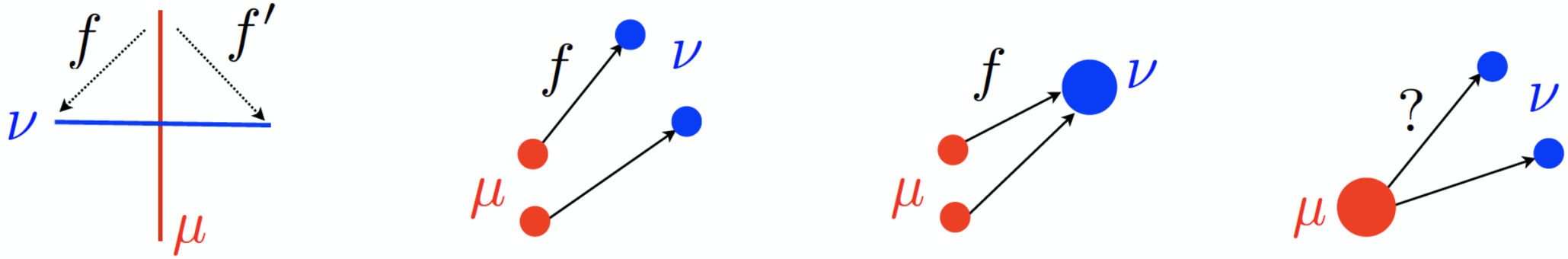
$$T\#\mu_s = \mu_t \equiv \rho(T(x))|\mathsf{det}(\partial T(x))| = \eta(x)$$

- a.k.a. change of variable formula

Non-existence / Non-uniqueness

Solving for this push-forward operator is a non-convex optimization problem,

- for which existence is not guaranteed,
- nor unicity



Note: [Brenier, 1991] proved existence and unicity of the Monge map for $c(x, y) = \|x - y\|^2$ and distributions with densities (i.e. continuous).

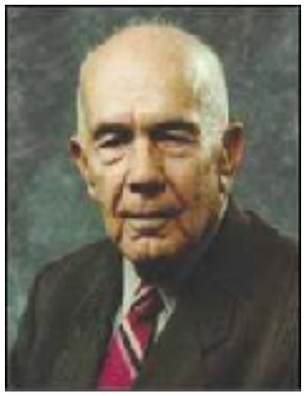
Kantorovich Problem

Kantorovich

1939

Tolstoi

1930



Hitchcock

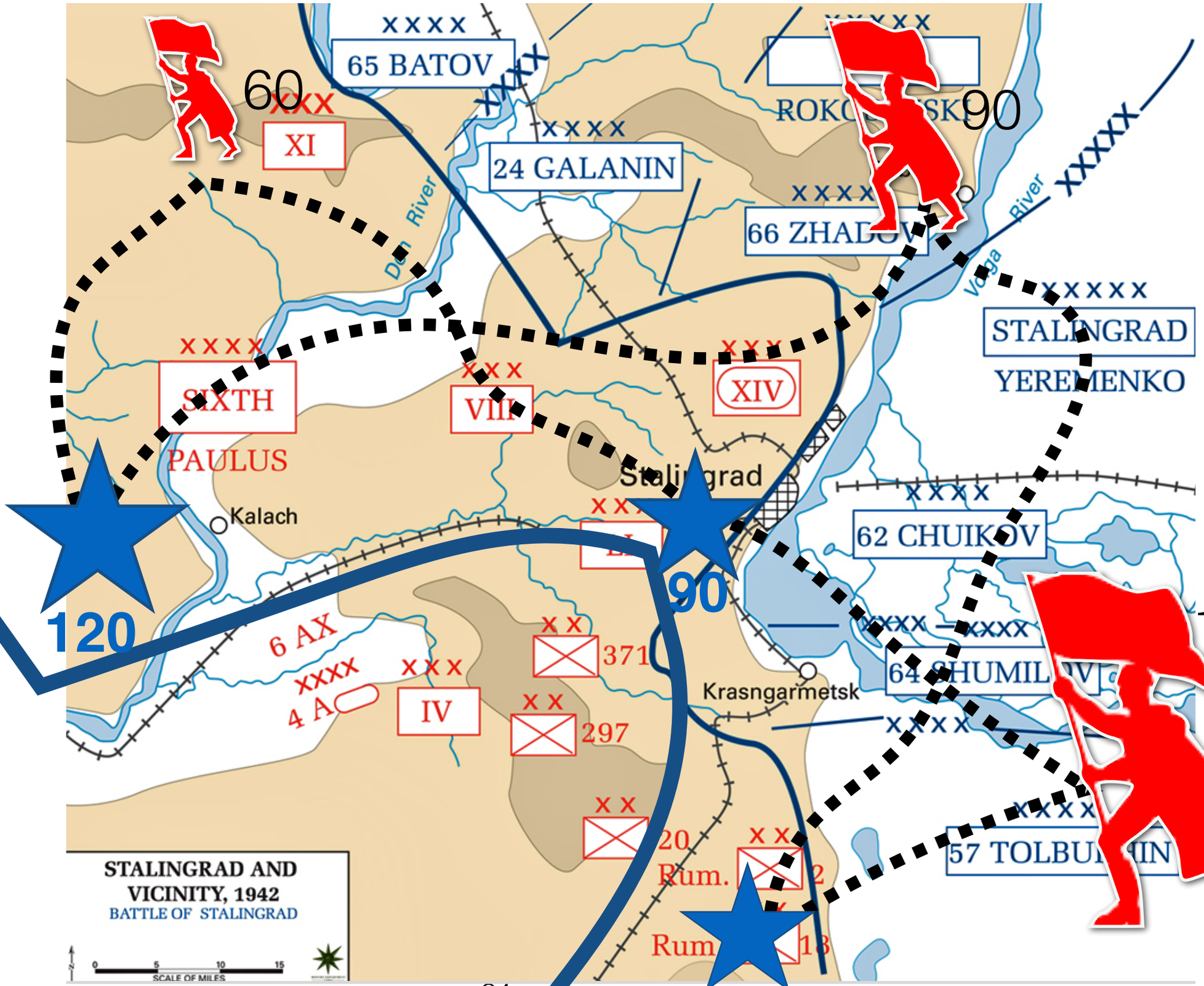
THE DISTRIBUTION OF A PRODUCT FROM SEVERAL SOURCES TO NUMEROUS LOCALITIES

By Frank L. Hitchcock

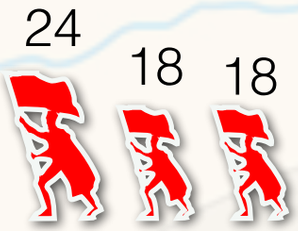
1. Statement of the problem. When several factories supply a product to a number of cities we desire the least costly manner of distribution. Due to freight rates and other matters the cost of a ton of product to a particular city will vary according to which factory supplies it, and will also vary from city to city.

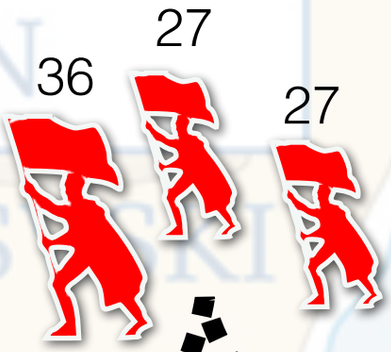
1941

Kantorovich Problem



Kantorovich Problem

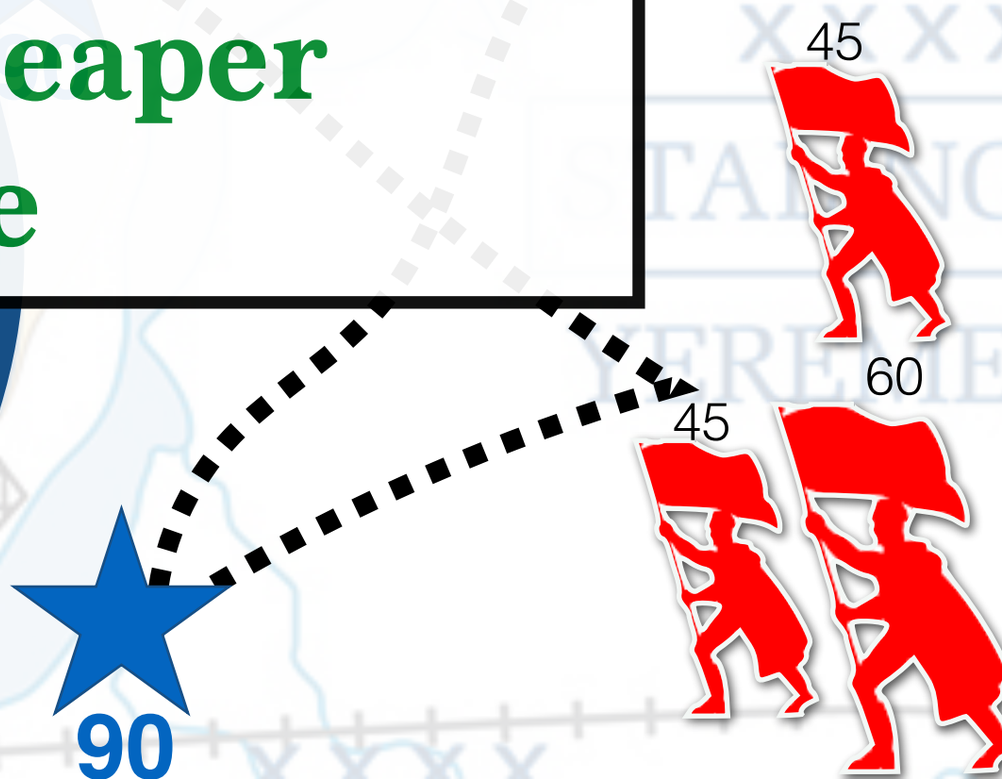




Naive approach results in too many displacements.

Goal: find a cheaper alternative

Easy solution: split the task with proportions 120:90:90 = 4:3:3



Kantorovich Problem

60

90

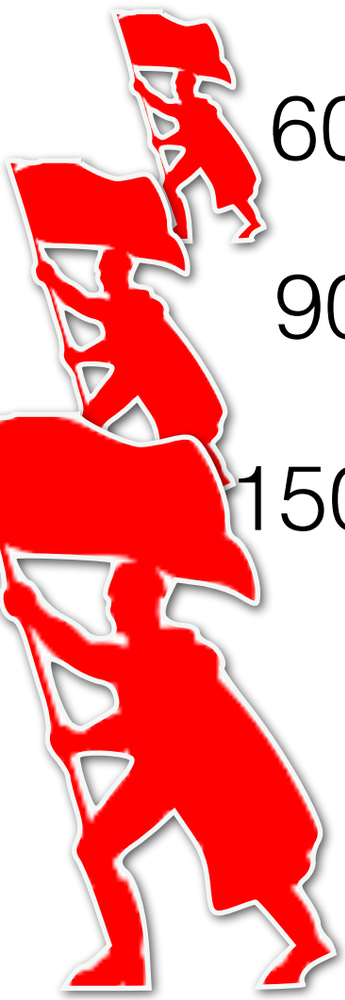
120

90

150

90

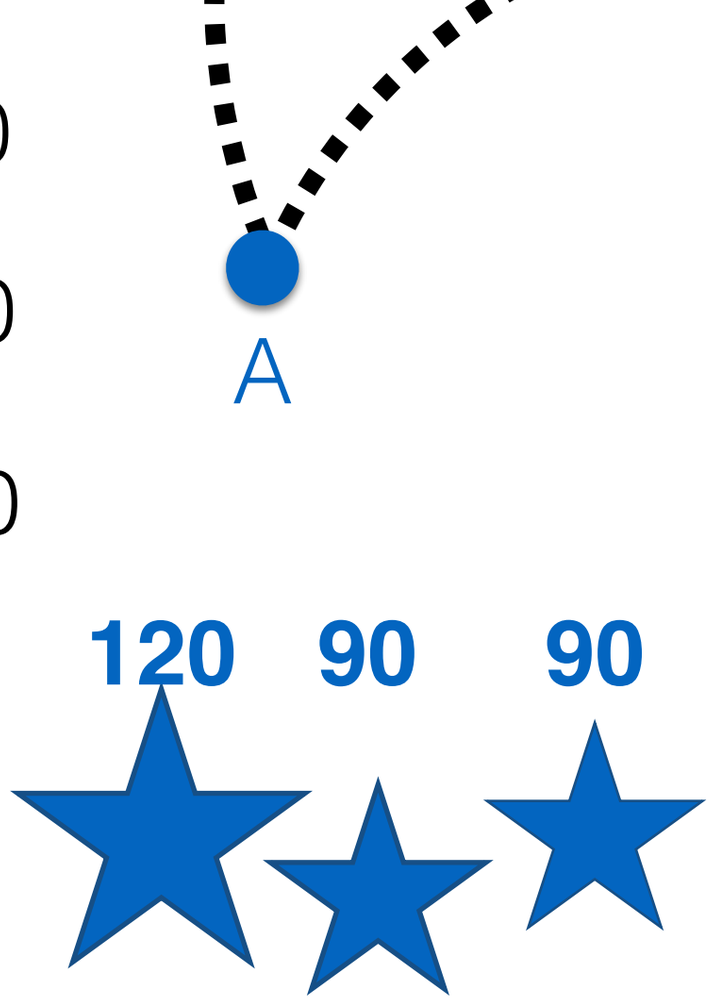
Kantorovich Problem



60

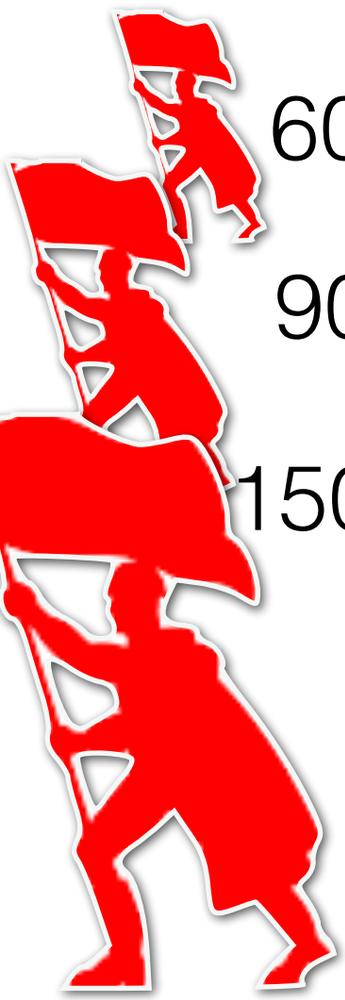
90

150



120 90 90

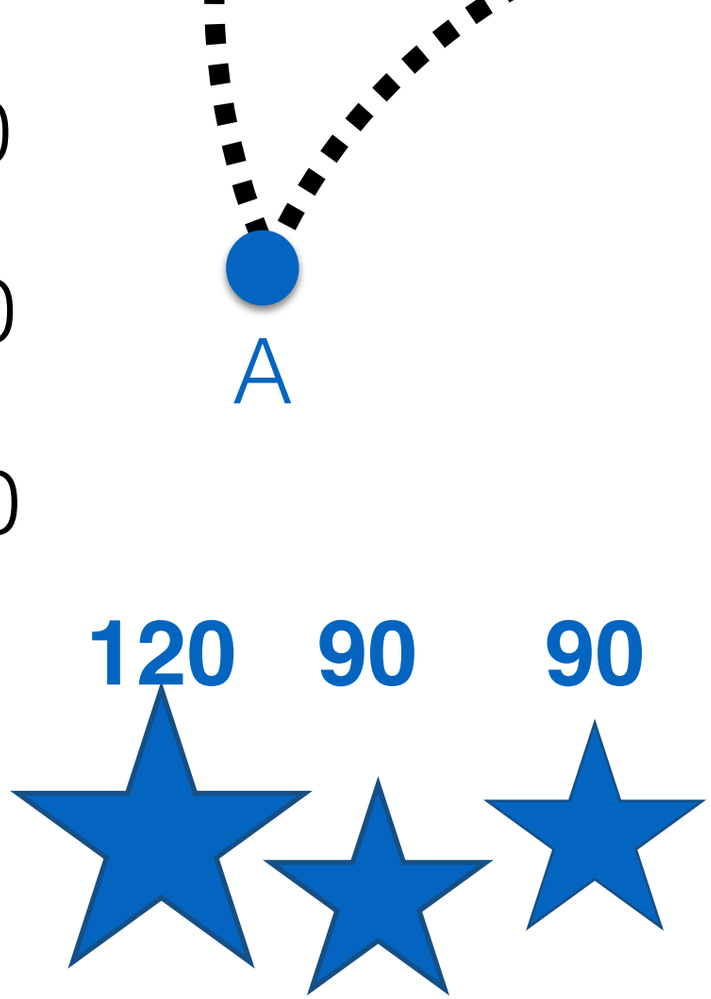
Kantorovich Problem



60

90

150



120 90 90

Kantorovich Problem

The problem is entirely described by ***counts*** *and a* ***cost/distance matrix***

Transportation matrix

60	?	?	?
90	?	?	?
150	?	?	?
	120	**90**	**90**

Distance matrix

1	d_{1A}	d_{1B}	d_{1C}
2	d_{2A}	d_{2B}	d_{2C}
3	d_{3A}	d_{3B}	d_{3C}
	A	B	C

Kantorovich Problem

Transportation matrix

60 a_1	p_{1A}	p_{1B}	p_{1C}
90 a_2	p_{2A}	p_{2B}	p_{2C}
150 a_3	p_{3A}	p_{3B}	p_{3C}
	120 b_A	90 b_B	90 b_C

Distance matrix

1	d_{1A}	d_{1B}	d_{1C}
2	d_{2A}	d_{2B}	d_{2C}
3	d_{3A}	d_{3B}	d_{3C}
	A	B	C

Constraints

$$\forall i \in \{1,2,3\}, \sum_{j \in \{A,B,C\}} p_{ij} = a_i$$

$$\forall j \in \{A,B,C\}, \sum_{i \in \{1,2,3\}} p_{ij} = b_j$$

$$p_{ij} \geq 0$$

Cost function

$$C(P) = \sum_{j \in \{A,B,C\}} \sum_{i \in \{1,2,3\}} p_{ij} d_{ij}$$

Problem

$$\min_{\text{all valid } P} C(P)$$

Kantorovich Problem

60

90

120

90

150

90

Kantorovich Problem

30 30

90

120

90

90

60 90

60	30	30	
90	90		
150		60	90
	120	90	90

Kantorovich formulation

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovich formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$

Kantorovich formulation

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovich formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$

Set of couplings/ transport plans

$$\Pi(\mathbf{a}, \mathbf{b})$$

Kantorovich formulation

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovich formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j) \pi_{ij}$$

How much is shifted from x_i to y_j

Kantorovich formulation

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovich formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$

Cost of moving masses from x_i to y_j

Kantorovich formulation

Two probability distributions

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{i=j}^{m} b_j \delta_{y_j}$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovich formulation

$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{i,j=1}^{m,n} c(x_i, y_j)\pi_{ij}$$

Total cost

Kantorovich Formulation

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}),\ \nu \in \mathcal{P}(\mathcal{Y})$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovich formulation

$$\mathcal{T}_c(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) \mathrm{d}\pi(x, y)$$

Kantorovich Formulation

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}),\ \nu \in \mathcal{P}(\mathcal{Y})$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

Kantorovich formulation

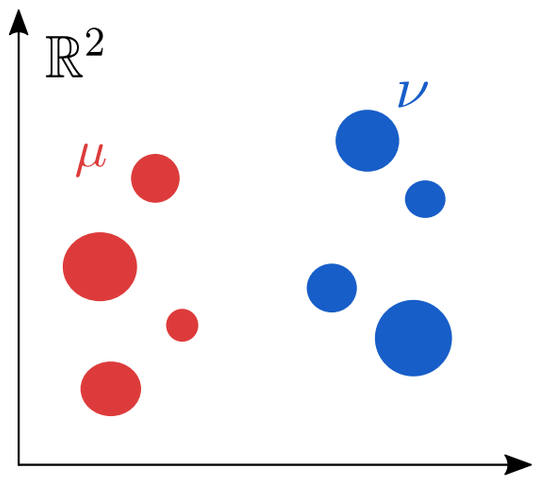
$$\mathcal{T}_c(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) \mathrm{d}\pi(x, y)$$

Set of couplings

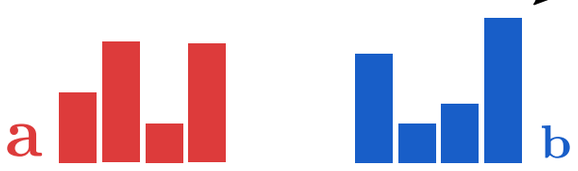
$$\pi \in \Pi(\mu, \nu)$$

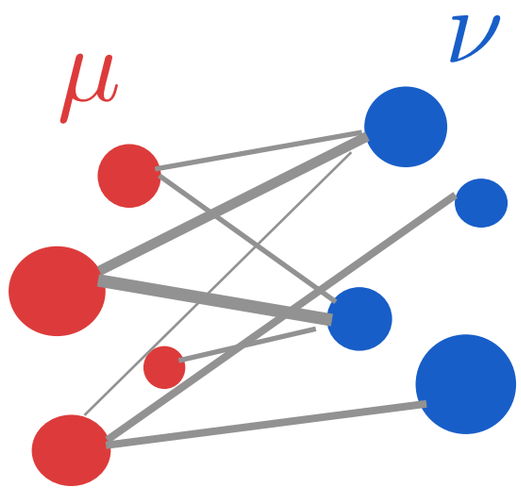
the mass can be split!

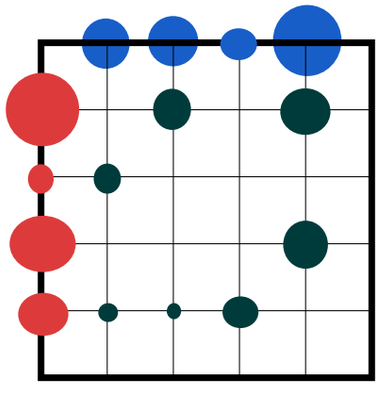
Kantorovich Formulation



$$\Pi(\mu,\nu) = \{\boldsymbol{\pi} \in \mathbb{R}_+^{n\times m} \ | \forall (i,j),\ \sum_{j=1}^{m} \pi_{ij} = a_i \ ; \sum_{i=1}^{n} \pi_{ij} = b_j\}$$







$\pi \in \mathbb{R}_+^{n\times m}$

Kantorovich Formulation

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}),\ \nu \in \mathcal{P}(\mathcal{Y})$$

A cost function

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

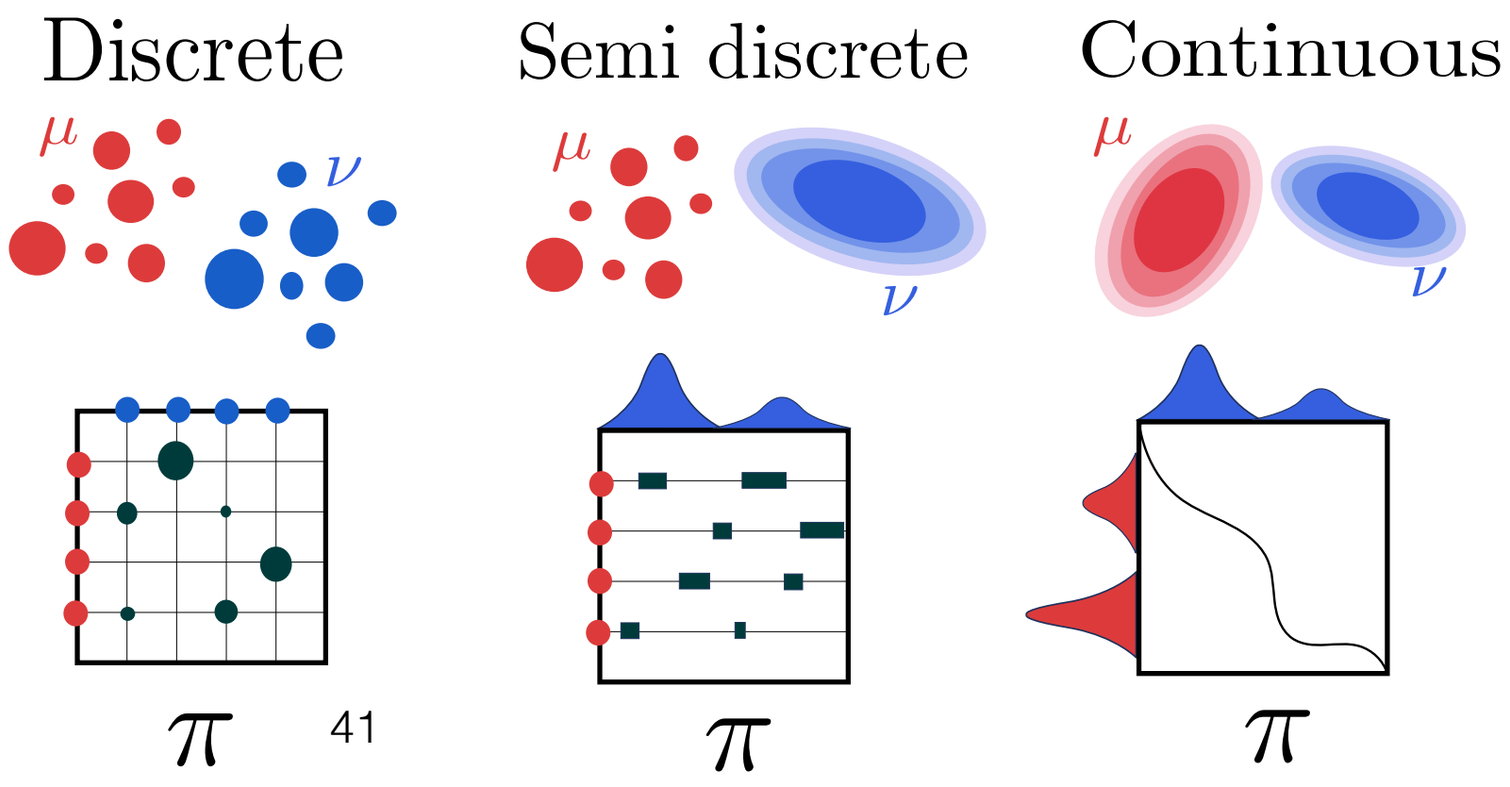
Kantorovich formulation

$$\mathcal{T}_c(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) \mathrm{d}\pi(x, y)$$

Set of couplings

$$\pi \in \Pi(\mu, \nu)$$

the mass can be split!



