Discrete Optimal Transport: Scaling up to 1,000,000 samples in 1s

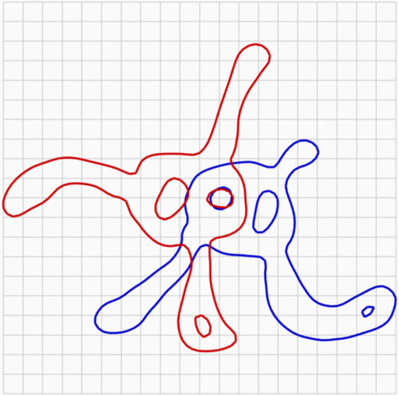
Jean Feydy

Cortona, Tuscany – June 2019

Écoles Normales Supérieures de Paris et Paris-Saclay
Collaboration with B. Charlier, J. Glaunès (KeOps library);
F.-X. Vialard, G. Peyré, T. Séjourné, A. Trouvé (OT theory).

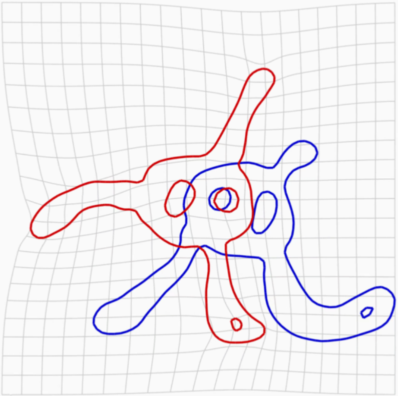
My main motivation: shape registration

Source A, target B,



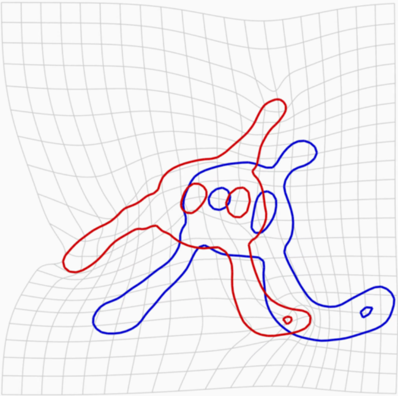
My main motivation: shape registration

Source A, target B, mapping φ



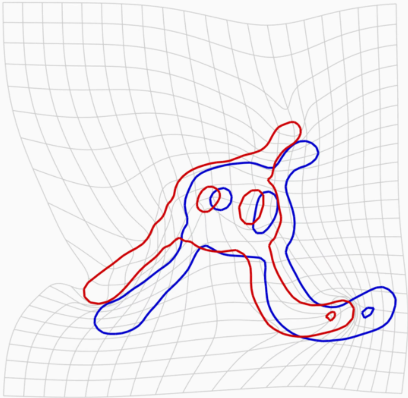
My main motivation: shape registration

Source A, target B, mapping φ



My main motivation: shape registration

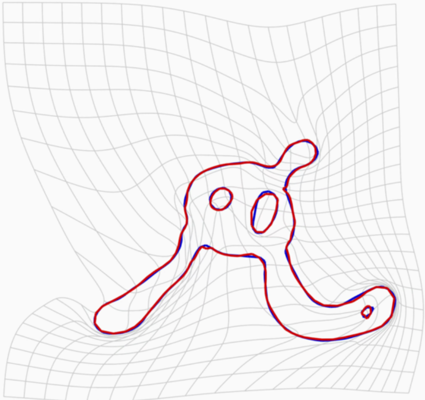
Source A, target B, mapping φ



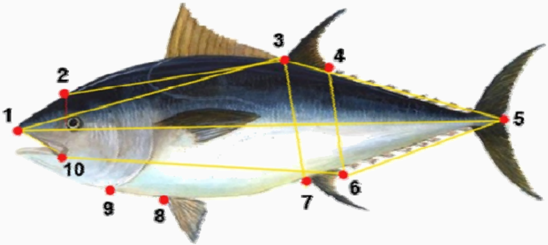
My main motivation: shape registration

Source A, target B, mapping φ

$$A \xrightarrow[\text{Model}]{\varphi} \varphi(A) = A' \underset{\textbf{Loss}}{\rightleftarrows} B$$