Wasserstein distance

Two probability distributions

$$\mu \in \mathcal{P}(\Omega), \nu \in \mathcal{P}(\Omega)$$

A distance

$$d : \Omega \times \Omega \to \mathbb{R}_+$$

Wasserstein distance

$$W_p^p(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \int_{\Omega \times \Omega} d^p(x, y) \mathrm{d}\pi(x, y)$$

$\mathcal{P}(\Omega)$ is a metric space

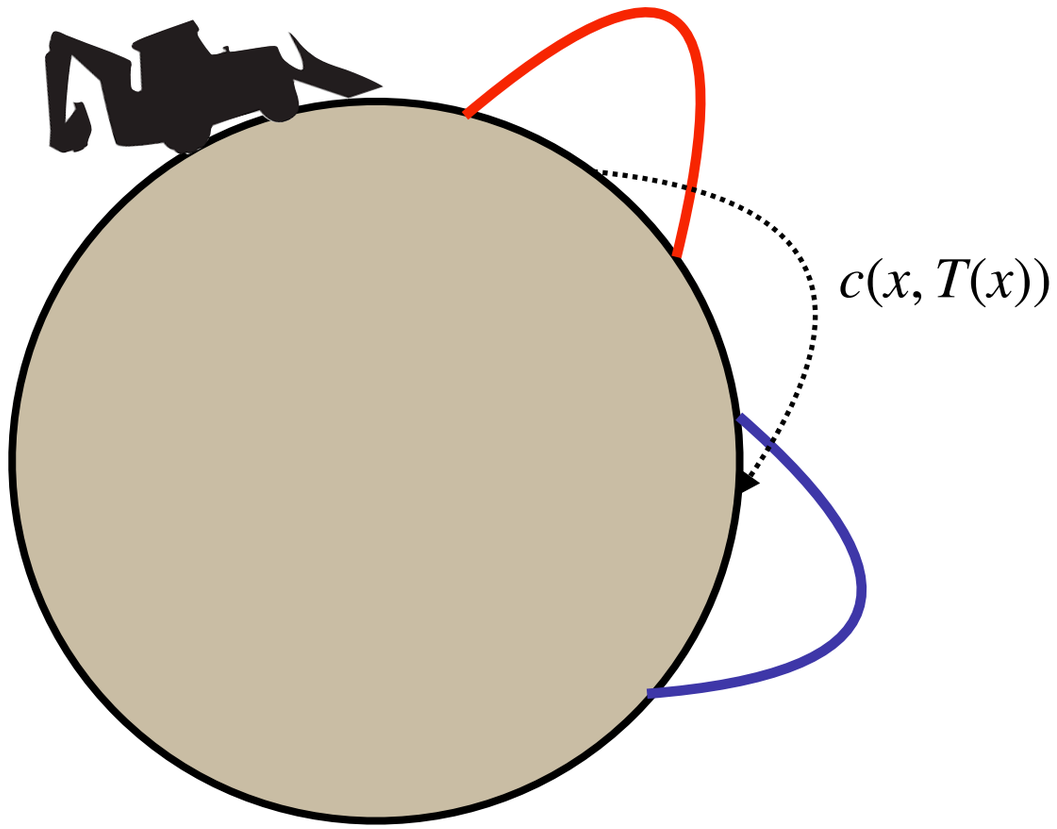
$$W_p(\mu, \nu) = 0 \iff \mu = \nu$$

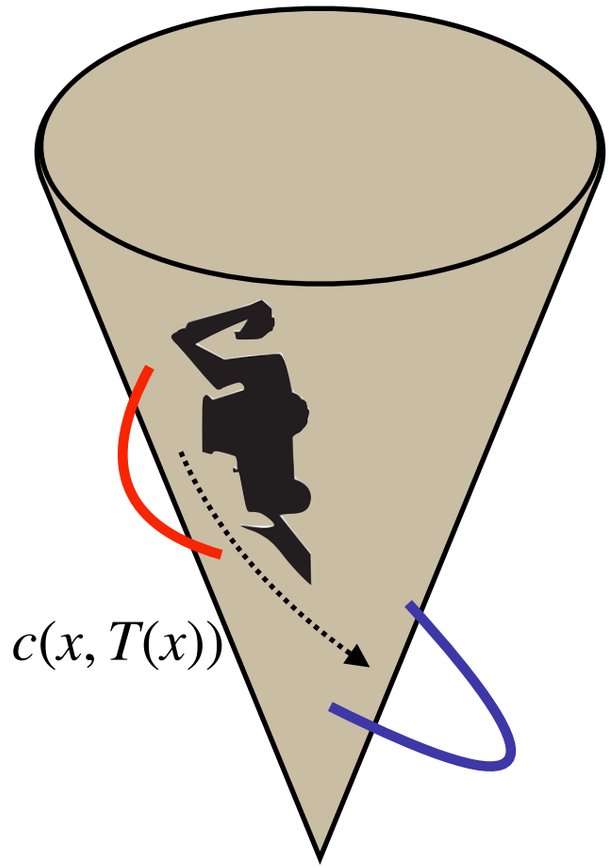
Powerful tool for comparing probability distributions on the same space

- The Wasserstein distance ‘metricizes’ the convergence in probability (weak convergence)

Manifold cases

- Transport can happen on non-flat manifolds
  - The **cost function** $c(\,.\,,.\,)$ encodes implicitly the geometry





Wasserstein space

The space of probability distribution equipped with the Wasserstein metric ($\mathcal{P}_p(X)$, $W_2^2(X)$) defines a geodesic space with a Riemannian structure [Santambrogio, 2014].

- Geodesics are shortest curves on $\mathcal{P}_p(X)$ that link two distributions

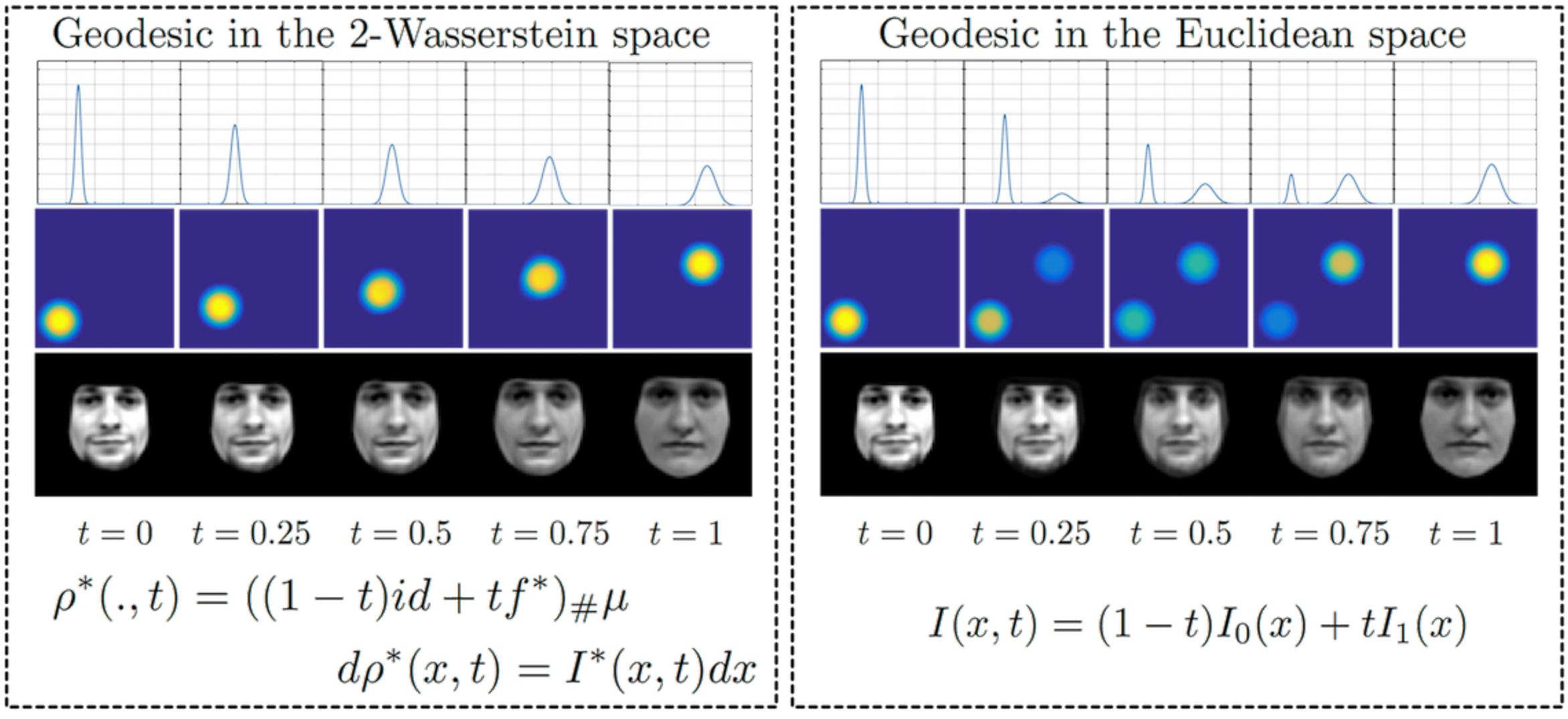
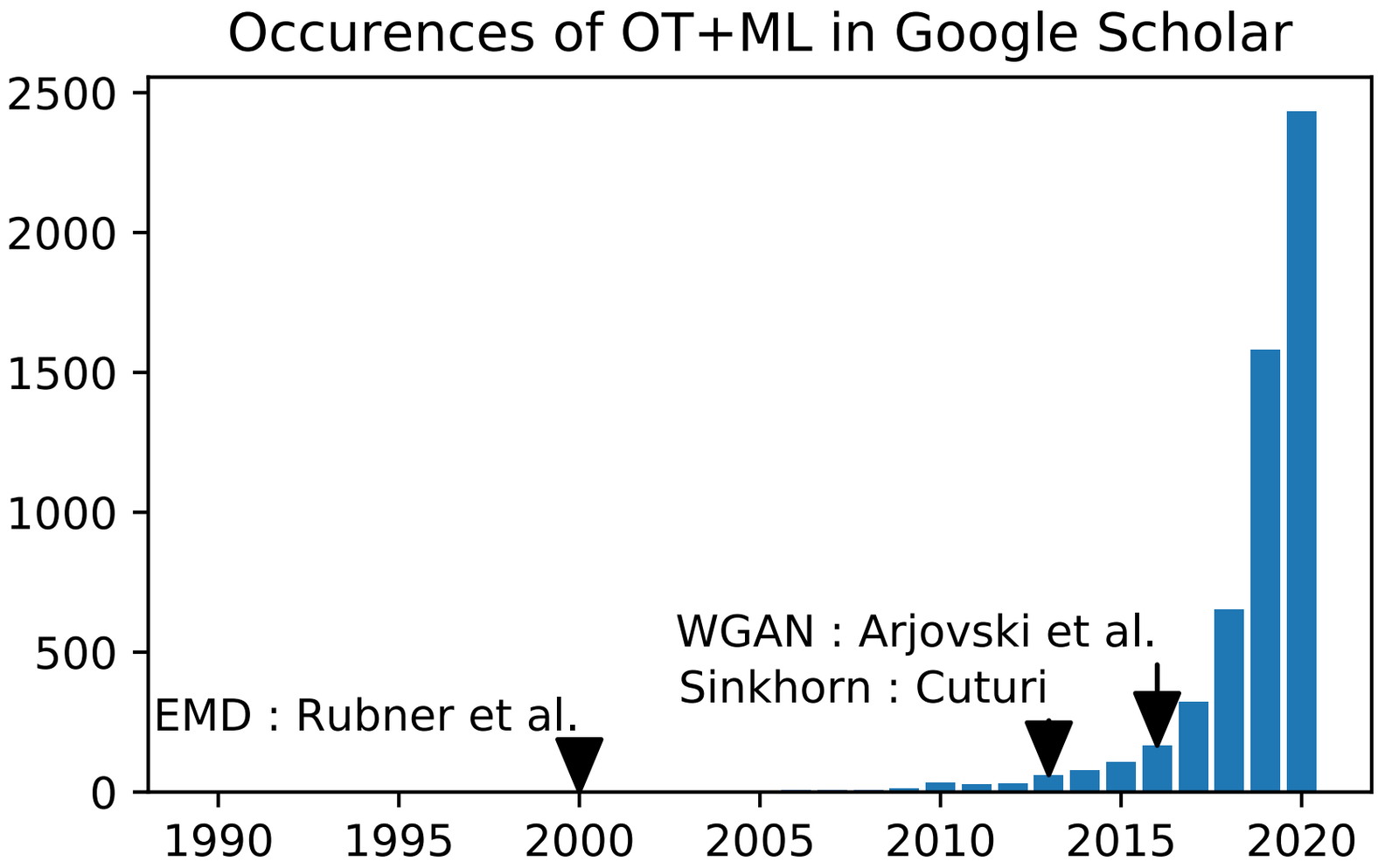
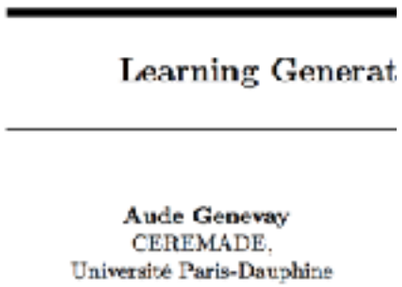


Illustration by S. Kolhouri

Applications of Wasserstein distances

- Numerous applications in Machine Learning and Computer graphics



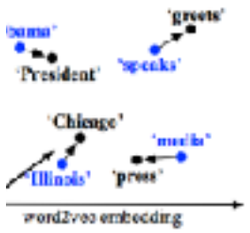

Learning Generat

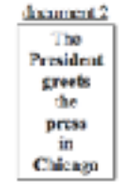
Aude Genevay
CEREMADE,
Université Paris-Dauphine

Optimal Transpo

Nicolas Courty, Rémi Fl
Alain Rakc

Wasserstei




The President greets the press in Chicago

Justin Solomon
Raif M. Rustamov
Leonidas Guibas
RUSTAMOV@STANFORD.EDU
GUIBAS@CS.STANFORD.EDU
Department of Computer Science, Stanford University, 353 Serra Mall, Stanford, California 94305 USA

Adrian Butscher
ADRIAN.BUTSCHER@GMAIL.COM
Max Planck Center for Visual Computing and Communication, Campus E1 4, 66123 Saarbrücken, Germany

frogner@mit.edu, chiyuan@mit.edu
hmobahi@csail.mit.edu

Mauricio Araya-Polo
Shell International E & P Inc.
Mauricio.Araya@shell.com

Tomaso Poggio
Center for Brains, Minds and Machines
Massachusetts Institute of Technology
tp@ai.mit.edu


Siberian husky


Eskimo dog

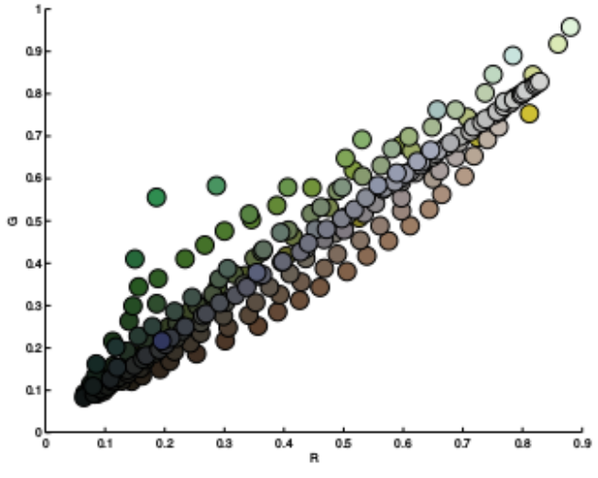
Histogram matching in images : color grading

Pixels as empirical distribution [Ferradans et al., 2014]

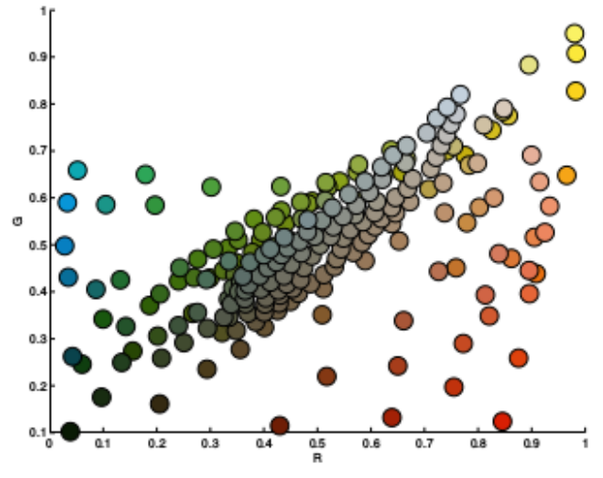
X^0

Y^0

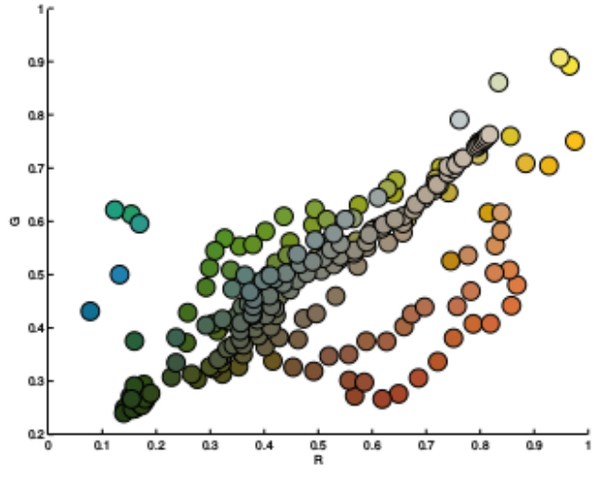
$\tilde{X}^0$



μ_{X^0}



μ_{Y^0}

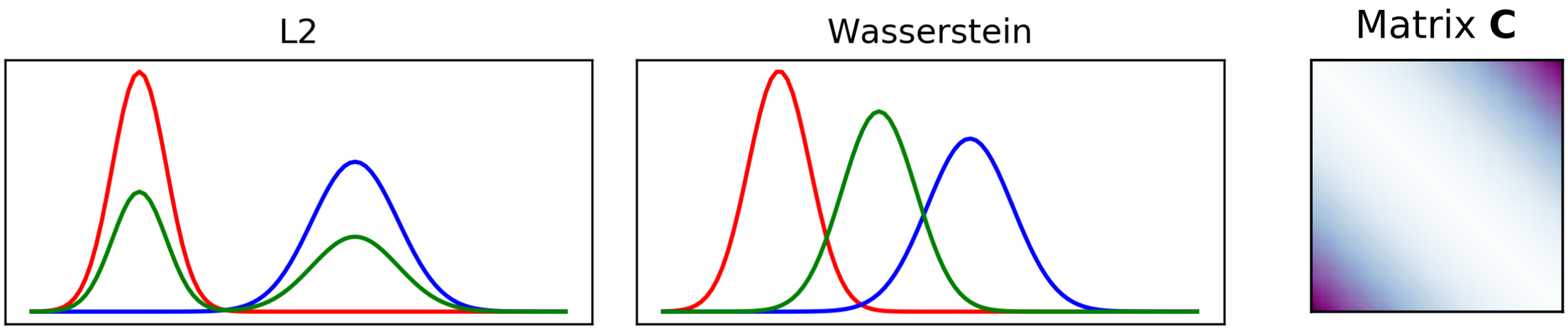


$\mu_{\tilde{X}^0}$

Matching words embedding

- Words are embedded in a high-dimensional space with neural networks
- Matching two documents is an OT problem, with the cost being the l_2 distance in the embedded space

Wasserstein barycenter

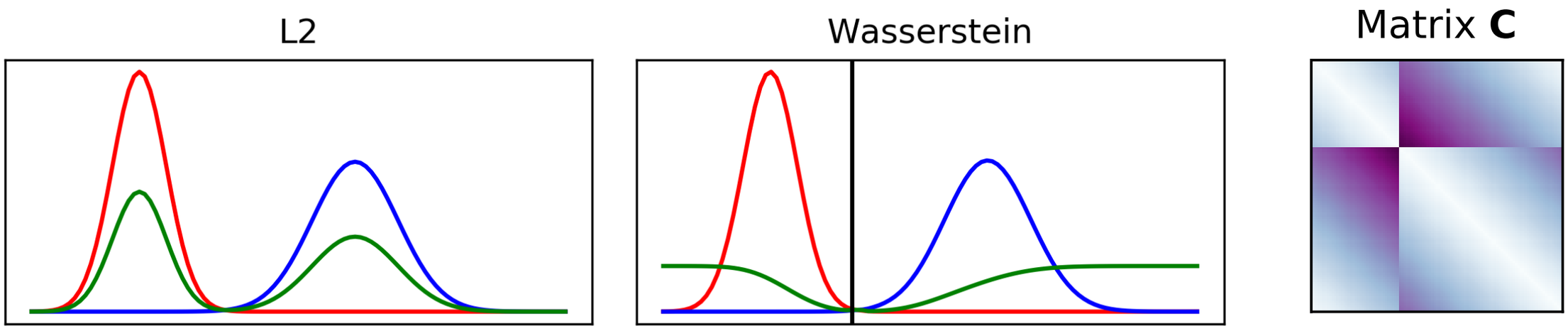


Barycenters [Agueh and Carlier, 2011]

$$\bar{\mu} = \arg\min_{\mu} \quad \sum_{i}^{n} \lambda_i W_p^p(\mu^i, \mu)$$

- $\lambda_i > 0$ and $\sum_i^n \lambda_i = 1$.
- Uniform barycenter has $\lambda_i = \frac{1}{n}, \forall i$.
- Interpolation with n=2 and $\boldsymbol{\lambda} = [1-t, t]$ with $0 \leq t \leq 1$ [McCann, 1997].
- Regularized barycenters using Bregman projections [Benamou et al., 2015].
- The cost and regularization impacts the interpolation trajectory.

Wasserstein barycenter



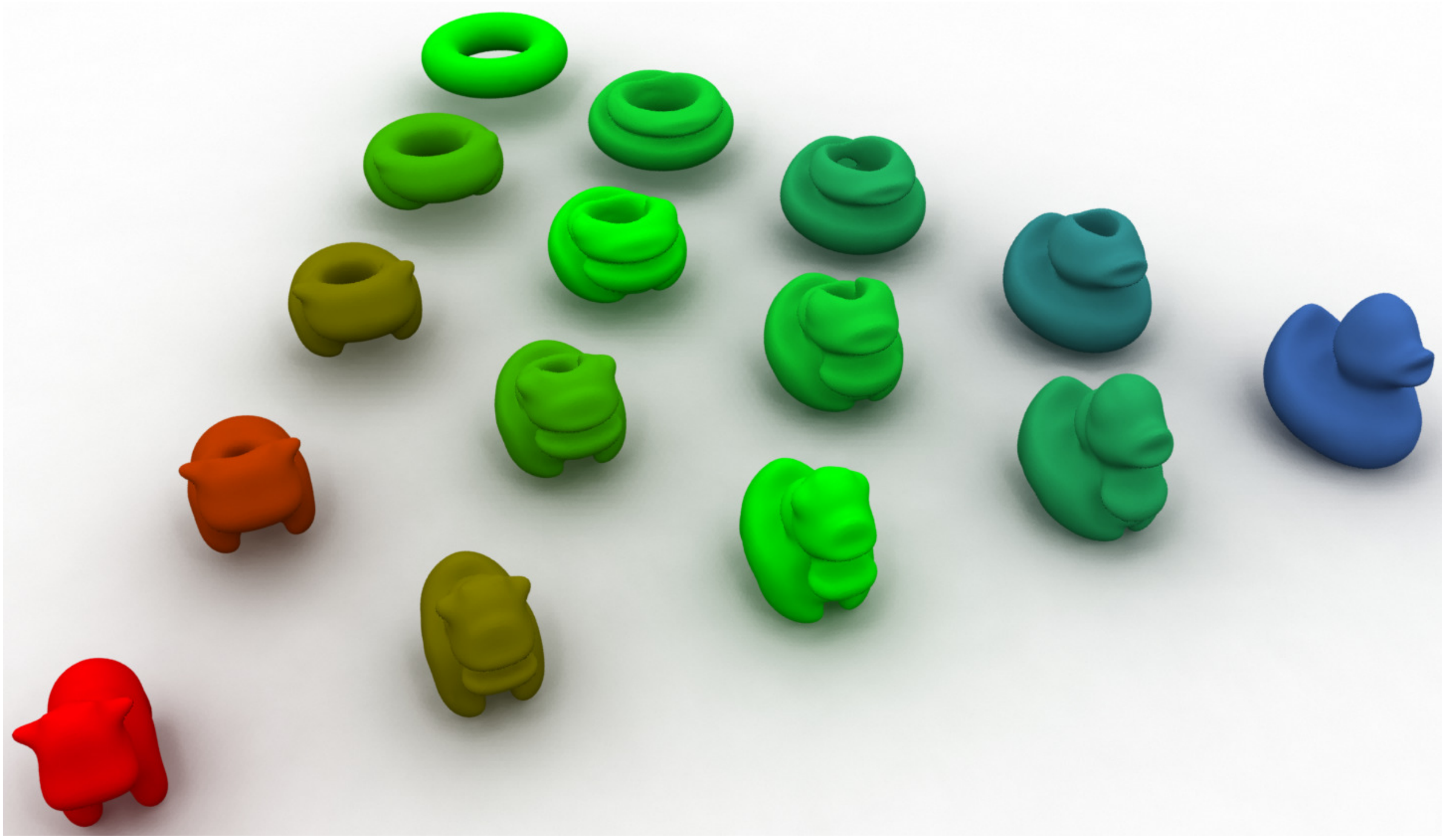
Barycenters [Agueh and Carlier, 2011]

$$\bar{\mu} = \arg\min_{\mu} \quad \sum_{i}^{n} \lambda_i W_p^p(\mu^i, \mu)$$

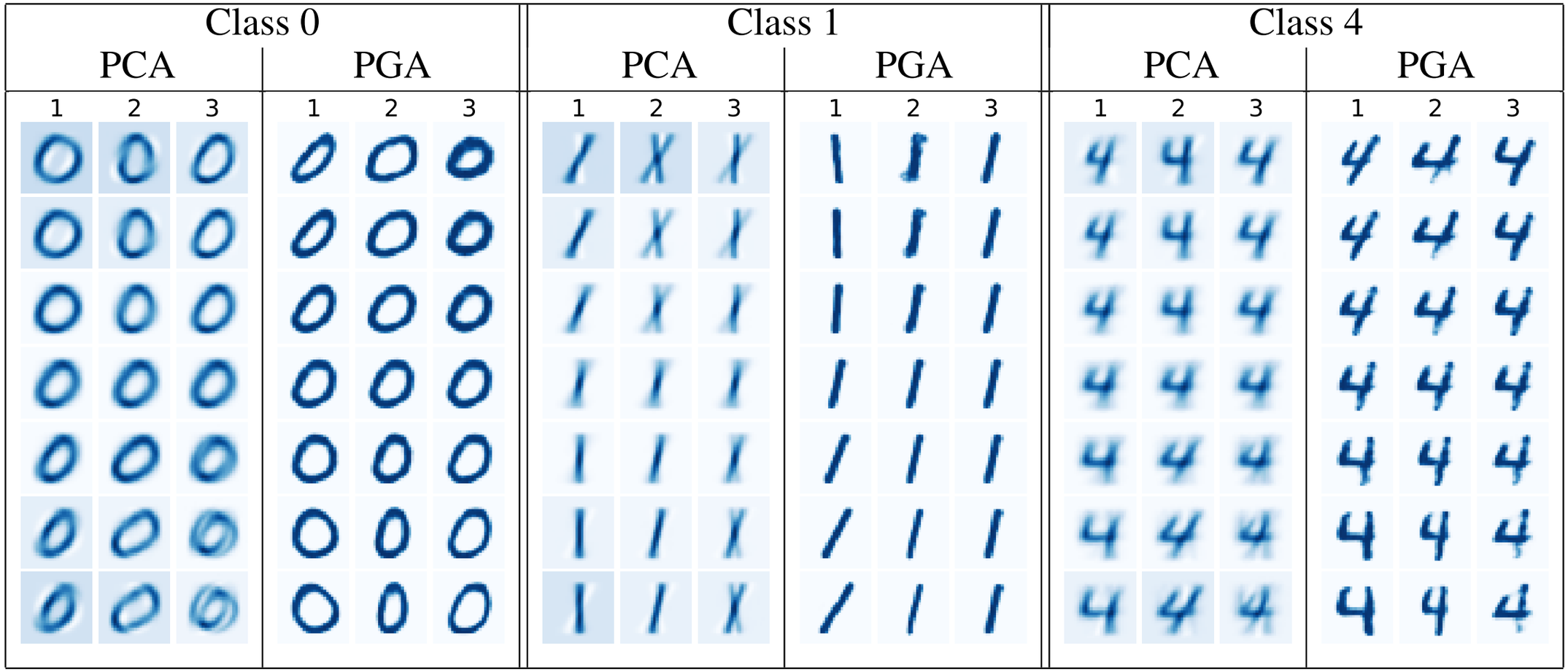
- $\lambda_i > 0$ and $\sum_i^n \lambda_i = 1$.
- Uniform barycenter has $\lambda_i = \frac{1}{n}, \forall i$.
- Interpolation with n=2 and $\boldsymbol{\lambda} = [1-t, t]$ with $0 \leq t \leq 1$ [McCann, 1997].
- Regularized barycenters using Bregman projections [Benamou et al., 2015].
- The cost and regularization impacts the interpolation trajectory.

3D Wasserstein barycenter

Shape interpolation [Solomon et al., 2015]



Principal Geodesics Analysis



Geodesic PCA in the Wasserstein space [Bigot et al., 2017]

- Generalization of Principal Component Analysis to the Wassertsein manifold.
- Regularized OT [Seguy and Cuturi, 2015].
- Approximation using Wasserstein embedding [Courty et al., 2017].
- Also note recent Wasserstein Dictionary Learning approaches [Schmitz et al., 2017].

Computational Aspects

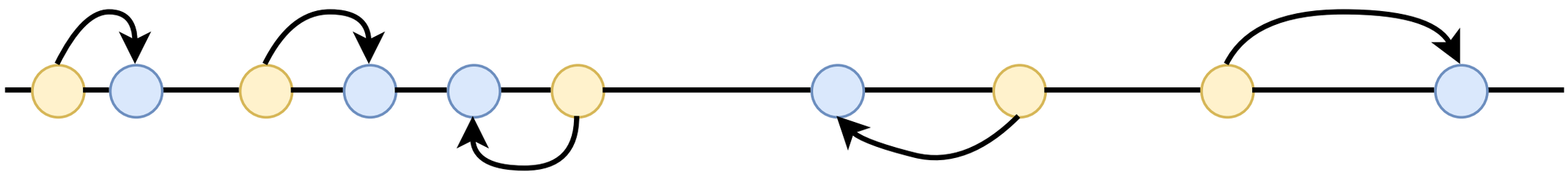
Is this problem tractable ?
- Closed form solutions
- Approximate solutions

Special case: 1D distribution

We consider the case where $c(x, y)$ is a strictly convex and increasing function of $|x - y|$.

- if $x_1 < x_2$ and $y_1 < y_2$, it is easy to check that $c(x_1, y_1) + c(x_2, y_2) < c(x_1, y_2) + c(x_2, y_1)$
- As such, any optimal transport plan respects the ordering of the elements, and the solution is given by the monotone rearrangement of μ_1 onto μ_2

This gives very simple algorithm to compute the transport in $O(N \log N)$, by sorting both x_i and y_i and summing the absolute values of differences.



Special case: 1D distribution

Consider the cumulative distribution functions F_μ associated to the μ distribution.

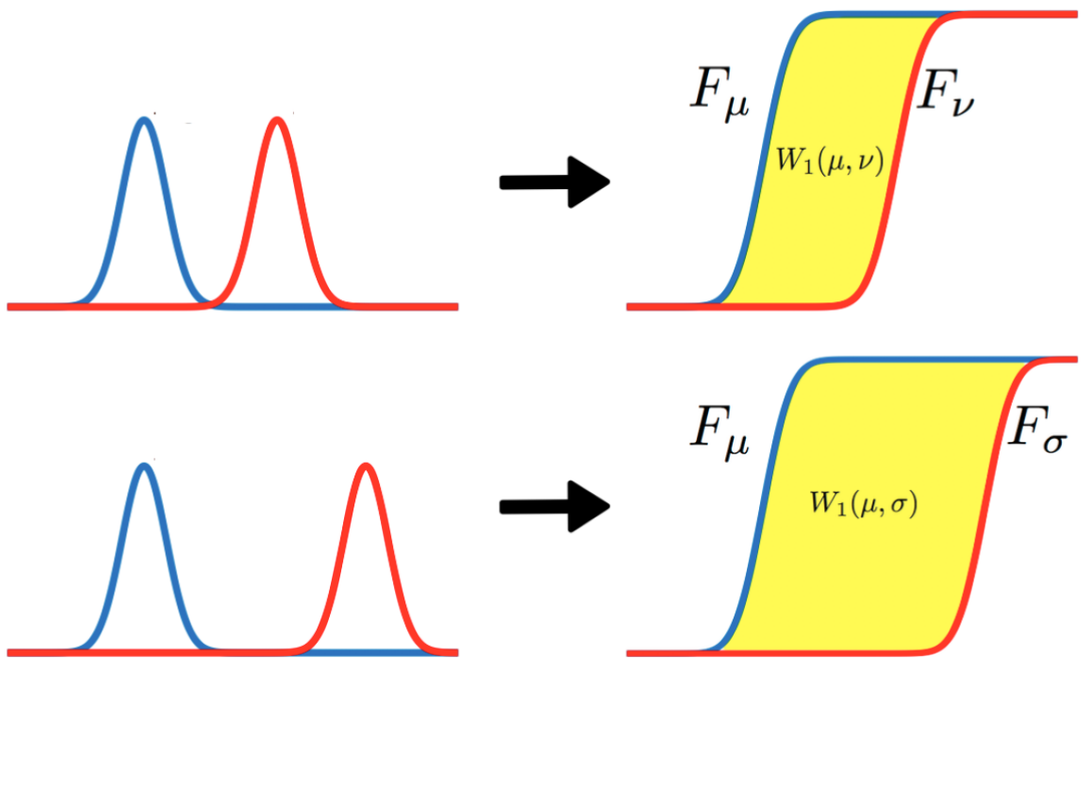
- It is defined such that $F_\mu(t) = \mu(-\infty, t]$.

We will note $F_\mu^{-1}(q),\ q \in [0, 1]$ the corresponding generalized inverse distribution (or quantile function)

- defined as $F_\mu^{-1}(q) = \inf\{x \in \mathbb{R} : F_\mu(x) \geq q\}$.

Then,

$$W_1(\mu_s, \mu_t) = \int_0^1 c(F_{\mu_s}^{-1}(q), F_{\mu_t}^{-1}(q))dq$$



Sliced-Wasserstein on $\mathbb{R}^d$

Wasserstein on $\mathbb{R}$:

$$\forall p \geq 1, \forall \mu, \nu \in \mathcal{P}_p(\mathbb{R}),\ W_p^p(\mu, \nu) = \int_0^1 |F_\mu^{-1}(u) - F_\nu^{-1}(u)|^p \ \mathrm{d}u \tag{4}$$

This property gives a method for computing Wasserstein in higher dimensions ($n > 1$).

The principle is simple. Slice the distribution along lines, project the measures onto it and compute $1D$ Wasserstein along those projections.

Sliced-Wasserstein [Rabin et al., 2011b]

Let $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$,

$$SW_p^p(\mu, \nu) = \int_{S^{d-1}} W_p^p(P_\#^\theta \mu, P_\#^\theta \nu) \ \mathrm{d}\lambda(\theta), \tag{5}$$

where $P^\theta(x) = \langle x, \theta \rangle$, λ uniform measure on S^{d-1}.

Properties:

- Distance
- Topogically equivalent to the Wasserstein distance
- Monte-Carlo approximation in $O(Ln \log n)$

Special cases: Wasserstein on the Circle

Let $\mu, \nu \in \mathcal{P}(S^1)$ where $S^1 = \mathbb{R}/\mathbb{Z}$.

- Parametrize S^1 by $[0, 1[$
- $\forall x, y \in [0, 1[,\ d_{S^1}(x, y) = \min(|x - y|, 1 - |x - y|)$
- For a cost function $c(x, y) = h(d_{S^1}(x, y))$ with $h : \mathbb{R} \to \mathbb{R}^+$ increasing and convex
- $\forall \mu, \nu \in \mathcal{P}(S^1)$, [Rabin et al., 2011a]

$$W_c(\mu, \nu) = \inf_{\alpha \in \mathbb{R}} \int_0^1 h(|F_\mu^{-1}(t) - (F_\nu - \alpha)^{-1}(t)|)\ \mathrm{d}t. \tag{6}$$

- To find α: binary search [Delon et al., 2010]

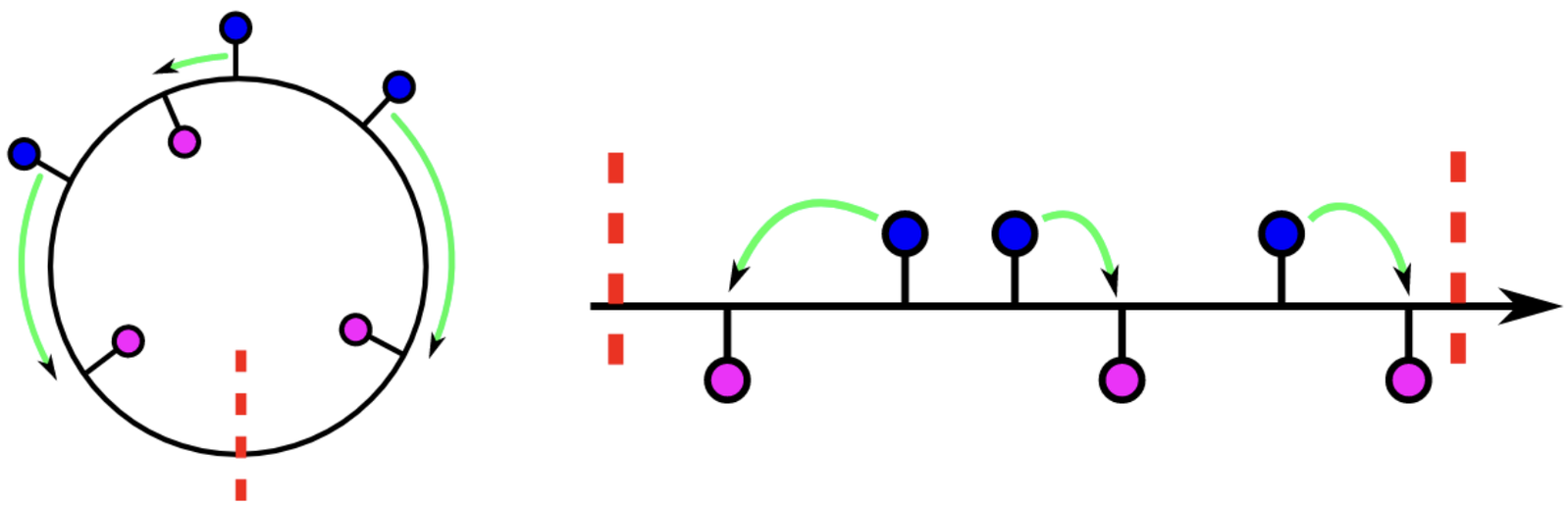


Image from [Rabin et al., 2011a]

Particular Cases

- For $h = \mathrm{Id}$, [Hundrieser et al., 2021]

$$W_1(\mu,\nu) = \int_0^1 |F_\mu(t) - F_\nu(t) - \mathrm{LevMed}(F_\mu - F_\nu)| \, \mathrm{d}t, \tag{7}$$

where

$$\mathrm{LevMed}(f) = \inf\left\{ t \in \mathbb{R},\ \beta(\{x \in [0,1[,\ f(x) \le t\}) \ge \frac{1}{2} \right\}. \tag{8}$$

- For $h(x) = x^2$ and $\nu = \mathrm{Unif}(S^1)$,[Bonet et al., 2022]

$$W_2^2(\mu,\nu) = \int_0^1 |F_\mu^{-1}(t) - t - \hat{\alpha}|^2 \, \mathrm{d}t \quad \text{with} \quad \hat{\alpha} = \int x \, \mathrm{d}\mu(x) - \frac{1}{2}. \tag{9}$$

In particular, if $x_1 < \cdots < x_n$ and $\mu_n = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$, then

$$W_2^2(\mu_n,\nu) = \frac{1}{n}\sum_{i=1}^n x_i^2 - \Big(\frac{1}{n}\sum_{i=1}^n x_i\Big)^2 + \frac{1}{n^2}\sum_{i=1}^n (n+1-2i)x_i + \frac{1}{12}. \tag{10}$$

Sliced-Wasserstein on the Sphere

The slicing strategy can be extended to manifolds ! In this case slices are geodesics of the considered manifold. Example on the Sphere [Bonet et al., 2022]:

- Great circle: Intersection between 2-plane and S^{d-1}
- Parametrize 2-plane by the Stiefel manifold

$$\mathbb{V}_{d,2} = \{U \in \mathbb{R}^{d\times 2},\ U^T U = I_2\}$$

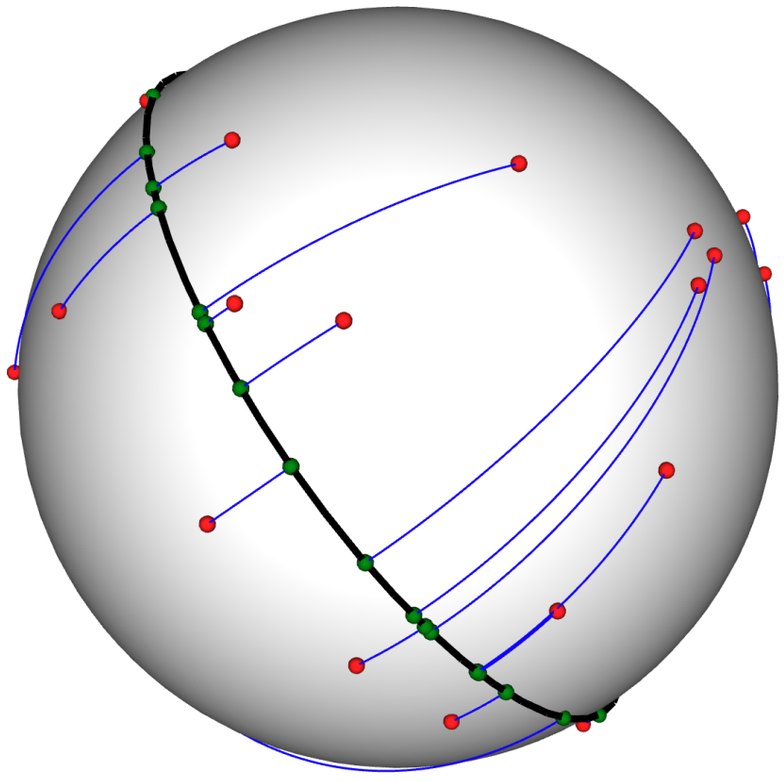
- Projection on great circle C: For a.e. $x \in S^{d-1}$,

$$P^C(x) = \operatorname*{argmin}_{y\in C}\ d_{S^{d-1}}(x,y),$$

where $d_{S^{d-1}}(x,y) = \arccos(\langle x, y\rangle)$.

- For $U \in \mathbb{V}_{d,2}$, $C = \operatorname{span}(UU^T) \cap S^{d-1}$,

$$P^U(x) = U^T \operatorname*{argmin}_{y\in C} d_{S^{d-1}}(x,y)$$
$$= \frac{U^T x}{\|U^T x\|_2}.$$



Special case: transport between Gaussians

In the case where $\mu_s \sim \mathcal{N}(\mathbf{m}_1, \Sigma_1)$ and $\mu_t \sim \mathcal{N}(\mathbf{m}_2, \Sigma_2)$ the Wasserstein distance with $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$ reduces to:

W_2^2 between Gaussians

$$W_2^2(\mu_s, \mu_t) = ||\mathbf{m}_1 - \mathbf{m}_2||_2^2 + \mathcal{B}(\Sigma_1, \Sigma_2)^2$$

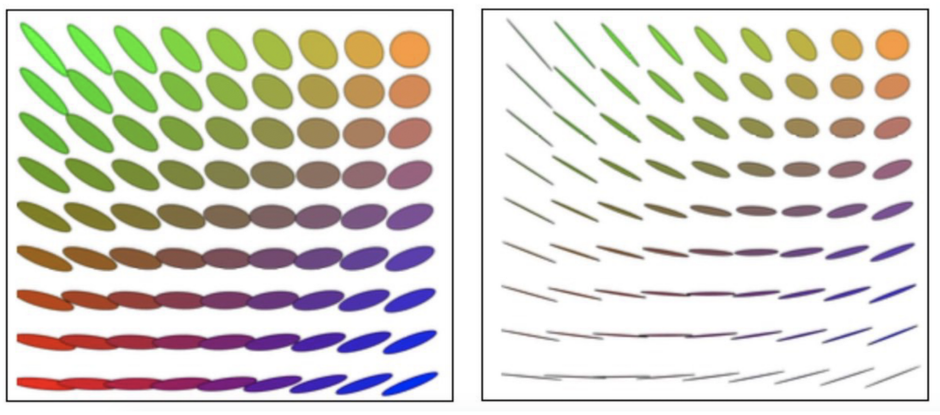
where $\mathbb{B}(,)$ is the so-called Bures metric:

$$\mathcal{B}(\Sigma_1, \Sigma_2)^2 = \text{trace}(\Sigma_1 + \Sigma_2 - 2(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}).$$

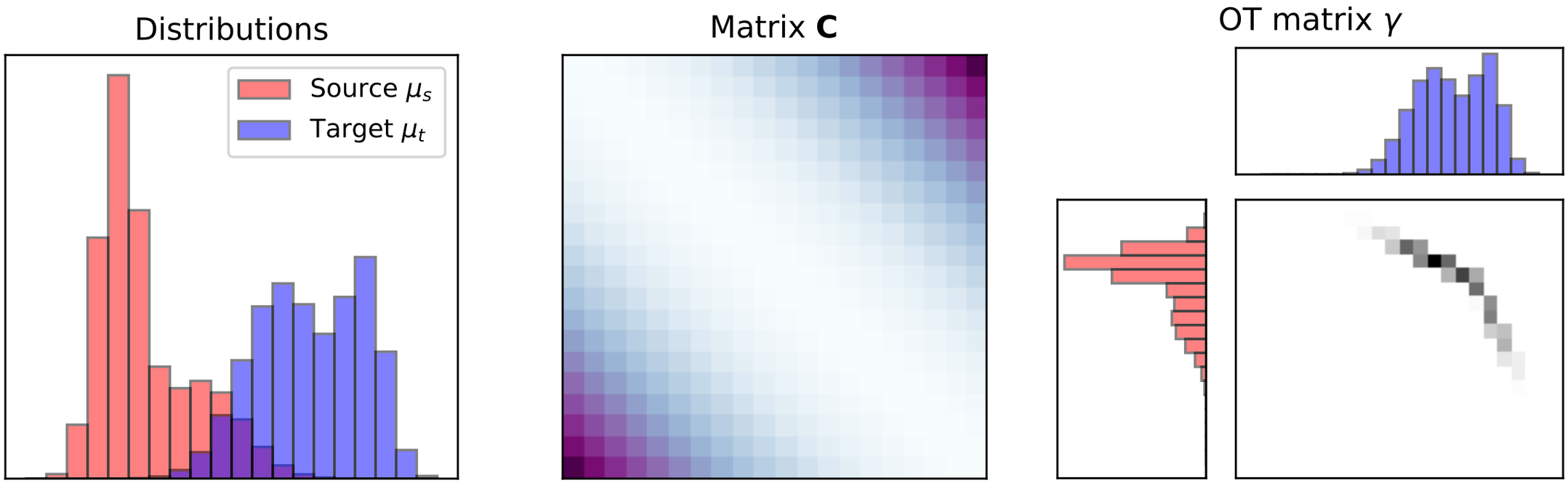
The optimal map T is given by

$$T(\mathbf{x}) = \mathbf{m}_2 + A(\mathbf{x} - \mathbf{m}_1)$$

with $A = \Sigma_1^{-1/2}(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2}\Sigma_1^{-1/2}$



Optimal transport with discrete distributions



OT Linear Program

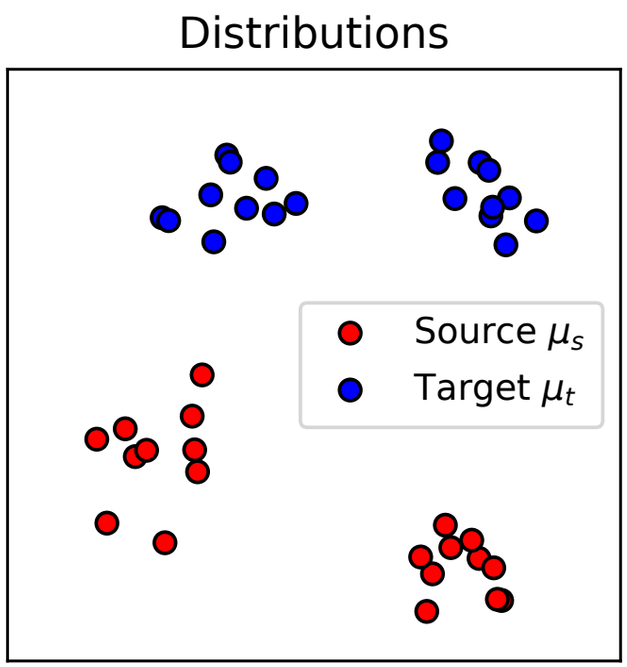
$$\boldsymbol{\pi}_0 = \underset{\boldsymbol{\pi} \in \mathbf{\Pi}}{\operatorname{argmin}} \quad \left\{ \langle \boldsymbol{\pi}, \mathbf{C} \rangle_F = \sum_{i,j} \pi_{i,j} c_{i,j} \right\}$$

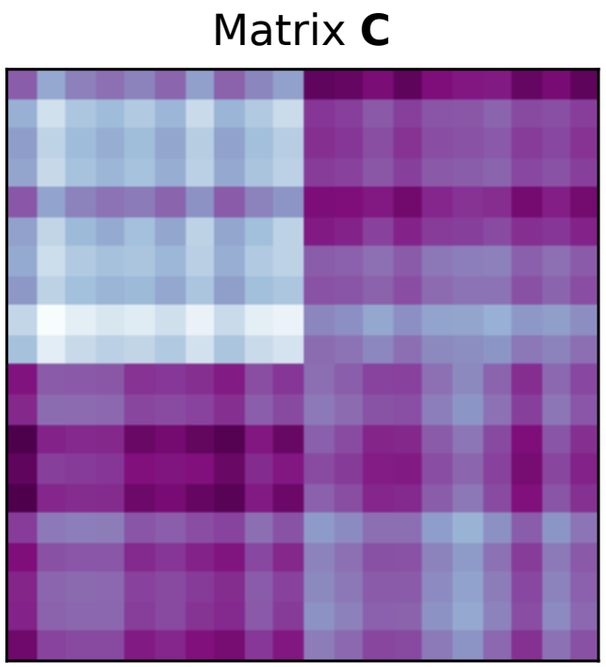
where $\mathbf{C}$ is a cost matrix with $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ and the marginals constraints are

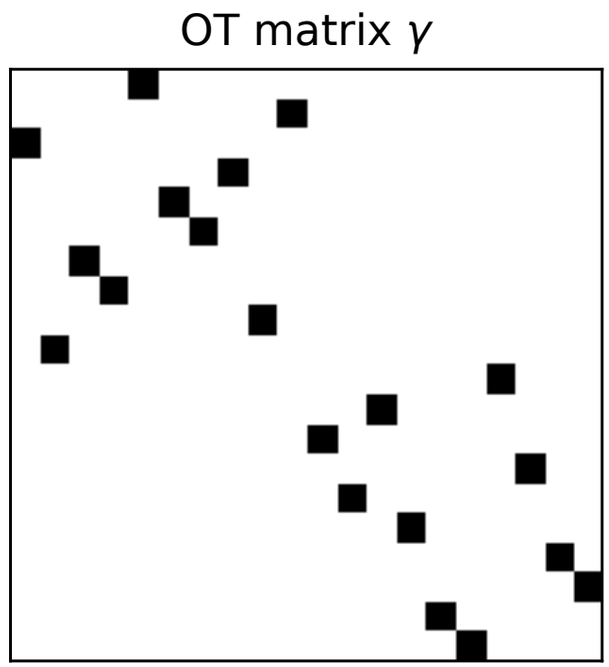
$$\mathbf{\Pi} = \left\{ \boldsymbol{\pi} \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} \,|\, \boldsymbol{\pi} \mathbf{1}_{\mathbf{n_t}} = \mu_{\mathbf{s}}, \boldsymbol{\pi}^{\mathbf{T}} \mathbf{1}_{\mathbf{n_s}} = \mu_{\mathbf{t}} \right\}$$

Solved with Network Flow solver of complexity $O(n^3 \log(n))$.

Optimal transport with discrete distributions







OT Linear Program

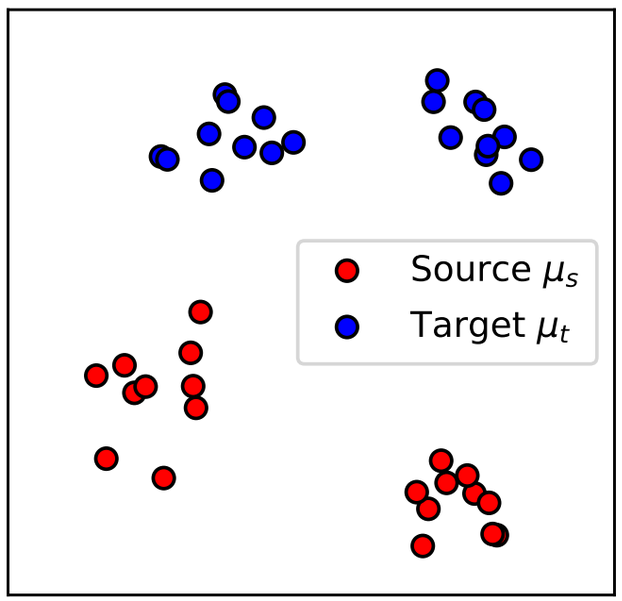
$$\boldsymbol{\pi}_0 = \underset{\boldsymbol{\pi} \in \mathbf{\Pi}}{\operatorname{argmin}} \quad \left\{ \langle \boldsymbol{\pi}, \mathbf{C} \rangle_F = \sum_{i,j} \pi_{i,j} c_{i,j} \right\}$$

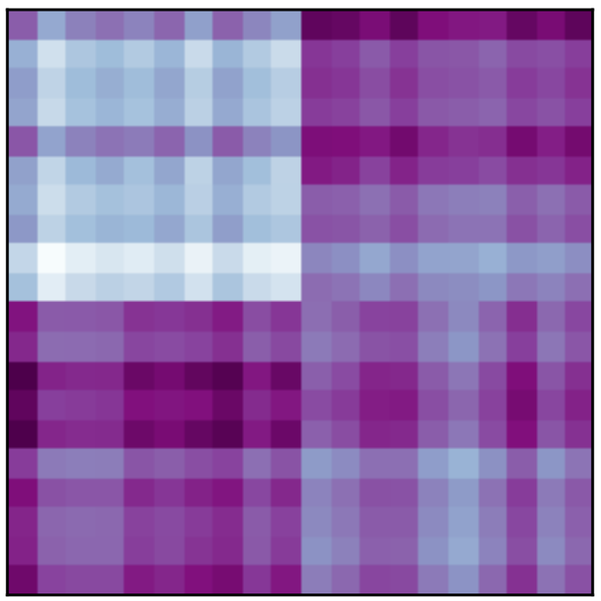
where $\mathbf{C}$ is a cost matrix with $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ and the marginals constraints are

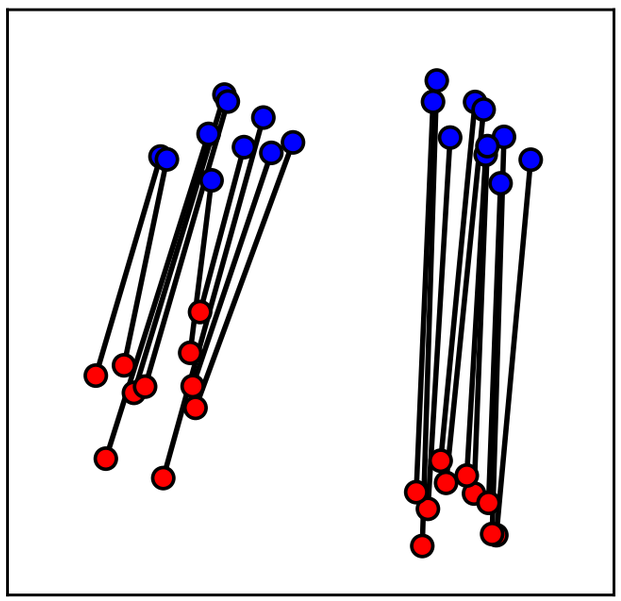
$$\mathbf{\Pi} = \left\{ \boldsymbol{\pi} \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} \,|\, \boldsymbol{\pi} \mathbf{1}_{\mathbf{n_t}} = \mu_{\mathbf{s}}, \boldsymbol{\pi}^{\mathbf{T}} \mathbf{1}_{\mathbf{n_s}} = \mu_{\mathbf{t}} \right\}$$

Solved with Network Flow solver of complexity $O(n^3 \log(n))$.

Optimal transport with discrete distributions







OT Linear Program

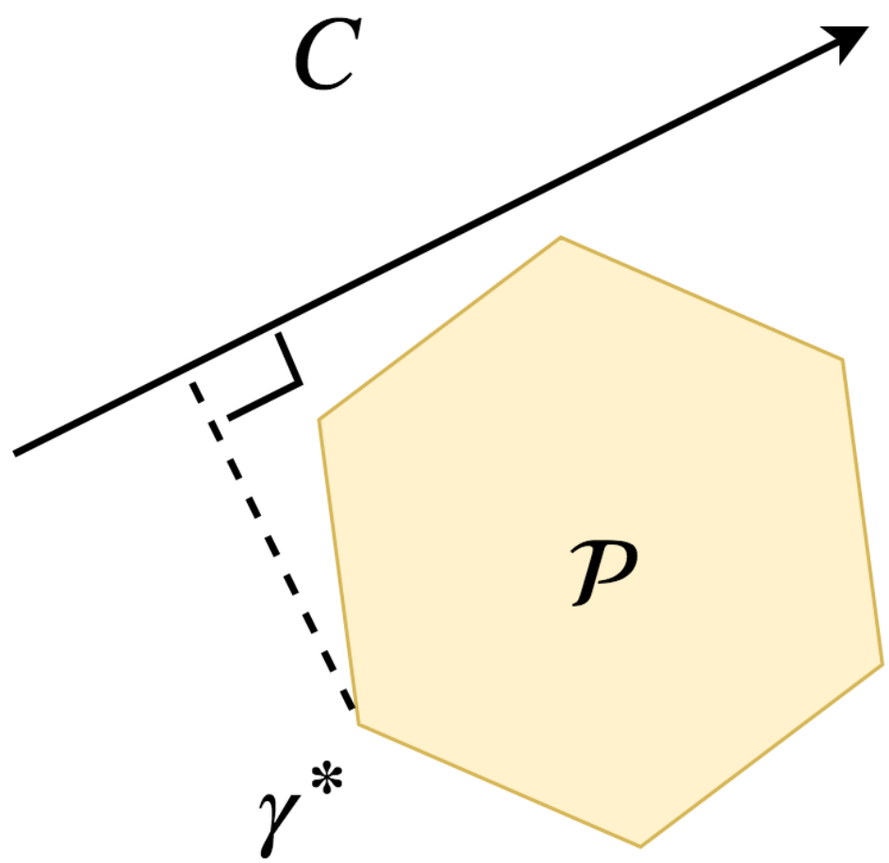
$$\boldsymbol{\pi}_0 = \underset{\boldsymbol{\pi} \in \mathbf{\Pi}}{\operatorname{argmin}} \quad \left\{ \langle \boldsymbol{\pi}, \mathbf{C} \rangle_F = \sum_{i,j} \pi_{i,j} c_{i,j} \right\}$$

where $\mathbf{C}$ is a cost matrix with $c_{i,j} = c(\mathbf{x}_i^s, \mathbf{x}_j^t)$ and the marginals constraints are

$$\mathbf{\Pi} = \left\{ \boldsymbol{\pi} \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} \,|\, \boldsymbol{\pi} \mathbf{1}_{\mathbf{n_t}} = \mu_{\mathbf{s}}, \boldsymbol{\pi}^{\mathbf{T}} \mathbf{1}_{\mathbf{n_s}} = \mu_{\mathbf{t}} \right\}$$

Solved with Network Flow solver of complexity $O(n^3 \log(n))$.

Optimal transport with discrete distributions



- $\mathcal{P}$ is the Birkhoff polytope
- No unique solution in some cases, numerical instabilities
- Not differentiable !

Regularized optimal transport

$$\boldsymbol{\pi}_0^{\lambda} = \underset{\boldsymbol{\pi} \in \boldsymbol{\Pi}}{\operatorname{argmin}} \quad \langle \boldsymbol{\pi}, \mathbf{C} \rangle_F + \lambda \Omega(\boldsymbol{\pi}), \qquad (11)$$

Regularization term $\Omega(\boldsymbol{\pi})$

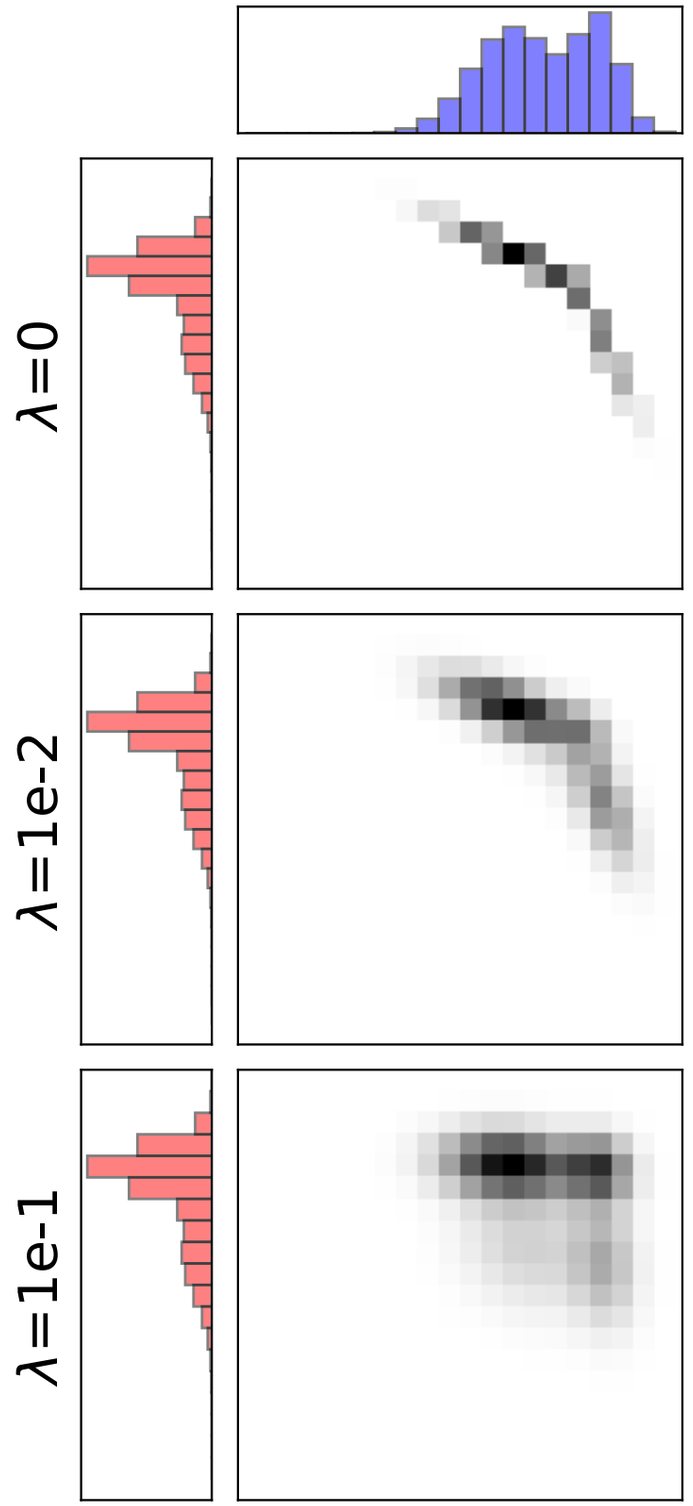
- Entropic regularization [Cuturi, 2013].
- Group Lasso [Courty et al., 2016].
- KL, Itakura Saito, β-divergences, [Dessein et al., 2016].

Why regularize?

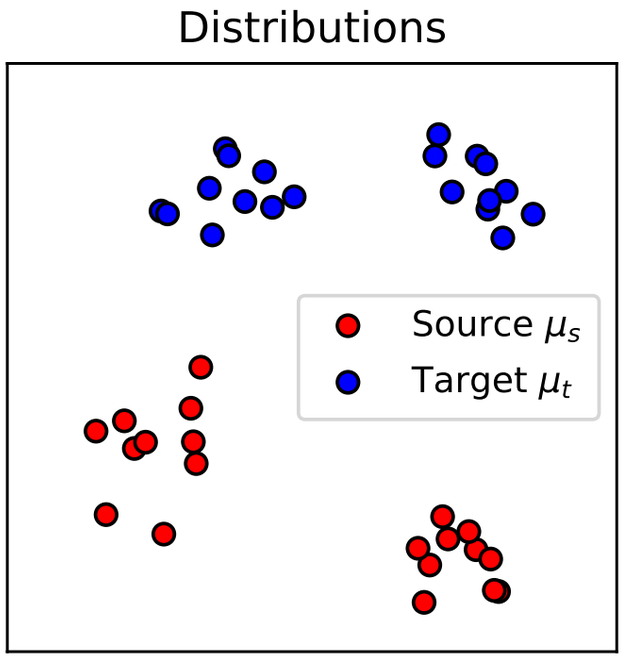
- Smooth the "distance" estimation:

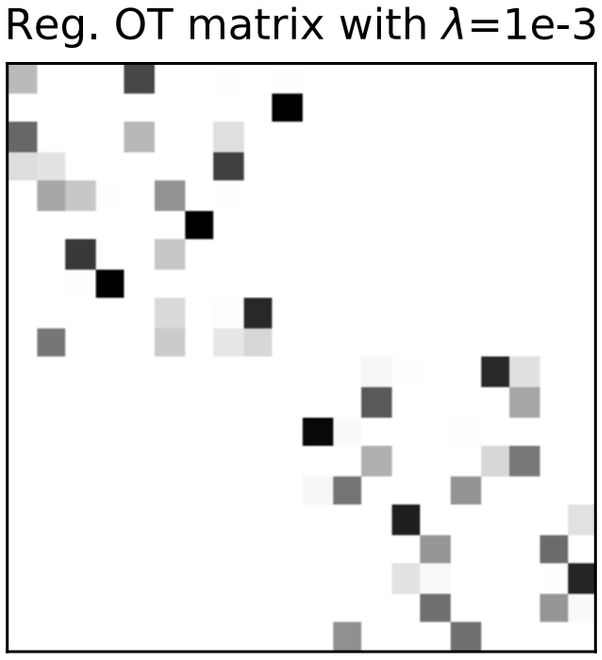
$$W_\lambda(\mu_s, \mu_t) = \left\langle \boldsymbol{\pi}_0^{\lambda}, \mathbf{C} \right\rangle_F$$

- Encode prior knowledge on the data.
- Better posed problem (convex, stability).
- Fast algorithms to solve the OT problem.



Entropic regularized optimal transport



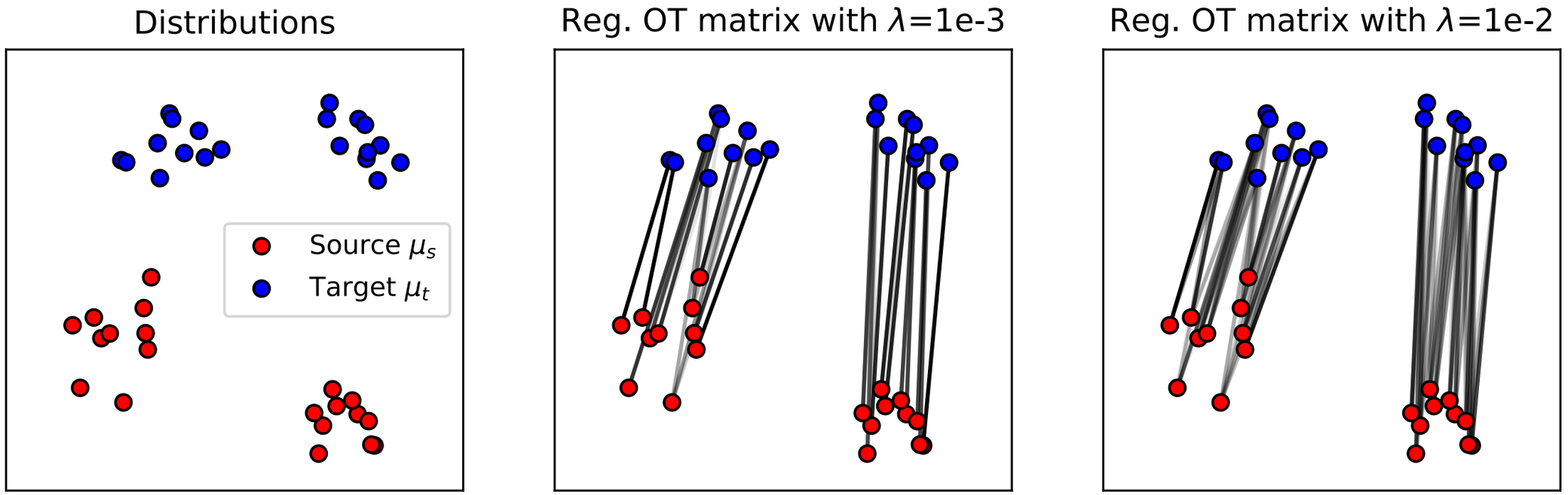


Entropic regularization [Cuturi, 2013]

$$\Omega(\boldsymbol{\pi}) = \sum_{i,j} \boldsymbol{\pi}(i,j)(\log \boldsymbol{\pi}(i,j) - 1)$$

- Regularization with the negative entropy of $\boldsymbol{\pi}$.

Entropic regularized optimal transport



Entropic regularization [Cuturi, 2013]

$$\Omega(\boldsymbol{\pi}) = \sum_{i,j} \boldsymbol{\pi}(i,j)(\log \boldsymbol{\pi}(i,j) - 1)$$

- Regularization with the negative entropy of $\boldsymbol{\pi}$.

Resolving the entropy regularized problem

Entropy-regularized transport

The solution of entropy regularized optimal transport problem is of the form

$$\boldsymbol{\pi}_0^{\lambda} = \text{diag}(\mathbf{u}) \exp(-\mathbf{C}/\lambda) \text{diag}(\mathbf{v})$$

Why ? Consider the Lagrangian of the optimization problem:

$$\mathcal{L}(\boldsymbol{\pi}, \alpha, \beta) = \sum_{ij} \boldsymbol{\pi}_{ij} \mathbf{C}_{ij} + \lambda \boldsymbol{\pi}_{ij} (\log \boldsymbol{\pi}_{ij} - 1) + \alpha^{\mathbf{T}} (\boldsymbol{\pi} \mathbf{1}_{n_t} - \mu_s) + \beta^{\mathbf{T}} (\boldsymbol{\pi}^T \mathbf{1}_{n_s} - \mu_t)$$

$$\begin{aligned} \partial \mathcal{L}(\boldsymbol{\pi}, \alpha, \beta)/\partial \boldsymbol{\pi}_{ij} &= \mathbf{C}_{ij} + \lambda \log \boldsymbol{\pi}_{ij} + \alpha_i + \beta_j \\ \partial \mathcal{L}(\boldsymbol{\pi}, \alpha, \beta)/\partial \boldsymbol{\pi}_{ij} = 0 &\implies \boldsymbol{\pi}_{ij} = \exp(\frac{\alpha_i}{\lambda}) \exp(-\frac{\mathbf{C}_{ij}}{\lambda}) \exp(\frac{\beta_j}{\lambda}) \end{aligned}$$

- Through the **Sinkhorn theorem** $\text{diag}(\mathbf{u})$ and $\text{diag}(\mathbf{v})$ exist and are unique.
- Can be solved by the **Sinkhorn-Knopp** algorithm (implementation in parallel, GPU).

Sinkhorn-Knopp algorithm

The Sinkhorn-Knopp algorithm performs alternatively a scaling along the rows and columns of $\mathbf{K} = \exp(-\frac{\mathbf{C}}{\lambda})$ to match the desired marginals.

Algorithm 1 Sinkhorn-Knopp Algorithm (SK).

Require: $\mathbf{a}, \mathbf{b}, \mathbf{C}, \lambda$

$\mathbf{u}^{(0)} = \mathbf{1}, \mathbf{K} = \exp(-\mathbf{C}/\lambda)$

for i in $1, \dots, n_{it}$ **do**

$\mathbf{v}^{(i)} = \mathbf{b} \oslash \mathbf{K}^\top \mathbf{u}^{(i-1)}$ // Update right scaling

$\mathbf{u}^{(i)} = \mathbf{a} \oslash \mathbf{K}\mathbf{v}^{(i)}$ // Update left scaling

end for

return $\mathbf{T} = \text{diag}(\mathbf{u}^{(n_{it})})\mathbf{K}\text{diag}(\mathbf{v}^{(n_{it})})$

- Complexity $O(kn^2)$, where k iterations are required to reach convergence
- Fast implementation in parallel, GPU friendly
- Convolutive/Heat structure for K [Solomon et al., 2015]

Sinkhorn as Bregman projections

Recalling that the Kullback Leibler (KL) divergence between two distribution is

$$\mathrm{KL}(\boldsymbol{\pi}, \rho) = \sum_{ij} \boldsymbol{\pi}_{ij} \log \frac{\boldsymbol{\pi}_{ij}}{\rho_{ij}} =< \boldsymbol{\pi}, \log \frac{\boldsymbol{\pi}}{\rho} >_F,$$

Benamou *et al.* [Benamou et al., 2015] showed that solving for the OT problem is actually a Bregman projection

OT as a Bregman projection

$\boldsymbol{\pi}^\star$ is the solution of the following Bregman projection

$$\boldsymbol{\pi}^\star = \operatorname*{argmin}_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \mathrm{KL}(\boldsymbol{\pi}, \zeta), \tag{12}$$

where $\zeta = \exp(-\frac{C}{\lambda})$.

- Sinkhorn in this case is an iterative projection scheme, with alternative projections on marginal constraints.
- Generalizes well for barycenters computation

Dual formulation of optimal transport

- Yet, solving for π is impractical to intractable when dealing with high-dimensional distributions
- especially if one is interested in computing the gradients of the Wasserstein distance
- Other solving strategies should be taken into consideration
- Recalling that any LP problem can be turnt into its dual form:

primal form :				**dual form :**			
minimize	z	$=$	$\mathbf{c}^T\mathbf{x}$,	maximize	$\tilde{z}$	$=$	$\mathbf{b}^T\mathbf{y}$,
so that	$\mathbf{Ax}$	$=$	$\mathbf{b}$	so that	$\mathbf{A}^T\mathbf{y}$	$\leq$	$\mathbf{c}$
and	$\mathbf{x}$	$\geq$	$\mathbf{0}$				

- Weak duality: $\tilde{z}$ is a lower bound of z, Strong duality $\tilde{z} = z$
- Strong duality is usually achieved via Farkas Theorem

Duality: general case with continuous distributions

We now introduce two functions scalar functions ϕ and ψ (also known as Kantorovich potentials) that will act as our dual variables. Then, we consider the optimal problem is equivalent (by the Rockafellar-Fenchel theorem) to:

$$\max_{\phi,\psi} \left\{ \int \phi d\mu_s + \int \psi d\mu_t \;\; | \;\; \phi(x) + \psi(y) \leq c(x,y) \right\} \tag{13}$$

Note that the marginal constraint has been turned into an equality constraint on ϕ and ψ

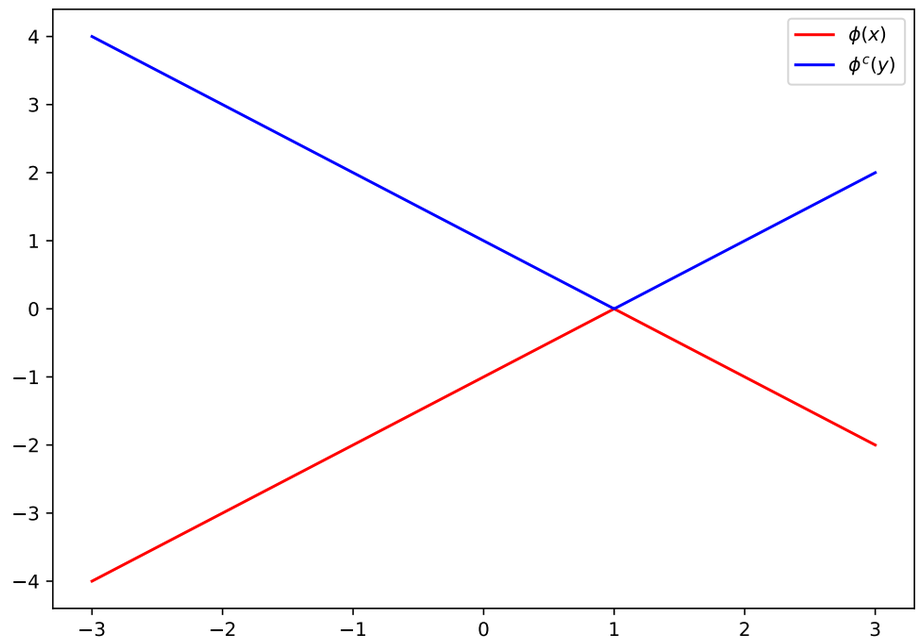
Introducing the *c-transform* (or *c-conjugate*) H^c which is in spirit close to a Legendre transform:

$$\phi^c \stackrel{\text{def}}{=} H^c(\phi) = \inf_x c(x,y) - \phi(x) \tag{14}$$

then the following problem is equivalent:

$$\max_{\phi} \left\{ \int \phi d\mu_s + \int \phi^c d\mu_t \;\; | \;\; \phi(x) + \phi^c(y) \leq c(x,y) \right\} \tag{15}$$

Case $c(x, y) = |x - y|$ (a.k.a W_1^1)



Whenever $c(x, y) = |x - y|$, then:

- existence of a solution but not unique
- For any $\phi \in \text{Lip}^1$ (set of 1-Lipschitz functions), we have $\phi^c(x) = -\phi(x)$

The optimal transport problem then amounts to find $\phi \in \text{Lip}^1$ as

$$\sup_{\phi \in \text{Lip}^1} \int \phi d(\mu_s - \mu_t) = \sup_{\phi \in \text{Lip}^1} \mathop{\mathbb{E}}_{\mathbf{x} \sim \mu_s} [\phi(x)] - \mathop{\mathbb{E}}_{\mathbf{y} \sim \mu_t} [\phi(y)] \tag{16}$$

- also known as **Kantorovich-Rubinstein duality**
- ϕ can be learnt as a neural network constrained to the set Lip^1, see next section on GAN

Case $c(x, y) = |x - y|^2/2$ (a.k.a W_2^2)

Whenever the cost is quadratic, $c(x, y) = |x - y|^2/2$, then:

- $T(x)$ the transport mapping exists and is unique
- More remarkably, it is a gradient of a convex functions $\Phi(x)$

$$T(x) = x - \nabla\phi(x) = \nabla(\frac{x^2}{2} - \phi(x)) = \nabla(\Phi(x)) \tag{17}$$

Brenier's Theorem

Let $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$, μ absolutely continuous with respect to the Lebesgue measure. Then, the optimal coupling π^* is unique and of the form $\pi^* = (Id, \nabla\Phi)_{\#}\mu$ with Φ a convex function.

- can be optimized for instance with Input Convex Neural Networks (ICNN) [Amos et al., 2017] to model the convex functions

Dual: empirical version

In the case when we have access to discrete distributions, μ_s (resp. μ_t) is characterized by a set of locations $\mathbf{X^s}$ and masses $\mathbf{a} \in \mathbb{R}^{n^s}$ (resp. $\mathbf{X^t}$ and $\mathbf{b} \in \mathbb{R}^{n^t}$)

Discrete dual version of OT

$$W(\mu_s, \mu_t) = \max_{\alpha \in \mathbb{R}^{n^s}, \beta \in \mathbb{R}^{n^t}, \alpha_i + \beta_j \leq c(\mathbf{x_i^s}, \mathbf{x_j^t})} \alpha^T \mathbf{a} + \beta^T \mathbf{b} \tag{18}$$

i.e. find a scalar values per sample

Regularized case

Adding regularization to the original problem turns the dual computation to an unconstrained problem !

In the case of entropy regularization, *i.e.*

$$W_\lambda(\mu_s, \mu_t) = \min_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \quad \langle \boldsymbol{\pi}, \mathbf{C} \rangle_F + \lambda \Omega(\boldsymbol{\pi}) \text{ with } \Omega(\boldsymbol{\pi}) = \sum_{i,j} \boldsymbol{\pi}(i,j) \log \boldsymbol{\pi}(i,j),$$

the dual now reads (in a discrete settings, measures are collections of Diracs):

$$\max_{\alpha, \beta} \alpha^T \mu_s + \beta^T \mu_t - \frac{1}{\lambda} \exp(\frac{\alpha}{\lambda})^T \mathbf{K} \exp(\frac{\beta}{\lambda}) \tag{19}$$

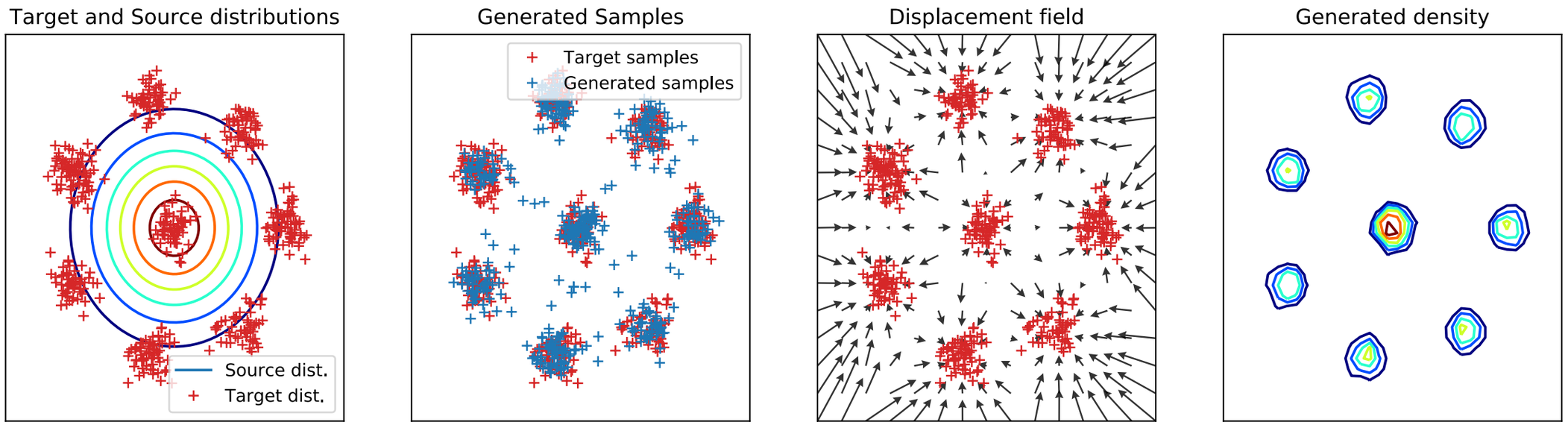
with $\mathbf{K} = \exp(-\frac{\mathbf{C}}{\lambda})$.

Remark: The Sinkhorn algorithm is a gradient ascent on the dual variables !

Regularized case

With this unconstrained problem, incremental gradients techniques (SGD, SAG) can be used to solve the problem !

- [Genevay et al., 2016] used the semi-dual formulation (one variable is removed by replacing it with its c-transform) int the first stochastic version of Optimal Transport problem
- [Seguy et al., 2017] used the full dual version with entropic and L2 regularizations, together with neural networks to parameterize the problem.



2 ways of minimizing the Wasserstein distance

In machine learning applications, one can be interested in finding distributions that minimize the Wasserstein distance wrt. a reference measure. There are two ways of understanding this:

- case 1: **for a fixed support** $\mathbf{X}$, find the corresponding probability masses $\mathbf{m}$
- case 2: **for a fixed vector of probability masses** $\mathbf{m}$, e.g. uniform distribution, find the corresponding support $\mathbf{X}$

Case 1: fixed support

Recalling the form of the dual

$$W(\mu, \mu_t) = \max_{\alpha \in \mathbb{R}^{n^s}, \beta \in \mathbb{R}^{n^t}, \alpha_i + \beta_j \leq c(\mathbf{x}, \mathbf{x_j^t})} \alpha^T \mathbf{m} + \beta^T \mathbf{b} \tag{20}$$

- $W(\mu, \mu_t)$ is convex wrt. $\mathbf{m}$
- $\partial_{\mathbf{m}} W(\mu, \mu_t) = \alpha^*$
- **Entropy regularized case**: $W_\lambda(\mu, \mu_t)$ is convex and $\nabla_{\mathbf{m}} W_\lambda(\mu, \mu_t) = \lambda \log \mathbf{u}$

Case 2: fixed probability masses $\mathbf{m}$

Recalling the form of the primal problem

$$W_2^2(\mu, \mu_t) = \min_{\boldsymbol{\pi} \in \boldsymbol{\Pi}} \quad < \boldsymbol{\pi}, \mathbf{1_{n^s}} \mathbf{1_{n^t}^T} \mathbf{X}^2 + \mathbf{X^t}^{2T} \mathbf{1_{n^t}} \mathbf{1_{n^s}} - 2\mathbf{X}\mathbf{X^t} > \tag{21}$$

- $W_2^2(\mu, \mu_t)$ decreases if $\mathbf{X} \leftarrow \mathbf{X^t} \boldsymbol{\pi}^{*T} \text{diag}(\mathbf{m}^{-1})$
- explicit gradient for the regularized case.
- Barycentric interpolation !

General case: autodifferentiation

Automatic differentiation to the rescue !

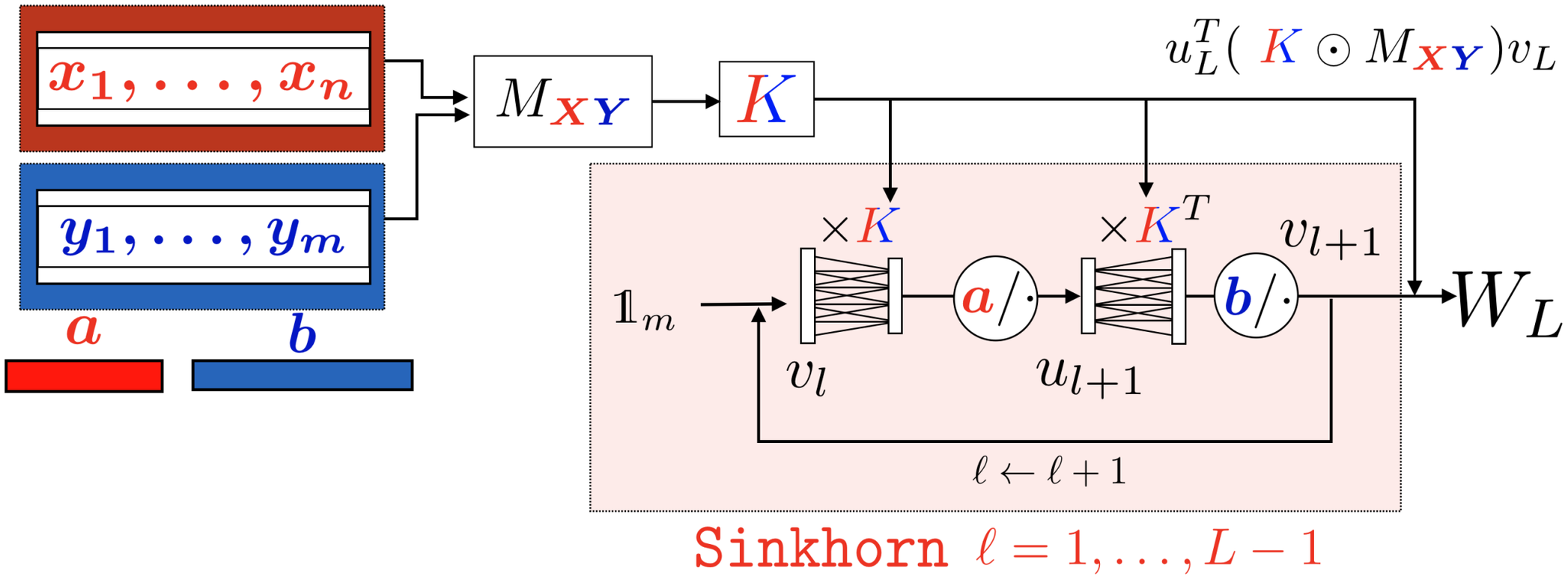


Image from Marco Cuturi

But also consider using the Enveloppe theorem, *i.e.* take the gradient in the optimal solution of the primal problem.

Variants of OT

Incomparable spaces

...to Gromov-Wasserstein

What if ?

Data are in Incomparable spaces

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y}) \text{ with } \mathcal{X}, \mathcal{Y} \nsubseteq \Omega$$

A cost function ?????

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

⇨ Not straightforward to find a suitable cost (e.g. no distance available)

...to Gromov-Wasserstein

What if ?

Data are in Incomparable spaces

Two probability distributions

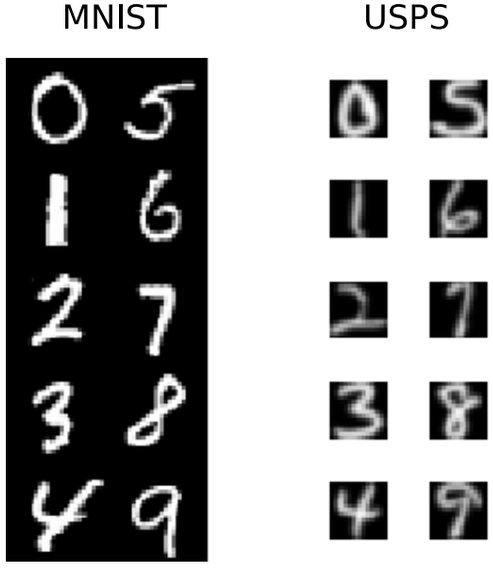
$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y}) \text{ with } \mathcal{X}, \mathcal{Y} \nsubseteq \Omega$$

A cost function ?????

$$c(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

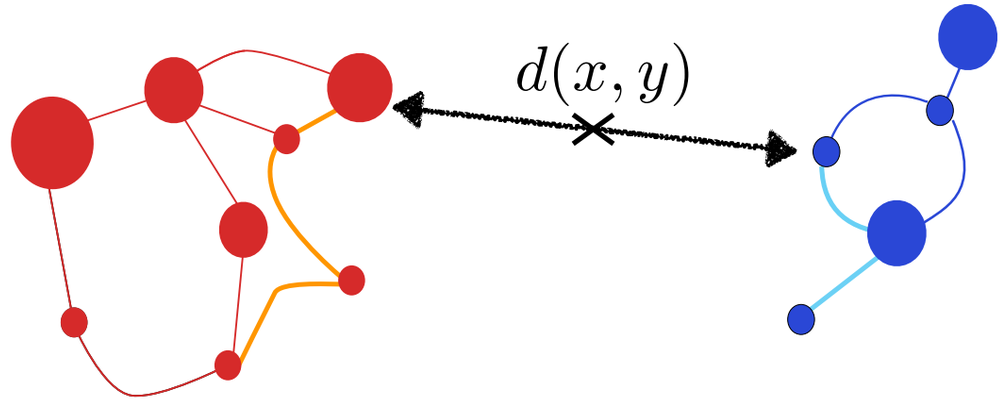
⇨ Not straightforward to find a suitable cost (e.g. no distance available)

Different Euclidean spaces



Example: $\mathcal{X} = \mathbb{R}^{28*28}, \mathcal{Y} = \mathbb{R}^{16*16}$

No notion of distance 2 nodes of different graphs



Example: $\mathcal{X} = \text{Graph } 1, \mathcal{Y} = \text{Graph } 2$

...to Gromov-Wasserstein

Gromov-Wasserstein distance

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$$

Two « intra-domain » costs

$$c_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$
$$c_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

…to Gromov-Wasserstein

Gromov-Wasserstein distance

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$$

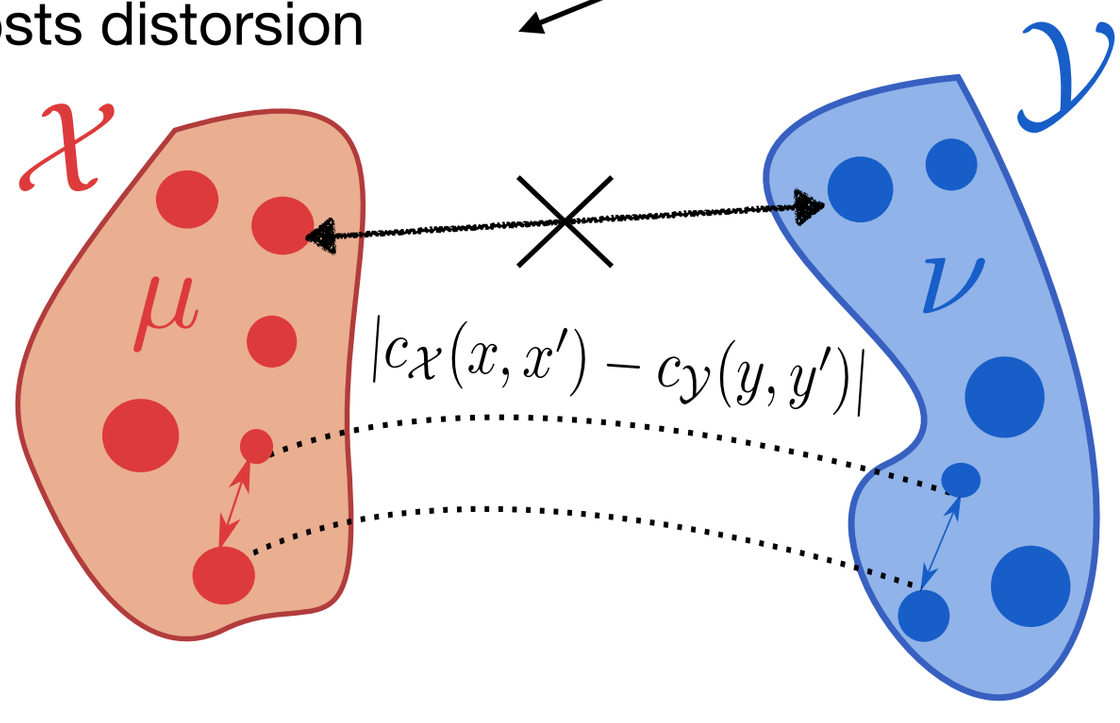
Two « intra-domain » costs

$$c_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$
$$c_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

Measure the costs distorsion



...to Gromov-Wasserstein

Gromov-Wasserstein distance

Two probability distributions

$$\mu \in \mathcal{P}(\mathcal{X}), \nu \in \mathcal{P}(\mathcal{Y})$$

Two « intra-domain » costs

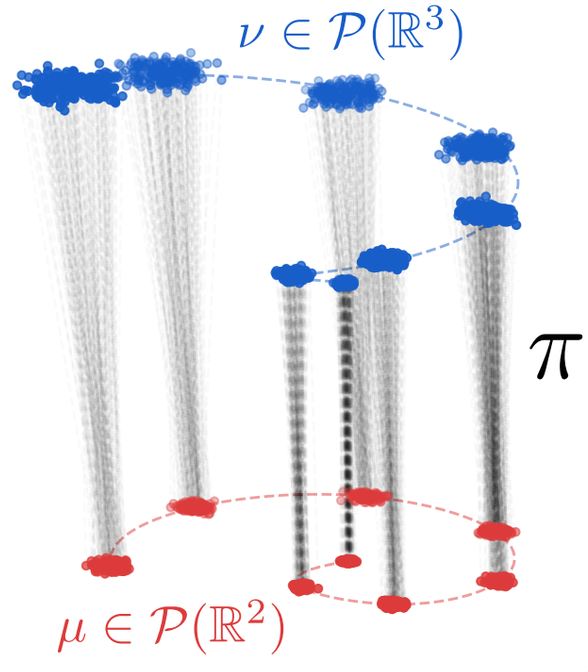
$$c_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$
$$c_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$$

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

The transportation problem is not linear anymore but **quadratic**

Associate pair of points with similar costs in each space



...to Gromov-Wasserstein

Gromov-Wasserstein distance


Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

A distance w.r.t isomorphism

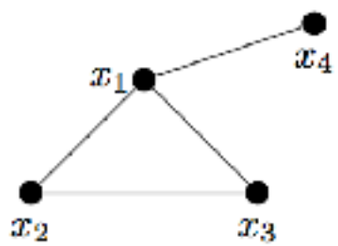
GW is a distance on the "space of all spaces":

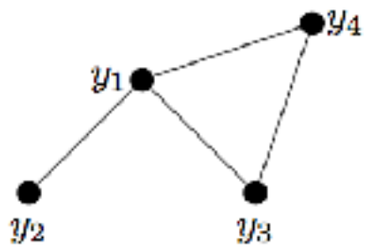
$$\mathbb{X} = \{(\mathcal{X}, d_{\mathcal{X}}, \mu \in \mathcal{P}(\mathcal{X})); d_{\mathcal{X}} \text{ metric } \} \text{ (mm-spaces)}$$

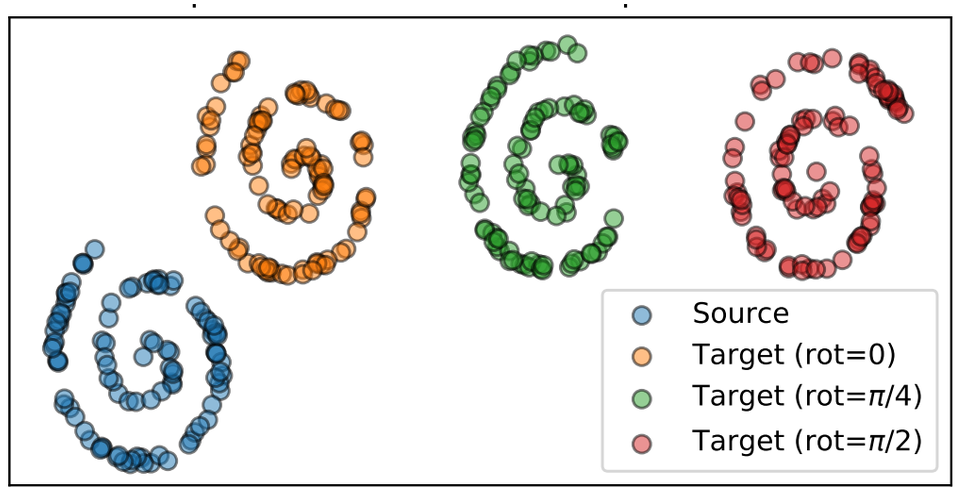
- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x,x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

Isometry: permutations, rotations, translations,...







...to Gromov-Wasserstein

Gromov-Wasserstein distance


Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

A distance w.r.t isomorphism

GW is a distance on the "space of all spaces":

$\mathbb{X} = \{(\mathcal{X}, d_{\mathcal{X}}, \mu \in \mathcal{P}(\mathcal{X})); d_{\mathcal{X}} \text{ metric }\}$ (mm-spaces)

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x,x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

  ϕ is measure-preserving: $\phi \# \mu = \nu$

Push-forward $\phi \# \mu$

$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \underset{\phi\#\mu}{\to} \sum_{i=1}^n a_i \delta_{\phi(x_i)}$$

...to Gromov-Wasserstein

Gromov-Wasserstein distance

Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

A distance w.r.t isomorphism

GW is a distance on the "space of all spaces":

$\mathbb{X} = \{(\mathcal{X}, d_{\mathcal{X}}, \mu \in \mathcal{P}(\mathcal{X})); d_{\mathcal{X}} \text{ metric }\}$ (mm-spaces)

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x,x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

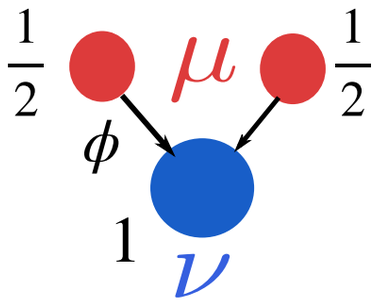
  ϕ is measure-preserving: $\phi \# \mu = \nu$

(Weights are compatible)

Push-forward $\phi \# \mu$

$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \underset{\phi\#\mu}{\to} \sum_{i=1}^n a_i \delta_{\phi(x_i)}$$

Compatible



$$\frac{1}{2} + \frac{1}{2} \to 1$$

...to Gromov-Wasserstein

Gromov-Wasserstein distance


Gromov-Wasserstein distance

$$GW_p^p(c_{\mathcal{X}}, c_{\mathcal{Y}}, \mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \int_{\mathcal{X}\times\mathcal{Y}} \left|c_{\mathcal{X}}(x,x') - c_{\mathcal{Y}}(y,y')\right|^p \mathrm{d}\pi(x,y)\mathrm{d}\pi(x',y'))$$

A distance w.r.t isomorphism

GW is a distance on the "space of all spaces":

$\mathbb{X} = \{(\mathcal{X}, d_{\mathcal{X}}, \mu \in \mathcal{P}(\mathcal{X})); d_{\mathcal{X}} \text{ metric } \}$ (mm-spaces)

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x,x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

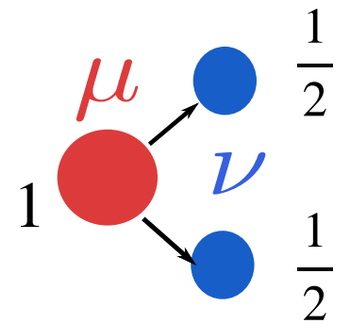
  ϕ is measure-preserving: $\phi \# \mu = \nu$

(Weights are compatible)

Push-forward $\phi \# \mu$

$$\mu = \sum_{i=1}^n a_i \delta_{x_i} \underset{\phi\#\mu}{\to} \sum_{i=1}^n a_i \delta_{\phi(x_i)}$$

Not compatible



$$1 \nrightarrow (\frac{1}{2}, \frac{1}{2})$$

...to Gromov-Wasserstein

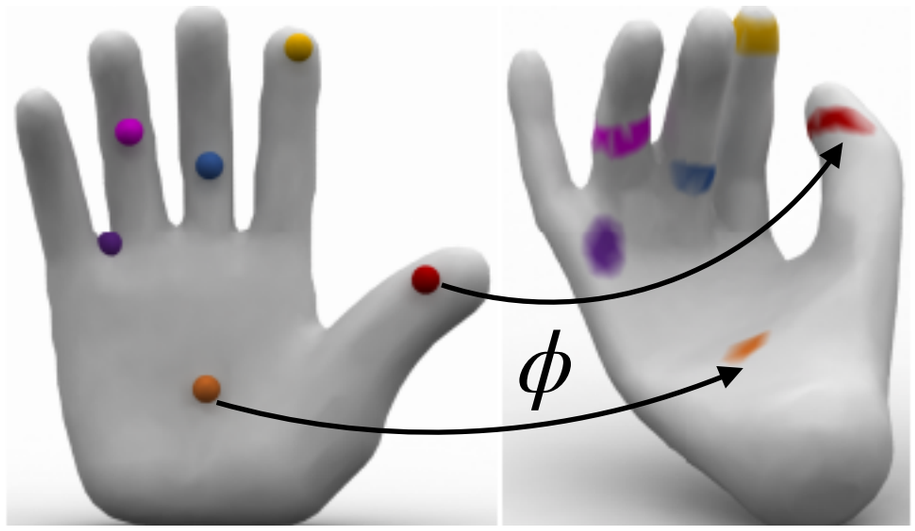
Gromov-Wasserstein distance

Gromov-Wasserstein = a bending invariant distance

- $GW_p(d_{\mathcal{X}}, d_{\mathcal{Y}}, \mu, \nu) = 0$ iff $\exists \phi : \mathcal{X} \to \mathcal{Y}$

  ϕ is a isometry $d_{\mathcal{X}}(x, x') = d_{\mathcal{Y}}(\phi(x), \phi(x'))$

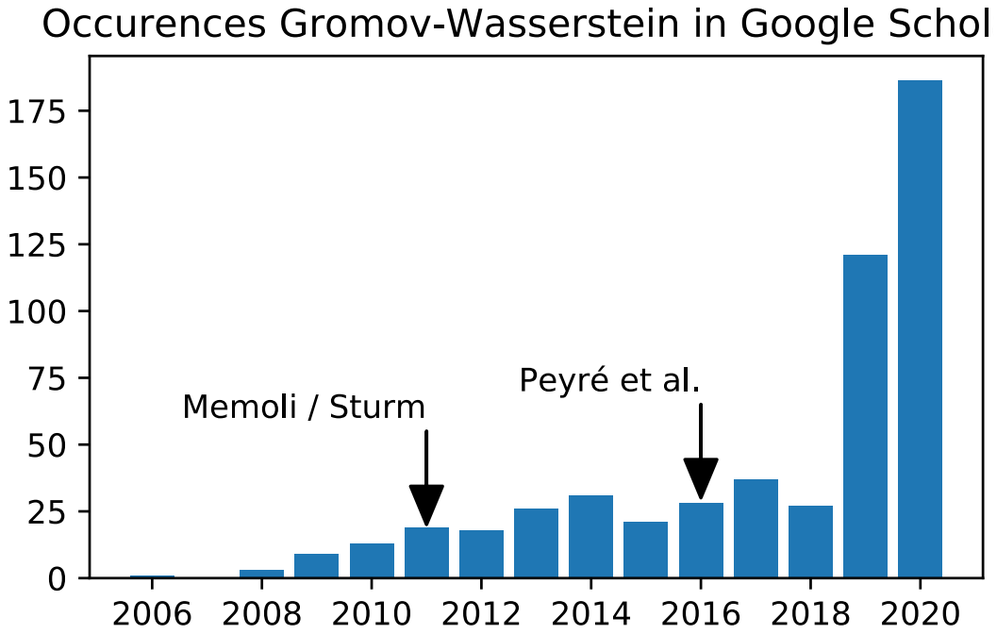
  ϕ is measure-preserving $\phi \# \mu = \nu$



[Solomon 2016]

Applications for geometric data

- Barycenter of relational data [Peyré 2016], Point clouds/meshes [Ezuz 2017]
- Shape comparison [Mémoli 2011, Solomon 2016]
- Graphs [Vayer 2019, Xu 2019, Fey 2020, Vincent-Cuaz 2021], biology [Demetci 2020], generative modeling [Bunne 2019]



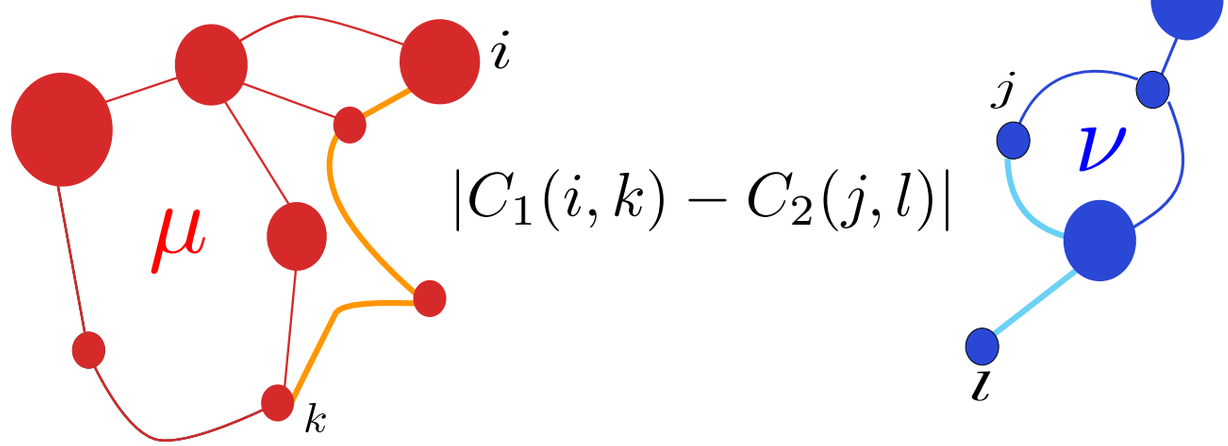
Solving OT

A quadratic problem (QP)

Discrete probability measures

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{j=1}^{m} b_j \delta_{y_j}$$

$$\mathcal{X}, \mathcal{Y} \not\subset \Omega$$



$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{ijkl} |C_1(i,k) - C_2(j,l)|^p \pi_{ij} \pi_{kl}$$

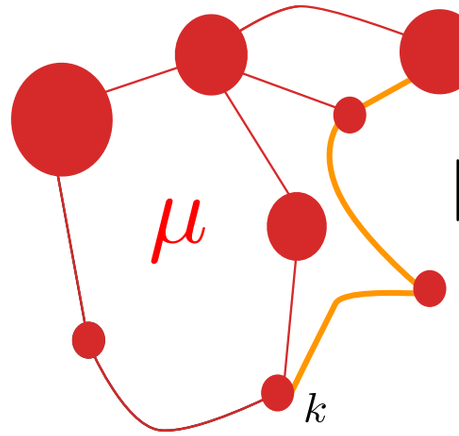
Solving OT

A quadratic problem (QP)

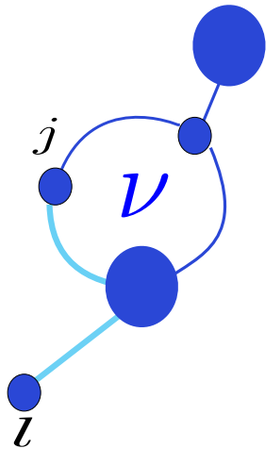
Discrete probability measures

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{j=1}^{m} b_j \delta_{y_j}$$

$$\mathcal{X}, \mathcal{Y} \not\subset \Omega$$



$$|C_1(i,k) - C_2(j,l)|$$



$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{ijkl} |C_1(i,k) - C_2(j,l)|^p \pi_{ij} \pi_{kl}$$

Non convex QP: NP-hard in general (graph matching problem)

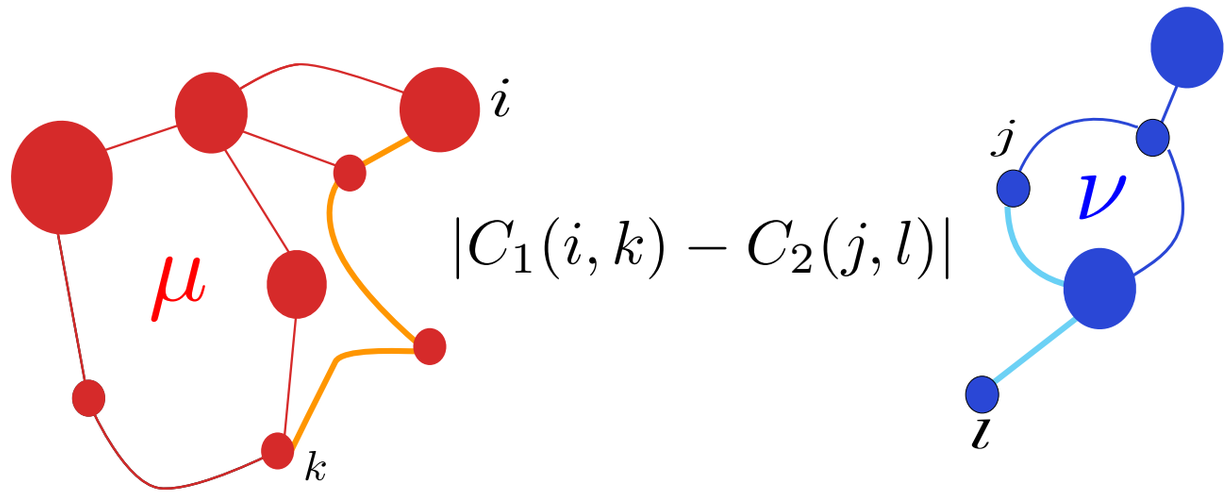
Solving OT

A quadratic problem (QP)

Discrete probability measures

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{j=1}^{m} b_j \delta_{y_j}$$

$$\mathcal{X}, \mathcal{Y} \not\subset \Omega$$



$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{ijkl} |C_1(i,k) - C_2(j,l)|^p \pi_{ij} \pi_{kl} \; -\varepsilon H(\boldsymbol{\pi})$$

- Non convex QP: NP-hard in general
- With entropic regularization [Peyré 2016, Solomon 2016]
- Can be solved using projected gradient descent under KL geometry
- Each gradient step: Sinkhorn algorithm

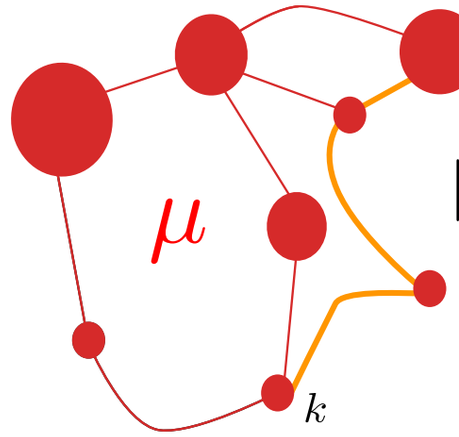
Solving OT

A quadratic problem (QP)

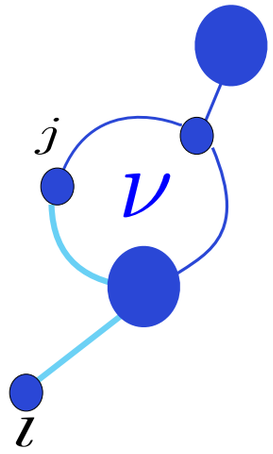
Discrete probability measures

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{j=1}^{m} b_j \delta_{y_j}$$

$$\mathcal{X}, \mathcal{Y} \not\subset \Omega$$



$$|C_1(i,k) - C_2(j,l)|$$



$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a},\mathbf{b})} \sum_{ijkl} |C_1(i,k) - C_2(j,l)|^p \pi_{ij} \pi_{kl} \; -\varepsilon H(\boldsymbol{\pi})$$

- Non convex QP: NP-hard in general
- With entropic regularization [Peyré 2016, Solomon 2016] $\sim O(n_{iter} * n^2 \log(n))$
- Can be solved using projected gradient descent under KL geometry
- Each gradient step: Sinkhorn algorithm

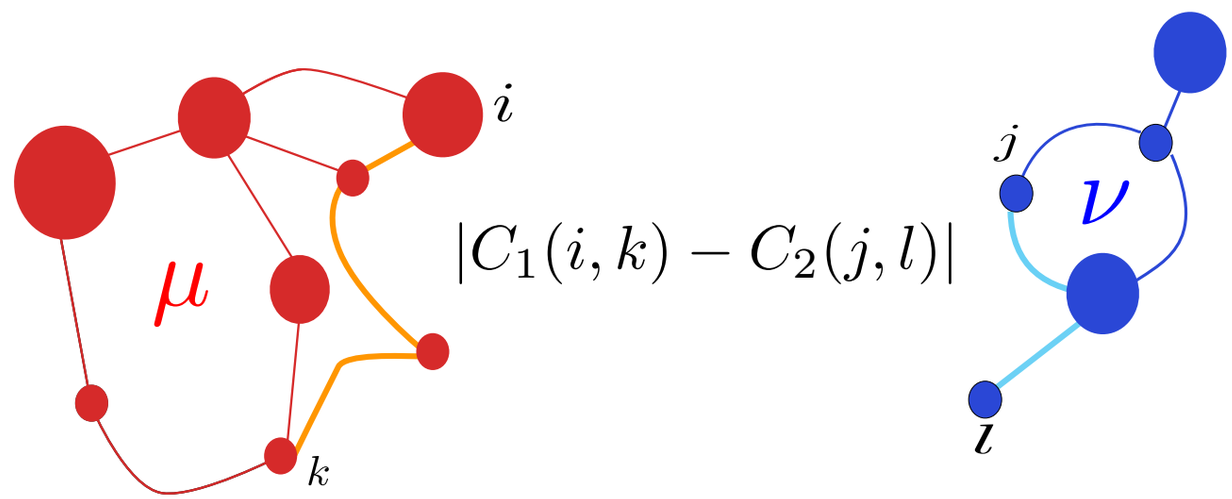
Solving OT

A quadratic problem (QP)

Discrete probability measures

$$\mu = \sum_{i=1}^{n} a_i \delta_{x_i} \quad \nu = \sum_{j=1}^{m} b_j \delta_{y_j}$$

$$\mathcal{X}, \mathcal{Y} \not\subset \Omega$$



$$\min_{\boldsymbol{\pi} \in \Pi(\mathbf{a}, \mathbf{b})} \sum_{ijkl} |C_1(i,k) - C_2(j,l)|^p \pi_{ij} \pi_{kl}$$

- Non convex QP: NP-hard in general
- With entropic regularization [Peyré 2016, Solomon 2016] $\sim O(n_{iter} * n^2 \log(n))$
- Can be solved using projected gradient descent under KL geometry
- Each gradient step: Sinkhorn algorithm

Hard to solve and even to approximate…

CO-Optimal Transport

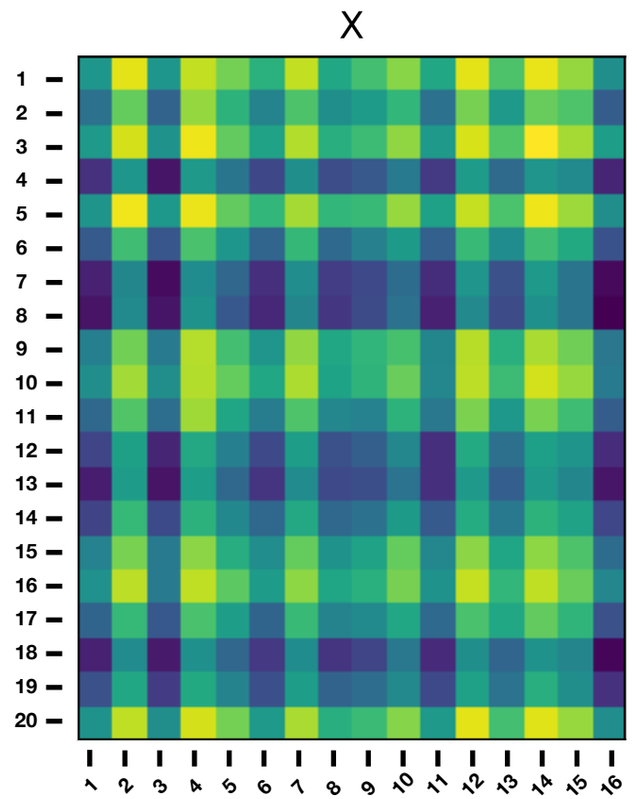
Motivations

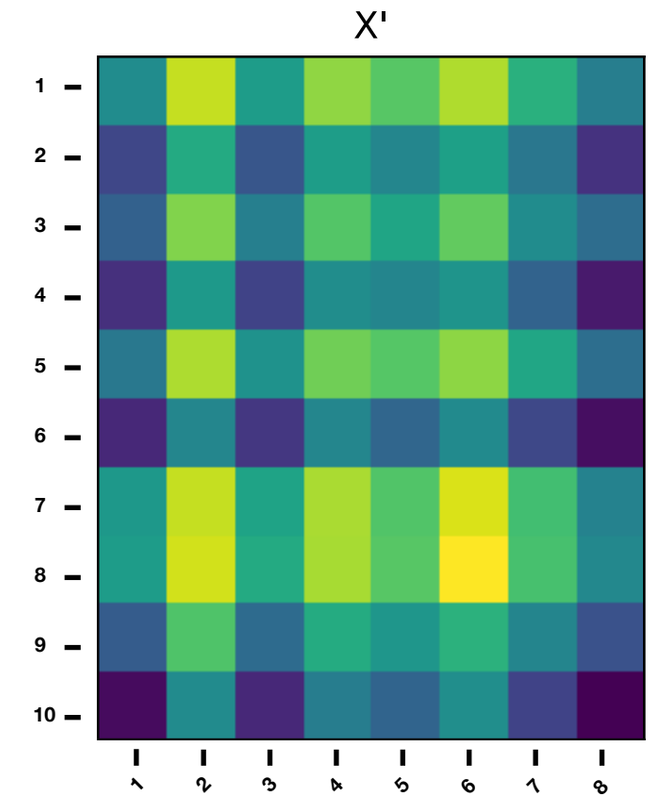
Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Row= samples, columns= features/variable





We want to measure the similarity of these two datasets (interpretable way)

Image registration [Haker 2001], HDA [Yang 2018], Word embeddings [Alvarez 2018]

GW is limited in this scenario

CO-Optimal Transport

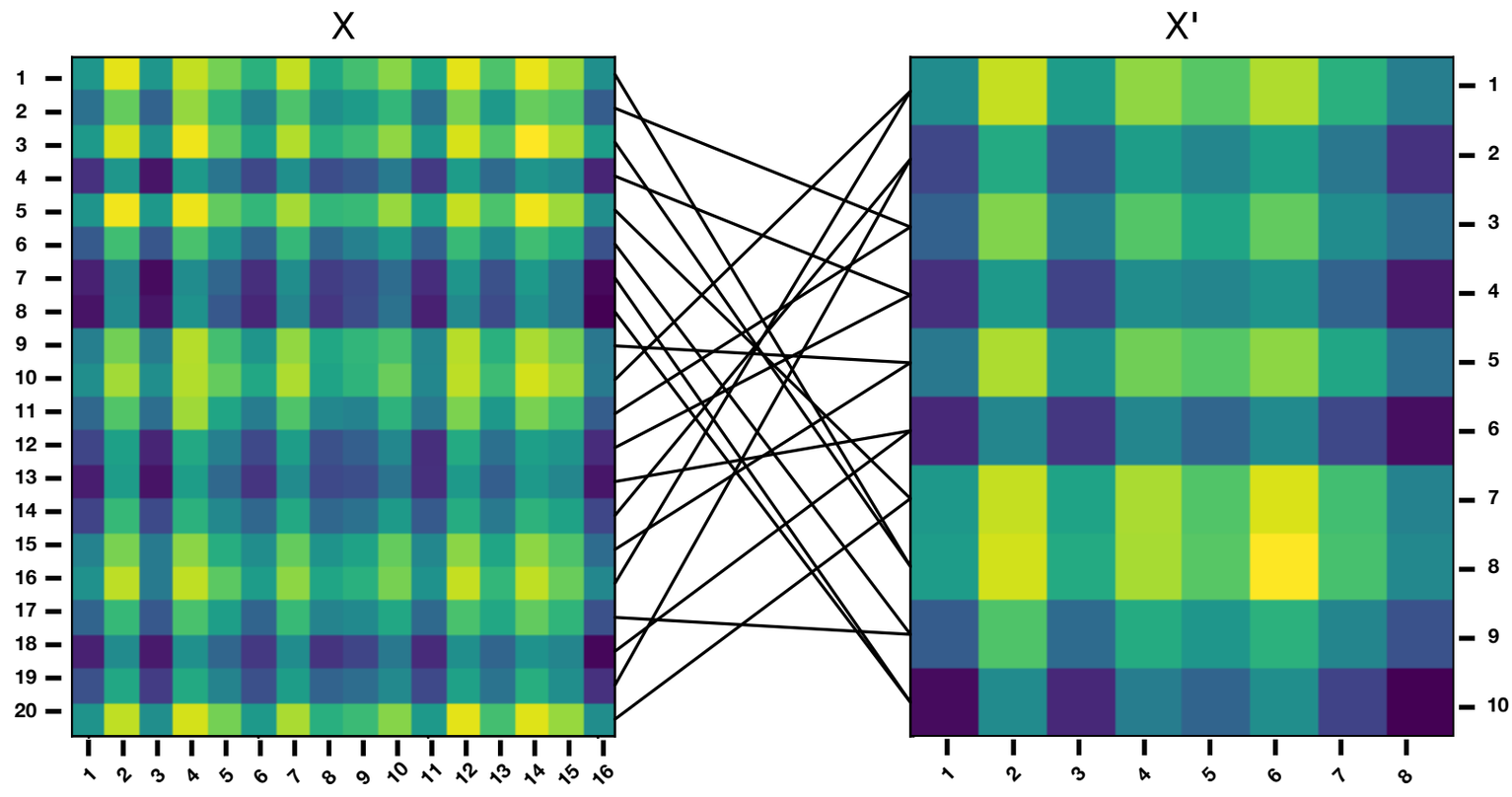
Motivations

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Row= samples, columns= features/variable



We can apply Gromov-Wasserstein based on the pairwise distances

$$c_X(\mathbf{x}_i, \mathbf{x}_j)$$
$$c_{X'}(\mathbf{x}'_i, \mathbf{x}'_j)$$

The OT matrix gives a reordering of the samples

CO-Optimal Transport

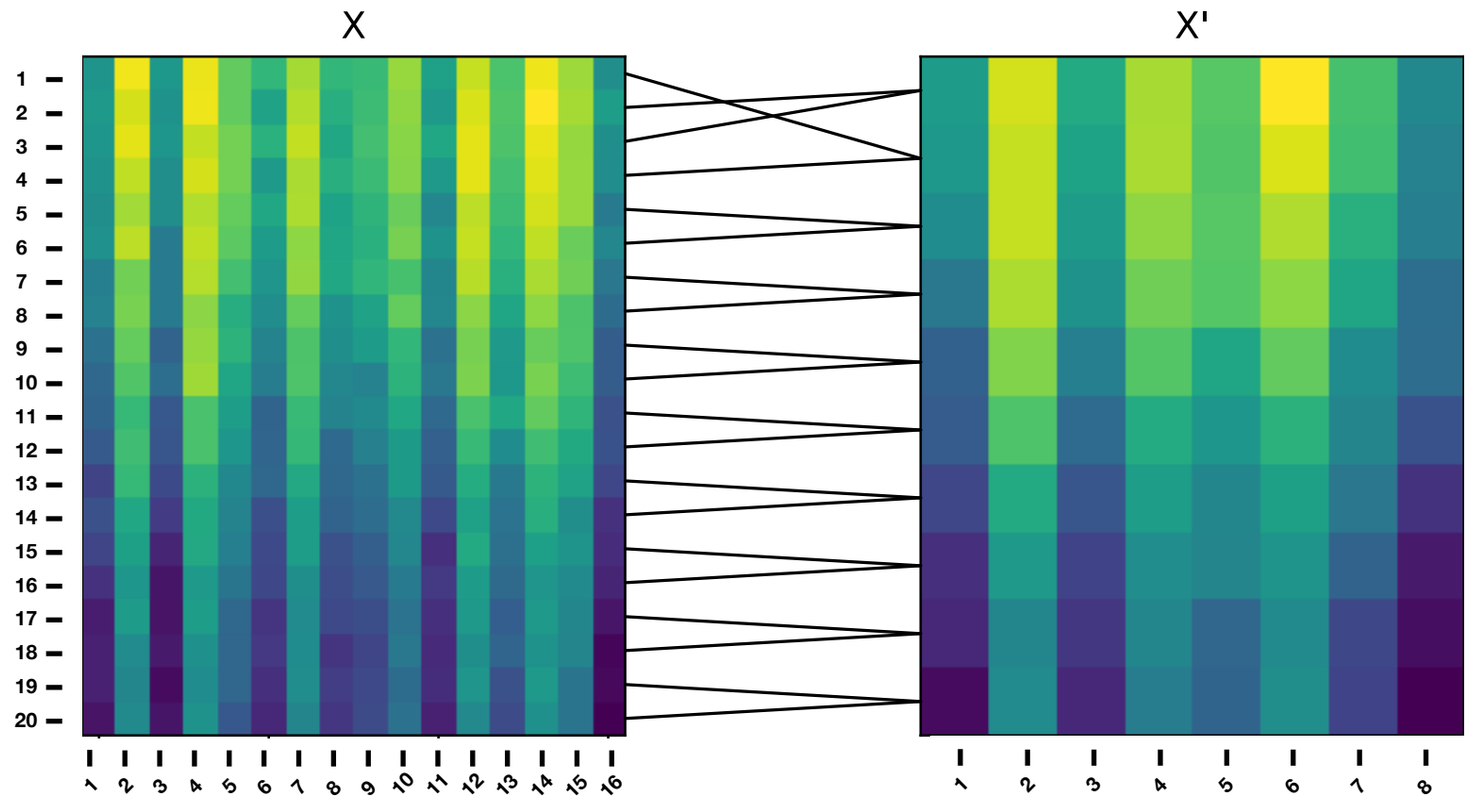
Motivations

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \ldots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Row= samples, columns= features/variable



We can apply Gromov-Wasserstein based on the pairwise distances

$$c_X(\mathbf{x}_i, \mathbf{x}_j)$$
$$c_{X'}(\mathbf{x}'_i, \mathbf{x}'_j)$$

The OT matrix gives a reordering of the samples

CO-Optimal Transport

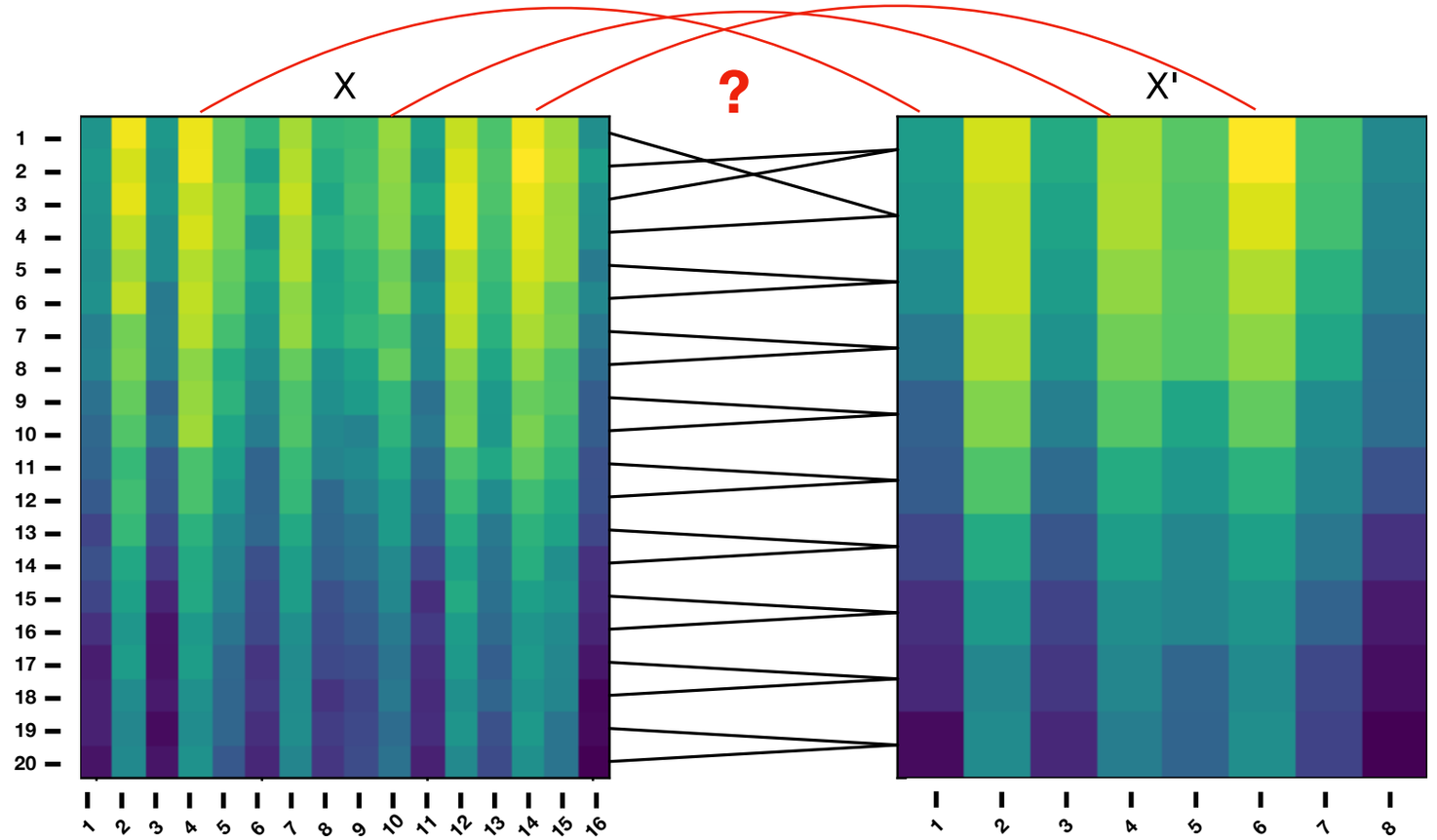
Motivations

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Row= samples, columns= features/variable



We can apply Gromov-Wasserstein based on the pairwise distances

$$c_X(\mathbf{x}_i, \mathbf{x}_j)$$
$$c_{X'}(\mathbf{x}'_i, \mathbf{x}'_j)$$

The OT matrix gives a reordering of the samples

But discards the relationship between the features…

CO-Optimal Transport

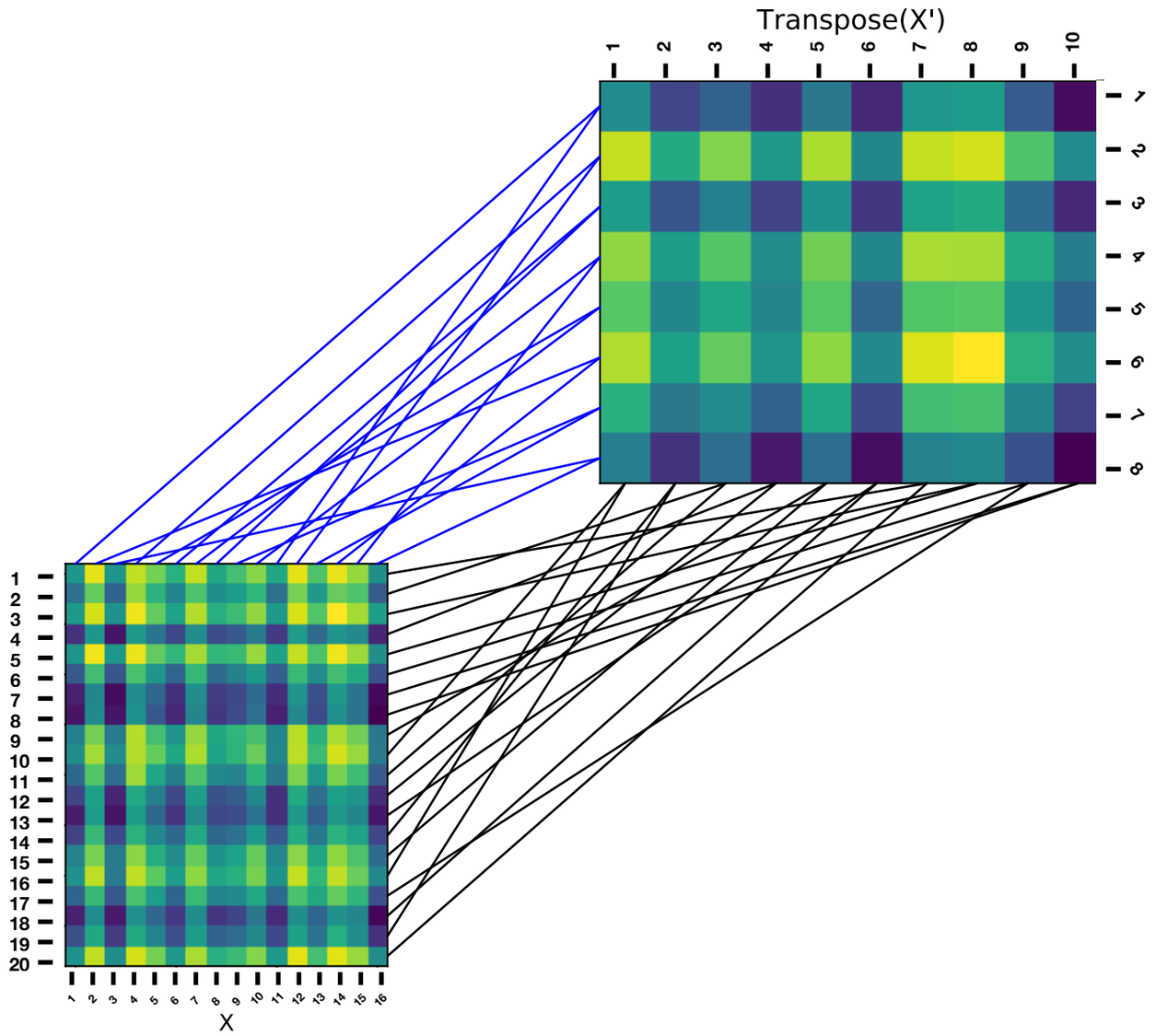
Motivations

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \ldots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Row= samples, columns= features/variable



COOT: estimate a transport matrix between the samples **and** one between the features

These matrices are estimated jointly and can be used for interpreting relationships across spaces

CO-Optimal Transport

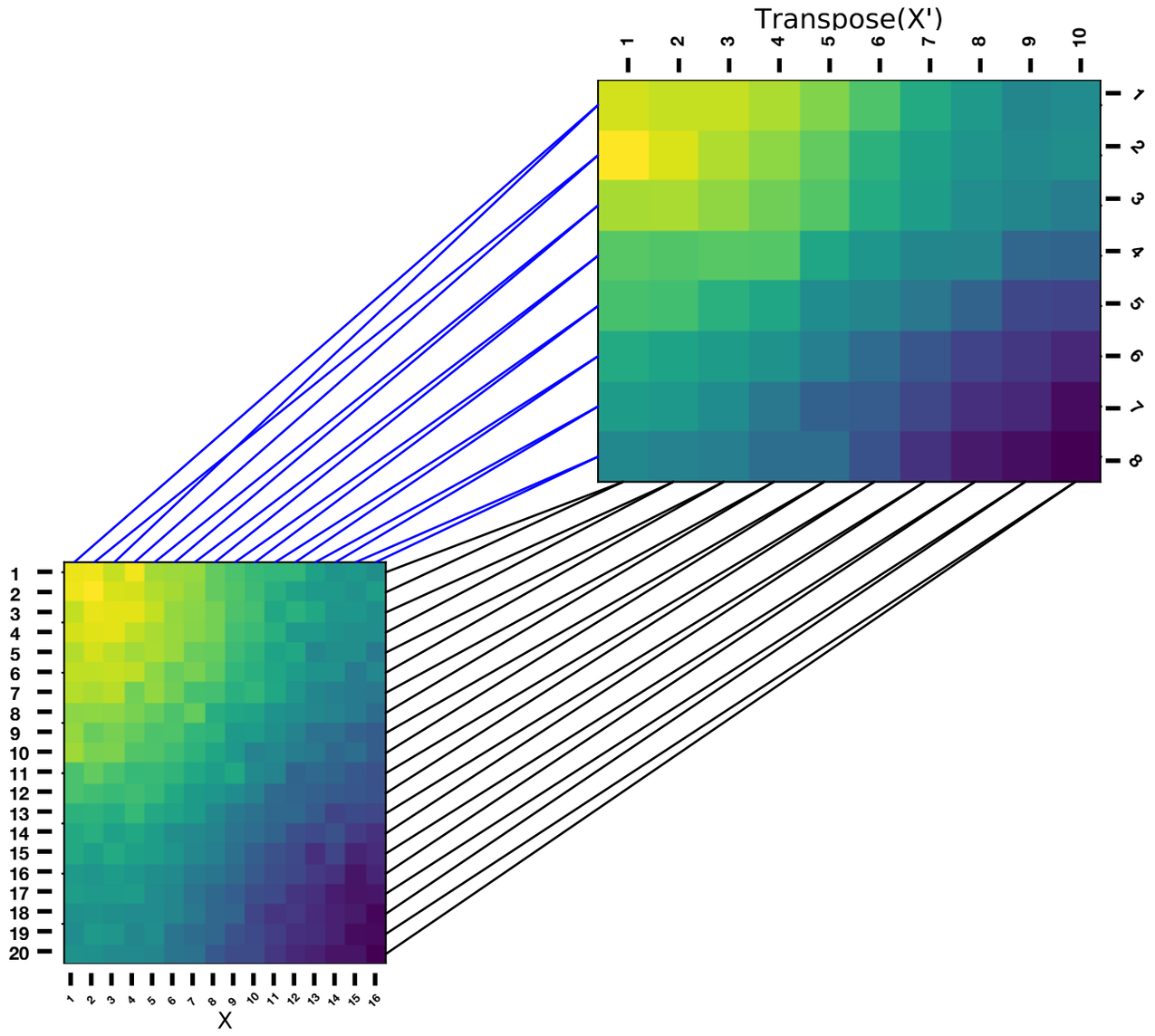
Motivations

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \ldots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Row= samples, columns= features/variable



The objective of COOT is to estimate a transport matrix between the samples **and** one between the features

These matrices are estimated jointly and can be used for interpreting relationships across spaces

CO-Optimal Transport

Formulation & example

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Weights

Samples: $\mathbf{w} \in \Sigma_n, \mathbf{w}' \in \Sigma_{n'}$

Features: $\mathbf{v} \in \Sigma_d, \mathbf{v}' \in \Sigma_{d'}$

CO-Optimal Transport

$$\min_{\substack{\boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}')}} \sum_{i,j,k,l} |X_{i,k} - X'_{j,l}|^p \boldsymbol{\pi}^s_{i,j} \boldsymbol{\pi}^v_{k,l}$$

$\boldsymbol{\pi}^s$: transport matrix between the samples

$\boldsymbol{\pi}^v$: transport matrix between the features/variables

CO-Optimal Transport [NeurIPS 2020]

Formulation & example

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Weights

Samples: $\mathbf{w} \in \Sigma_n, \mathbf{w}' \in \Sigma_{n'}$

Features: $\mathbf{v} \in \Sigma_d, \mathbf{v}' \in \Sigma_{d'}$

CO-Optimal Transport

$$\min_{\substack{\boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}')}} \sum_{i,j,k,l} |X_{i,k} - X'_{j,l}|^p \boldsymbol{\pi}^s_{i,j} \boldsymbol{\pi}^v_{k,l}$$

Applied on the raw values

$\boldsymbol{\pi}^s$: transport matrix between the samples

$\boldsymbol{\pi}^v$: transport matrix between the features/variables

CO-Optimal Transport

Formulation & example

Two heterogeneous datasets

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$$

$$\mathbf{X}' = [\mathbf{x}'_1, \dots, \mathbf{x}'_{n'}]^T \in \mathbb{R}^{n' \times d'}$$

Weights

Samples: $\mathbf{w} \in \Sigma_n, \mathbf{w}' \in \Sigma_{n'}$

Features: $\mathbf{v} \in \Sigma_d, \mathbf{v}' \in \Sigma_{d'}$

CO-Optimal Transport

$$\min_{\substack{\boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}')}} \sum_{i,j,k,l} |X_{i,k} - X'_{j,l}|^p \boldsymbol{\pi}^s_{i,j} \boldsymbol{\pi}^v_{k,l}$$

Applied on the raw values

$\boldsymbol{\pi}^s$: transport matrix between the samples

$\boldsymbol{\pi}^v$: transport matrix between the features/variables

Regularized version: add an entropy term for each transport matrix

CO-Optimal Transport

Formulation & example

CO-Optimal Transport

$$\min_{\substack{\boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}')}} \sum_{i,j,k,l} |X_{i,k} - X'_{j,l}|^p \boldsymbol{\pi}^s_{i,j} \boldsymbol{\pi}^v_{k,l}$$

MNIST/USPS example:

Samples: images, Features: pixels

$n = n' = 3000$

$d = 28 * 28, d' = 16 * 16$

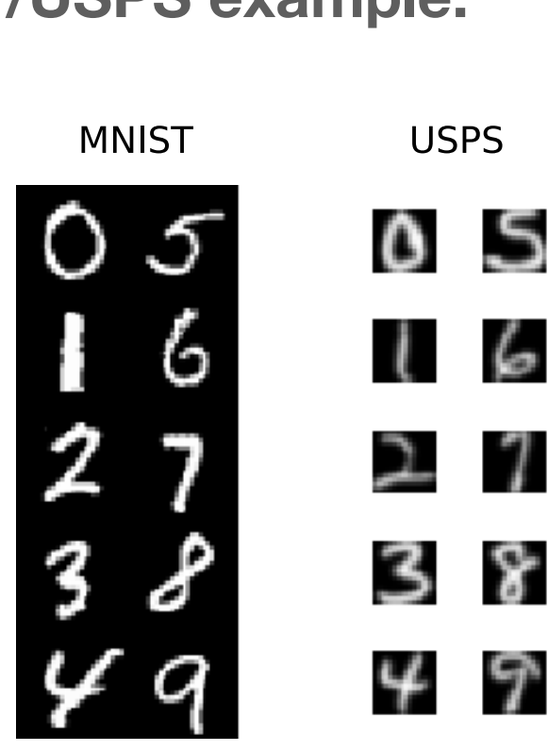
CO-Optimal Transport

Formulation & example

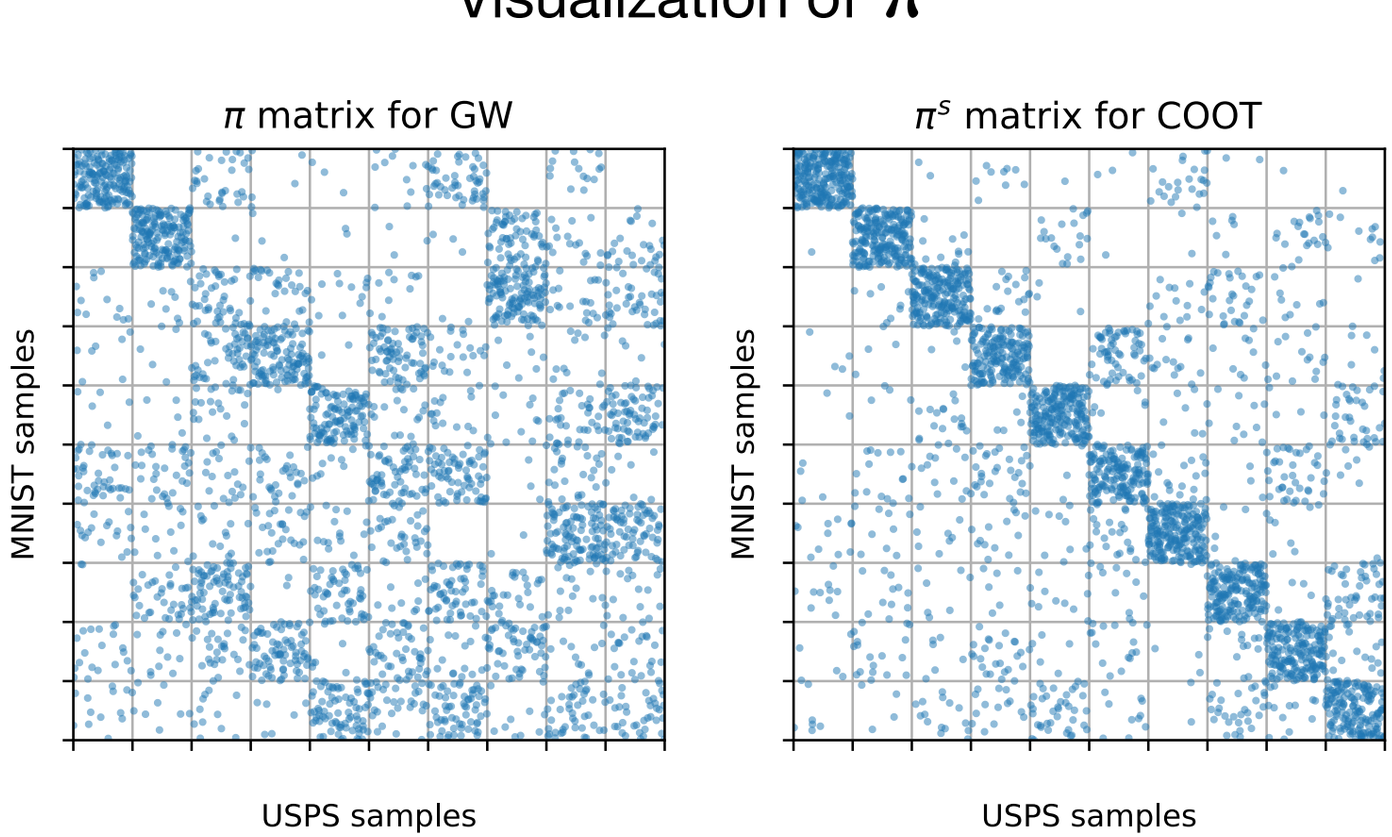
CO-Optimal Transport

$$\min_{\substack{\boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}')}} \sum_{i,j,k,l} |X_{i,k} - X'_{j,l}|^p \boldsymbol{\pi}^s_{i,j} \boldsymbol{\pi}^v_{k,l}$$

MNIST/USPS example:



Visualization of $\boldsymbol{\pi}^s$



Better class correspondence

CO-Optimal Transport

Formulation & example

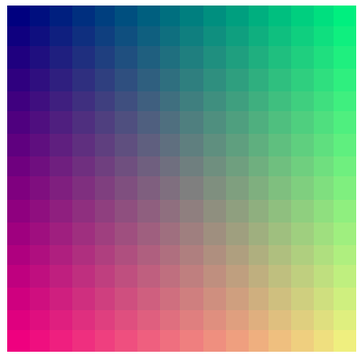
CO-Optimal Transport

$$\min_{\substack{\boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}')}} \sum_{i,j,k,l} |X_{i,k} - X'_{j,l}|^p \boldsymbol{\pi}^s_{i,j} \boldsymbol{\pi}^v_{k,l}$$

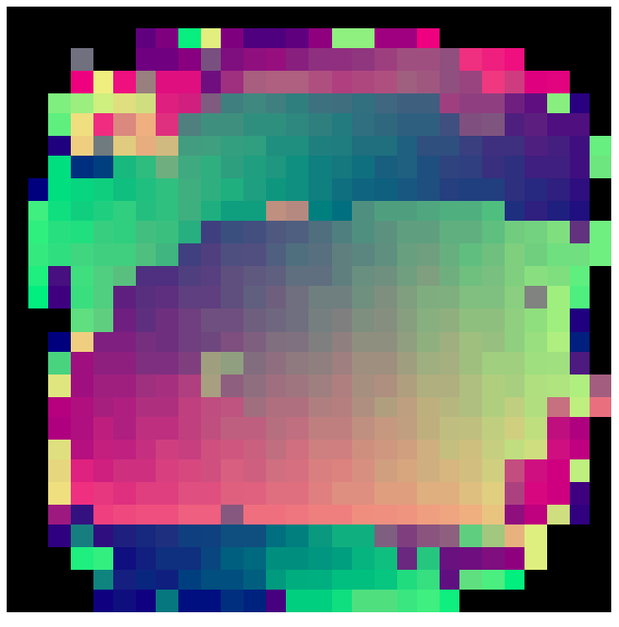
MNIST/USPS example:

Visualization of $\boldsymbol{\pi}^v$

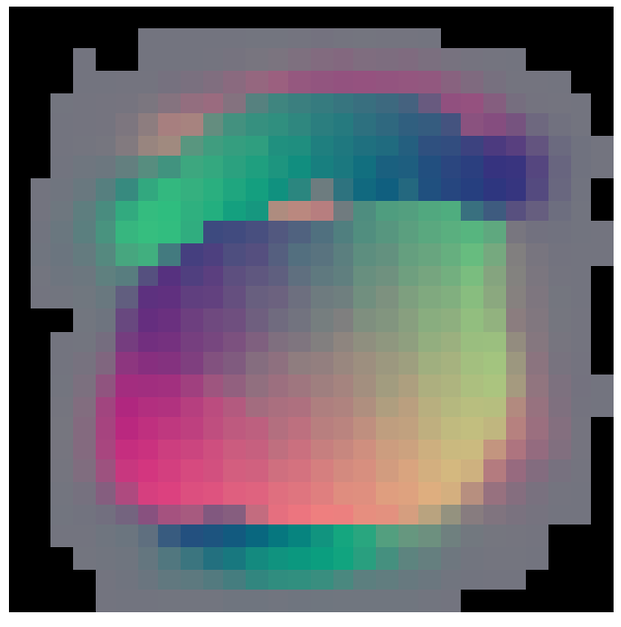
USPS colored pixels



MNIST pixels through π^v



MNIST pixels through entropic π^v



Spatial structure preserved (without supervision!)

CO-Optimal Transport

Solving COOT

CO-Optimal Transport

$$\min_{\substack{\boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}')}} \sum_{i,j,k,l} |X_{i,k} - X'_{j,l}|^p \boldsymbol{\pi}^s_{i,j} \boldsymbol{\pi}^v_{k,l}$$

Non-convex bilinear program: NP-Hard

Block Coordinate Descent (BCD): alternates OT problems -> local minima [Konno 1976]

Algorithm 1 BCD for COOT

$\pi^s_{(0)} \leftarrow \mathbf{w}\mathbf{w}'^T, \pi^v_{(0)} \leftarrow \mathbf{v}\mathbf{v}'^T, k \leftarrow 0$
while $k <$ maxIt **and** $err > 0$ **do**
  $\boldsymbol{\pi}^v_{(k)} \leftarrow OT(\mathbf{v}, \mathbf{v}', \mathbf{L}(\mathbf{X}, \mathbf{X}') \otimes \boldsymbol{\pi}^s_{(k-1)})$
  $\boldsymbol{\pi}^s_{(k)} \leftarrow OT(\mathbf{w}, \mathbf{w}', \mathbf{L}(\mathbf{X}, \mathbf{X}') \otimes \boldsymbol{\pi}^v_{(k-1)})$
  $err \leftarrow ||\boldsymbol{\pi}^v_{(k-1)} - \boldsymbol{\pi}^v_{(k)}||_F$
  $k \leftarrow k + 1$
end while

CO-Optimal Transport

Solving COOT

CO-Optimal Transport

$$\min_{\boldsymbol{\pi}^s \in \Pi(\mathbf{w},\mathbf{w}')} \sum_{i,j,k,l} |X_{i,k} - X'_{j,l}|^p \boldsymbol{\pi}^s_{i,j} \boldsymbol{\pi}^v_{k,l}$$
$$\boldsymbol{\pi}^v \in \Pi(\mathbf{v},\mathbf{v}')$$

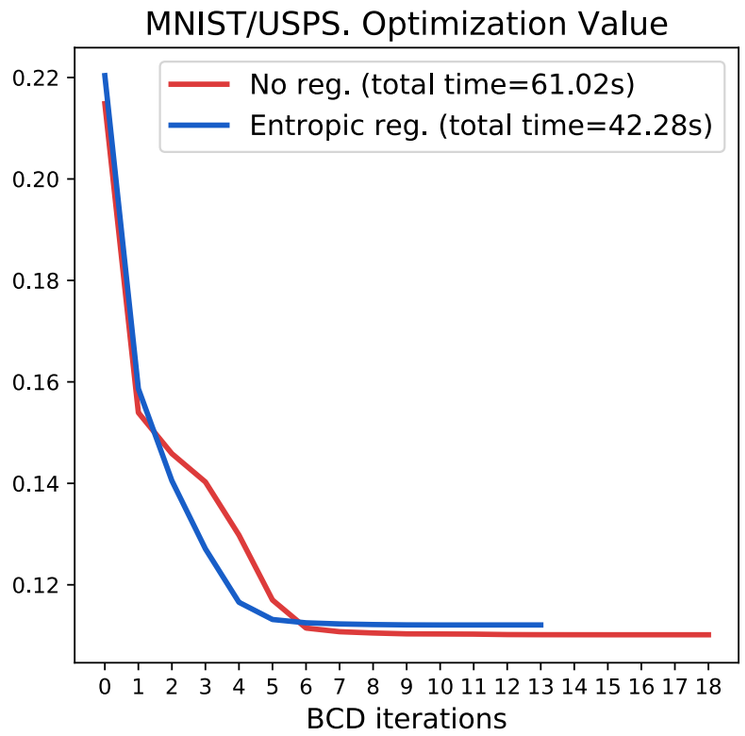
Non-convex bilinear program: NP-Hard

Block Coordinate Descent (BCD): alternates OT problems -> local minima [Konno 1976]

Algorithm 1 BCD for COOT

1: $\pi^s_{(0)} \leftarrow \mathbf{w}\mathbf{w}'^T, \pi^v_{(0)} \leftarrow \mathbf{v}\mathbf{v}'^T, k \leftarrow 0$
2: **while** $k <$ maxIt **and** $err > 0$ **do**
3: $\quad \boldsymbol{\pi}^v_{(k)} \leftarrow OT(\mathbf{v}, \mathbf{v}', \mathbf{L}(\mathbf{X}, \mathbf{X}') \otimes \boldsymbol{\pi}^s_{(k-1)})$
4: $\quad \boldsymbol{\pi}^s_{(k)} \leftarrow OT(\mathbf{w}, \mathbf{w}', \mathbf{L}(\mathbf{X}, \mathbf{X}') \otimes \boldsymbol{\pi}^v_{(k-1)})$
5: $\quad err \leftarrow ||\boldsymbol{\pi}^v_{(k-1)} - \boldsymbol{\pi}^v_{(k)}||_F$
6: $\quad k \leftarrow k + 1$
7: **end while**

In practice BCD converges in few iterations



CO-Optimal Transport

Solving COOT

CO-Optimal Transport

$$\min_{\substack{\boldsymbol{\pi}^s \in \Pi(\mathbf{w}, \mathbf{w}') \\ \boldsymbol{\pi}^v \in \Pi(\mathbf{v}, \mathbf{v}')}} \sum_{i,j,k,l} |X_{i,k} - X'_{j,l}|^p \boldsymbol{\pi}^s_{i,j} \boldsymbol{\pi}^v_{k,l}$$

Non-convex bilinear program: NP-Hard

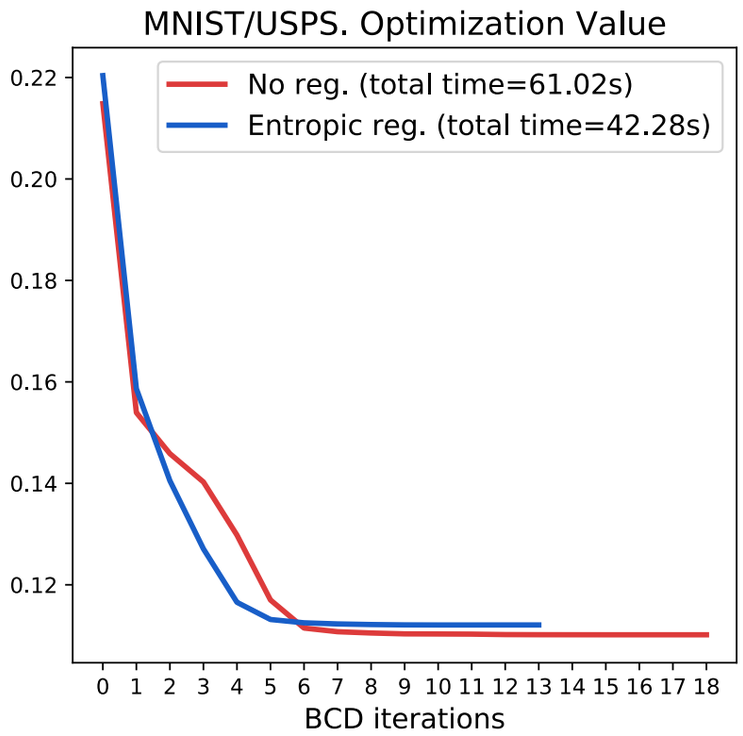
Block Coordinate Descent (BCD): alternates OT problems -> local minima [Konno 1976]

Algorithm 1 BCD for COOT

$\pi^s_{(0)} \leftarrow \mathbf{w}\mathbf{w}'^T, \pi^v_{(0)} \leftarrow \mathbf{v}\mathbf{v}'^T, k \leftarrow 0$
while $k <$ maxIt **and** $err > 0$ **do**
  $\boldsymbol{\pi}^v_{(k)} \leftarrow OT(\mathbf{v}, \mathbf{v}', \mathbf{L}(\mathbf{X}, \mathbf{X}') \otimes \boldsymbol{\pi}^s_{(k-1)})$
  $\boldsymbol{\pi}^s_{(k)} \leftarrow OT(\mathbf{w}, \mathbf{w}', \mathbf{L}(\mathbf{X}, \mathbf{X}') \otimes \boldsymbol{\pi}^v_{(k-1)})$
  $err \leftarrow ||\boldsymbol{\pi}^v_{(k-1)} - \boldsymbol{\pi}^v_{(k)}||_F$
  $k \leftarrow k + 1$
end while

can be used for GW in the concave regime!

In practice BCD converges in few iterations



Gradient Flows

In the Wasserstein space

Gradient flows

Point of View:

- Define Wasserstein gradient flows
- Analogies with gradient flows in Euclidean space
- For more abstract views, see [Santambrogio, 2017, Ambrosio et al., 2008]
- Talk from Anna Korba `https://mathtube.org/lecture/video/wasserstein-gradient-flows-machine-learning`

Gradient Flows on $\mathbb{R}^p$

Let $X = \mathbb{R}^p$, d a distance (*e.g.* $d(x, y) = \|x - y\|_2$), $F : X \to \mathbb{R}$.

Goal:

$$\min_x \; F(x)$$

Gradient Flows on $\mathbb{R}^p$

Let $X = \mathbb{R}^p$, d a distance (*e.g.* $d(x, y) = \|x - y\|_2$), $F : X \to \mathbb{R}$.

Goal:

$$\min_x \ F(x)$$

Definition (Gradient Flow on $\mathbb{R}^p$)
A gradient flow is a curve $x : [0, T] \to X$ which decreases as much as possible along the functional F.

i.e. If F is differentiable, x follows the Cauchy problem

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t}(t) = -\nabla F(x(t)) \\ x(0) = x_0 \end{cases}$$

Gradient Flows on $\mathbb{R}^p$

If F is differentiable, x follows the Cauchy problem

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t}(t) = -\nabla F(x(t)) \\ x(0) = x_0 \end{cases}$$

Solving the ODE in practice:

- Explicit Euler scheme ($x_k = x(k\tau)$):

$$x_{k+1} = x_k - \tau \nabla F(x_k)$$

- Implicit Euler scheme:

$$\begin{aligned} x_{k+1} = x_k - \tau \nabla F(x_{k+1}) &\iff 0 = \frac{x_{k+1} - x_k}{\tau} + \nabla F(x_{k+1}) \\ &\iff x_{k+1} \in \operatorname*{argmin}_{x \in X} \frac{\|x - x_k\|_2^2}{2\tau} + F(x) \\ &\iff x_{k+1} = \mathrm{prox}_{\tau F}(x_k) \end{aligned}$$

- Any ODE solver (Runge-Kutta...)

Other characterization

See [Santambrogio, 2017, Ambrosio et al., 2008]

- Energy Dissipation Equality (EDE):

$$\begin{aligned} &\frac{\mathrm{d}x}{\mathrm{d}t}(t) = -\nabla F(x(t)) \iff \forall 0 \leq s < t \leq 1, \\ &F(x(s)) - F(x(t)) = \int_s^t \left(\frac{1}{2}|x'(u)|^2 + \frac{1}{2}|\nabla F(x(u))|^2 \right) \mathrm{d}u, \end{aligned} \tag{22}$$

  where (speed)

$$|x'(t)| = \lim_{h \to 0} \frac{d\big(x(t+h), x(t)\big)}{h}$$

  and (descending slope)

$$|\nabla F|(x) = \limsup_{y \to x} \frac{|F(y) - F(x)|}{d(x,y)}.$$

- Evolution Variational Inequality (EVI): For F λ-geodesically convex,

$$x'(t) \in \partial F(x(t)) \iff \forall y \in X,\ \frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{2}|x(t) - y|^2 \leq F(y) - F(x(t)) - \frac{\lambda}{2}|x(t) - y|^2 \tag{23}$$

JKO Scheme

Let $\mathcal{P}_2(\mathbb{R}^p) = \{\mu \in \mathcal{P}(\mathbb{R}^p),\ \int \|x\|^2 \ \mathrm{d}\mu(x) < +\infty\}$.

Define Gradient Flows in $(\mathcal{P}_2(\mathbb{R}^p), W_2)$ via the JKO Scheme [Jordan et al., 1998]: Let $F : \mathcal{P}_2(\mathbb{R}^p) \to \mathbb{R}$, $\mu_0 \in \mathcal{P}_2(\mathbb{R}^p)$,

$$\forall k \geq 0,\ \mu_{k+1}^\tau \in \underset{\mu \in \mathcal{P}_2(\mathbb{R}^p)}{\operatorname{argmin}} \ \frac{W_2^2(\mu, \mu_k^\tau)}{2\tau} + F(\mu) = \mathrm{JKO}_{\tau F}(\mu_k^\tau) \tag{24}$$

Define a piecewise constant interpolation μ^τ, *i.e.*

$$\forall t \in [k\tau, (k+1)\tau[,\ \mu_t^\tau = \mu_k^\tau$$

Wasserstein gradient flows: $t \mapsto \mu_t$ such that, for $\mu^\tau \xrightarrow[\tau \to 0]{} \mu$.

(also called (generalized) minimizing movement [Bonnotte, 2013, Liutkus et al., 2019])

Wasserstein Gradient Flows

Gradient Flow in $(\mathcal{P}_2(\mathbb{R}^d), W_2)$:

Iterated Minimization scheme (JKO Scheme) [Jordan et al., 1998]:

$$\mu_{k+1}^{\tau} \in \underset{\mu \in \mathcal{P}_2(\mathbb{R}^d)}{\operatorname{argmin}} \ \frac{1}{2\tau} W_2^2(\mu, \mu_k^{\tau}) + F(\mu)$$

Examples

- $F(\mu) = \int \rho(x) \log \rho(x) \ \mathrm{d}x + \int V(x)\rho(x)\mathrm{d}x$ if $\mathrm{d}\mu = \rho \mathrm{dLeb}$
  Solution in the limit $\tau \to 0$ to the PDE: (Fokker-Planck)

$$\partial_t \rho_t = \operatorname{div}(\rho_t \nabla V) + \Delta \rho_t$$

- $F(\mu) = \frac{1}{2} SW_2^2(\mu, \nu) + \lambda \mathcal{H}(\mu)$ [Bonnotte, 2013, Liutkus et al., 2019]
- $F(\mu) = \frac{1}{2} MMD^2(\mu, \nu)$ [Arbel et al., 2019]
- $F(\mu) = \frac{1}{2} \mathrm{KSD}^2(\mu, \nu)$ [Korba et al., 2021]

Numerical Methods

- If an associated SDE is known, simulate from it
  [Liu et al., 2021, Liutkus et al., 2019, Arbel et al., 2019, Korba et al., 2021]

Examples

Let $F(\mu) = \int V(x)\rho(x)\mathrm{d}x + \int \log(\rho(x))\rho(x)\mathrm{d}x$,

Gradient Flow solution of:

$$\partial_t \rho_t = \mathrm{div}(\rho_t \nabla V) + \Delta \rho_t$$

Associated SDE (Langevin Equation):

$$\mathrm{d}X_t = -\nabla V(X_t)\mathrm{d}t + \sqrt{2}\ \mathrm{d}W_t$$

First Algorithms of Resolution of WGFs

If SDE is unknow, need specific numerical methods to solve for Wasserstein Gradient Flows by the JKO Scheme:

- By approximating W_2:
  - By the entropic regularized OT problem + Dykstra's algorithm + $\mu = \sum_{i=1}^n \rho_i \delta_{x_i}$, $(x_i)_i$ grid [Peyré, 2015, Carlier et al., 2017]

$$\forall k,\ \mu_{k+1}^\tau \in \underset{\mu \in \mathcal{P}_2(\mathbb{R}^p)}{\operatorname{argmin}} \ \frac{W_\epsilon^2(\mu, \mu_k^\tau)}{2\tau} + F(\mu) \tag{25}$$

  - By the dual formulation of the entropic OT problem [Caluya and Halder, 2019, Frogner and Poggio, 2020]
  - By using SW_2 [Bonet et al., 2021] + Neural networks g^θ and implicit modeling, *i.e.* $\mu = g_{\#}^\theta p_Z$

$$\forall k,\ \mu_{k+1}^\tau \in \underset{\mu \in \mathcal{P}_2(\mathbb{R}^p)}{\operatorname{argmin}} \ \frac{SW_2^2(\mu, \mu_k^\tau)}{2\tau} + F(\mu) \tag{26}$$

- By using the dynamic formulation of the transport + grid discretization [Laborde, 2016, Carrillo et al., 2021]
- JKO-ICNN [Alvarez-Melis et al., 2021, Mokrov et al., 2021, Bunne et al., 2021]

Theorem (Brenier's Theorem)
Let $\mu, \nu \in \mathcal{P}_2(\mathbb{R}^d)$, μ absolutely continuous with respect to the Lebesgue measure. Then, the optimal coupling γ^* is unique and of the form $\gamma^* = (Id, \nabla\varphi)_{\#}\mu$ with φ is a convex function.

- Reformulate the problem as:

$$u_{k+1}^{\tau} \in \underset{u \in \mathrm{cvx}}{\operatorname{argmin}} \ \frac{1}{2\tau} \int \|\nabla u(x) - x\|_2^2 \ \rho_k^{\tau}(x)\mathrm{d}x + F\big((\nabla u)_{\#}\rho_k^{\tau}\big)$$

- Implicitly define $\rho_{k+1}^{\tau} = (\nabla u_{k+1}^{\tau})_{\#}\rho_k^{\tau}$
- Use Input Convex Neural Networks (ICNN) [Amos et al., 2017] to model the convex functions:

$$\theta_{k+1}^{\tau} \in \underset{\theta \in \{\theta, u_\theta \in \mathrm{ICNN}\}}{\operatorname{argmin}} \ \frac{1}{2\tau} \int \|\nabla_x u_\theta(x) - x\|_2^2 \ \rho_k^{\tau}(x)\mathrm{d}x + F\big((\nabla_x u_\theta)_{\#}\rho_k^{\tau}\big)$$

- Backpropagate through gradient

First Algorithms of Resolution of WGFs

Resolution of Wasserstein Gradient Flows by the JKO Scheme:

- By approximating W_2
- By using the dynamic formulation of the transport + grid discretization [Laborde, 2016, Carrillo et al., 2021]
- JKOICNN: Modeling the Monge map with ICNNs [Alvarez-Melis et al., 2021, Mokrov et al., 2021, Bunne et al., 2021]

$$\theta_{k+1}^{\tau} \in \underset{\tau \in \{\theta, u_\theta \in \mathrm{ICNN}\}}{\operatorname{argmin}} \frac{1}{2\tau} \int \|\nabla u_\theta(x) - x\|_2^2 \, \mathrm{d}\mu_k^{\tau}(x) + F\big((\nabla_x u_\theta)_{\#}\mu_k^{\tau}\big) \tag{27}$$

- Modeling directly the Monge map

$$T_{k+1}^{\tau} \in \underset{T}{\operatorname{argmin}} \frac{1}{2\tau} \int \|T(x) - x\|_2^2 \, \mathrm{d}\mu_k^{\tau}(x) + F(T_{\#}\mu_k^{\tau}) \tag{28}$$

Summary

Optimal transport is a well theoretically grounded ways of comparing probability distributions

- that allows to compare empirical distributions in a non-parametric ways
- that leverages on a ground metric in the embedding space
- for which exist several algorithmic solutions

It comes in several flavours:

- Monge problem: find a mapping (transport map)
- Kantorovich problem: find a coupling (transport plan)

+ many applications not covered in this course

POT (PYTHON OPTIMAL TRANSPORT TOOLBOX)

https://pythonot.github.io/

README.md

POT: Python Optimal Transport

pypi package 0.4.0 | build passing | docs passing

This open source Python library provide several solvers for optimization problems related to Optimal Transport for signal, image processing and machine learning.

It provides the following solvers:

- OT solver for the linear program/ Earth Movers Distance [1].
- Entropic regularization OT solver with Sinkhorn Knopp Algorithm [2] and stabilized version [9][10] with optional GPU implementation (required cudamat).
- Bregman projections for Wasserstein barycenter [3] and unmixing [4].
- Optimal transport for domain adaptation with group lasso regularization [5]
- Conditional gradient [6] and Generalized conditional gradient for regularized OT [7].
- Joint OT matrix and mapping estimation [8].
- Wasserstein Discriminant Analysis [11] (requires autograd + pymanopt).
- Gromov-Wasserstein distances and barycenters [12]

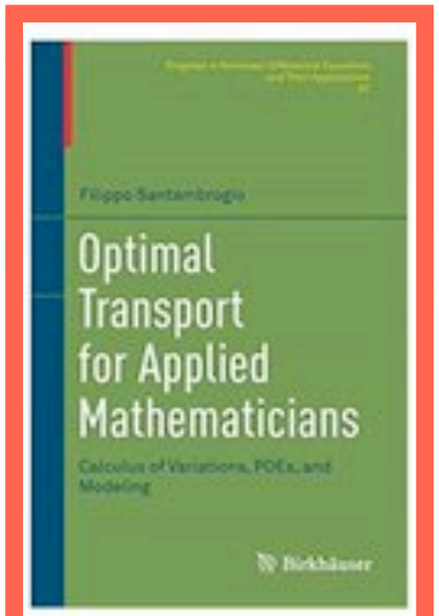
Some demonstrations (both in Python and Jupyter Notebook format) are available in the examples folder.

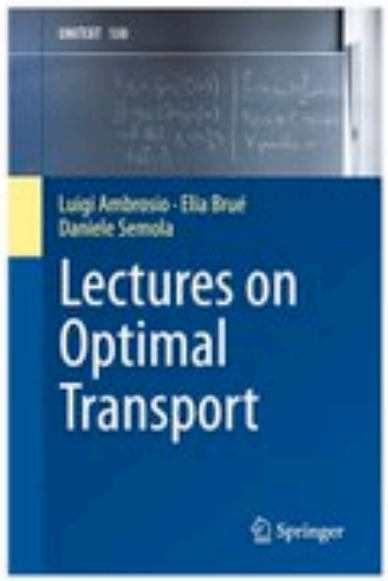
Installation

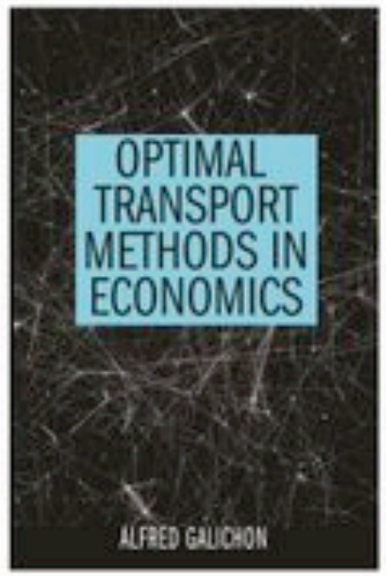
The library has been tested on Linux, MacOSX and Windows. It requires a C++ compiler for using the EMD solver and relies on the following Python modules:

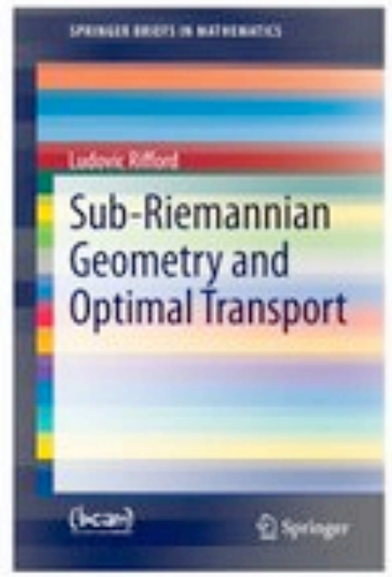
- Numpy (>=1.11)

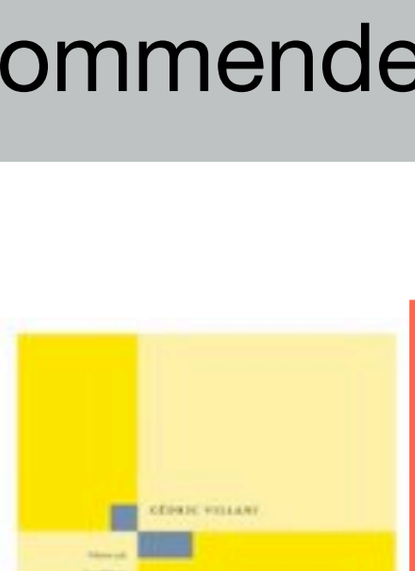
Recommended readings/books

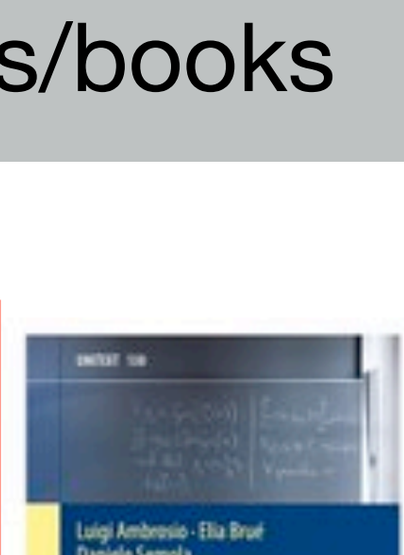


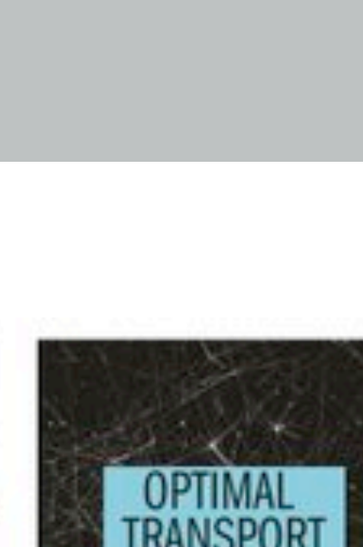












References i

Agueh, M. and Carlier, G. (2011).

Barycenters in the wasserstein space.

SIAM Journal on Mathematical Analysis, 43(2):904–924.

Alvarez-Melis, D., Schiff, Y., and Mroueh, Y. (2021).

Optimizing functionals on the space of probabilities with input convex neural networks.

Ambrosio, L., Gigli, N., and Savaré, G. (2008).

Gradient flows: in metric spaces and in the space of probability measures.

Springer Science & Business Media.

Amos, B., Xu, L., and Kolter, J. Z. (2017).

Input convex neural networks.

In *International Conference on Machine Learning*, pages 146–155. PMLR.


Arbel, M., Korba, A., Salim, A., and Gretton, A. (2019).
Maximum mean discrepancy gradient flow.
arXiv preprint arXiv:1906.04370.


Benamou, J.-D., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. (2015).
Iterative Bregman projections for regularized transportation problems.
SISC.


Bigot, J., Gouet, R., Klein, T., López, A., et al. (2017).
Geodesic pca in the wasserstein space by convex pca.
In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 53, pages 1–26. Institut Henri Poincaré.


Bonet, C., Berg, P., Courty, N., Septier, F., Drumetz, L., and Pham, M.-T. (2022).
Spherical sliced-wasserstein.

References iii

Bonet, C., Courty, N., Septier, F., and Drumetz, L. (2021).

Sliced-wasserstein gradient flows.

Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. (2015).

Sliced and radon Wasserstein barycenters of measures.

Journal of Mathematical Imaging and Vision, 51:22–45.

Bonnotte, N. (2013).

Unidimensional and evolution methods for optimal transportation.

PhD thesis, Paris 11.

Brenier, Y. (1991).

Polar factorization and monotone rearrangement of vector-valued functions.

Communications on pure and applied mathematics, 44(4):375–417.

Bunne, C., Meng-Papaxanthos, L., Krause, A., and Cuturi, M. (2021).

Jkonet: Proximal optimal transport modeling of population dynamics.

Caluya, K. F. and Halder, A. (2019).

Proximal recursion for solving the fokker-planck equation.

In *2019 American Control Conference (ACC)*, pages 4098–4103. IEEE.

Carlier, G., Duval, V., Peyré, G., and Schmitzer, B. (2017).

Convergence of entropic schemes for optimal transport and gradient flows.

SIAM Journal on Mathematical Analysis, 49(2):1385–1418.

Carrillo, J. A., Craig, K., Wang, L., and Wei, C. (2021).

Primal dual methods for wasserstein gradient flows.

Foundations of Computational Mathematics, pages 1–55.

Courty, N., Flamary, R., and Ducoffe, M. (2017).

Learning wasserstein embeddings.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2016).

Optimal transport for domain adaptation.

IEEE Transactions on Pattern Analysis and Machine Intelligence.

Cuturi, M. (2013).

Sinkhorn distances: Lightspeed computation of optimal transportation.

In *Neural Information Processing Systems (NIPS)*, pages 2292–2300.

Delon, J., Salomon, J., and Sobolevski, A. (2010).

Fast transport optimization for monge costs on the circle.

SIAM Journal on Applied Mathematics, 70(7):2239–2258.

Dessein, A., Papadakis, N., and Rouas, J.-L. (2016).

Regularized optimal transport and the rot mover's distance.

arXiv preprint arXiv:1610.06447.

Ferradans, S., Papadakis, N., Peyré, G., and Aujol, J.-F. (2014).

Regularized discrete optimal transport.

SIAM Journal on Imaging Sciences, 7(3).

Frogner, C. and Poggio, T. (2020).

Approximate inference with wasserstein gradient flows.

In *International Conference on Artificial Intelligence and Statistics*, pages 2581–2590. PMLR.

Genevay, A., Cuturi, M., Peyré, G., and Bach, F. (2016).

Stochastic optimization for large-scale optimal transport.

In *NIPS*, pages 3432–3440.

Hundrieser, S., Klatt, M., and Munk, A. (2021).

The statistics of circular optimal transport.

arXiv preprint arXiv:2103.15426.

Jordan, R., Kinderlehrer, D., and Otto, F. (1998).

The variational formulation of the fokker–planck equation.

SIAM journal on mathematical analysis, 29(1):1–17.

Kantorovich, L. (1942).

On the translocation of masses.

C.R. (Doklady) Acad. Sci. URSS (N.S.), 37:199–201.

Korba, A., Aubin-Frankowski, P.-C., Majewski, S., and Ablin, P. (2021).

Kernel stein discrepancy descent.

arXiv preprint arXiv:2105.09994.

Laborde, M. (2016).

Interacting particles systems, Wasserstein gradient flow approach.

PhD thesis, PSL Research University.

Liu, S., Sun, H., and Zha, H. (2021).

Approximating the optimal transport plan via particle-evolving method.

Liutkus, A., Simsekli, U., Majewski, S., Durmus, A., and Stöter, F.-R. (2019).

Sliced-wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions.

In *International Conference on Machine Learning*, pages 4104–4113. PMLR.

McCann, R. J. (1997).

A convexity principle for interacting gases.

Advances in mathematics, 128(1):153–179.

Mokrov, P., Korotin, A., Li, L., Genevay, A., Solomon, J., and Burnaev, E. (2021).

Large-scale wasserstein gradient flows.

Monge, G. (1781).

Mémoire sur la théorie des déblais et des remblais.

De l'Imprimerie Royale.

Peyré, G. (2015).

Entropic approximation of wasserstein gradient flows.

SIAM Journal on Imaging Sciences, 8(4):2323–2351.

Rabin, J., Delon, J., and Gousseau, Y. (2011a).

Transportation distances on the circle.

Journal of Mathematical Imaging and Vision, 41(1):147–167.

Rabin, J., Peyré, G., Delon, J., and Bernot, M. (2011b).

Wasserstein barycenter and its application to texture mixing.

In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 435–446. Springer.

Rubner, Y., Tomasi, C., and Guibas, L. J. (2000).

The earth mover's distance as a metric for image retrieval.

International journal of computer vision, 40(2):99–121.

References x

Santambrogio, F. (2014).

Introduction to optimal transport theory.

Notes.

Santambrogio, F. (2017).

{Euclidean, metric, and Wasserstein} gradient flows: an overview.

Bulletin of Mathematical Sciences, 7(1):87–154.

Schmitz, M. A., Heitz, M., Bonneel, N., Mboula, F. M. N., Coeurjolly, D., Cuturi, M., Peyré, G., and Starck, J.-L. (2017).

Wasserstein dictionary learning: Optimal transport-based unsupervised non-linear dictionary learning.

arXiv preprint arXiv:1708.01955.

Seguy, V., Bhushan Damodaran, B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. (2017).

Large-scale optimal transport and mapping estimation.

Seguy, V. and Cuturi, M. (2015).

Principal geodesic analysis for probability measures under the optimal transport metric.

In *Advances in Neural Information Processing Systems*, pages 3312–3320.

Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., and Guibas, L. (2015).

Convolutional wasserstein distances: Efficient optimal transportation on geometric domains.

ACM Transactions on Graphics (TOG), 34(4):66.