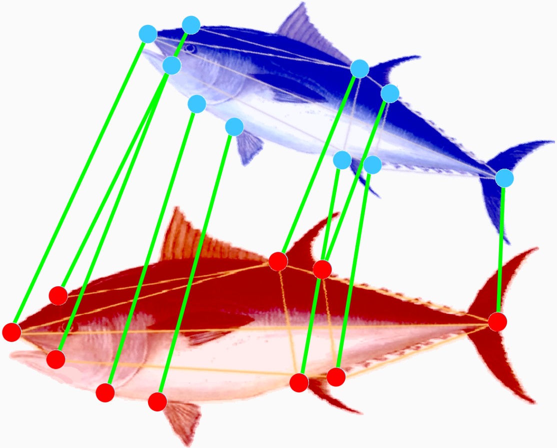


On labeled shapes, use a spring energy



Anatomical landmarks from *A morphometric approach for the analysis of body shape in bluefin tuna*, Addis et al., 2009.

On labeled shapes, use a spring energy



Anatomical landmarks from *A morphometric approach for the analysis of body shape in bluefin tuna*, Addis et al., 2009.

Encoding unlabeled shapes as measures

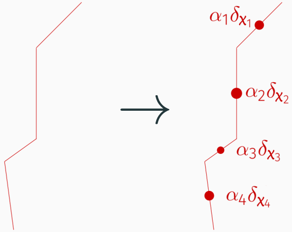
Let's enforce sampling invariance:

$$A \longrightarrow \alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}\,, \qquad B \longrightarrow \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}\,.$$

Encoding unlabeled shapes as measures

Let's enforce sampling invariance:

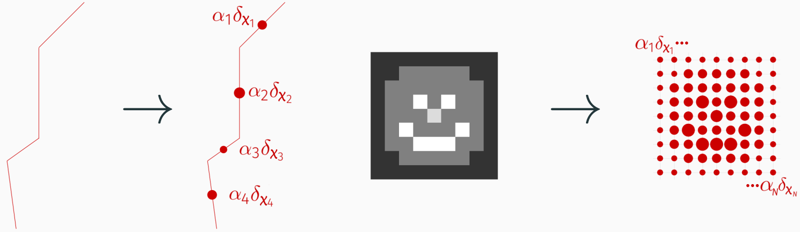
$$A \longrightarrow \alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}, \qquad B \longrightarrow \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}.$$



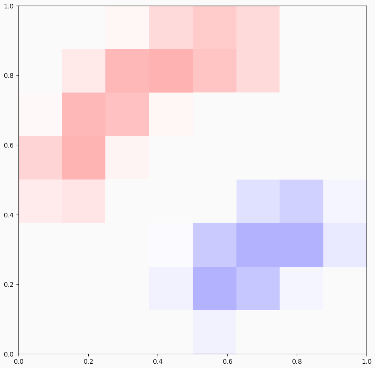
Encoding unlabeled shapes as measures

Let's enforce sampling invariance:

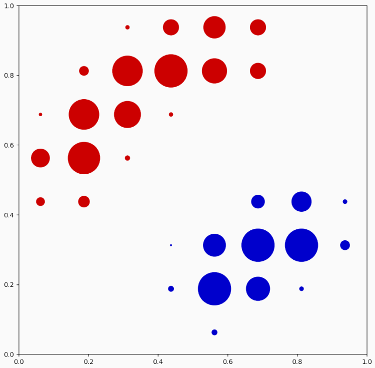
$$A \longrightarrow \alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}, \qquad B \longrightarrow \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}.$$



A baseline setting: density registration

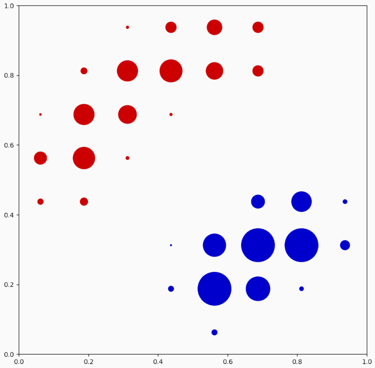


A baseline setting: density registration



$$\alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}\,, \quad \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}\,.$$

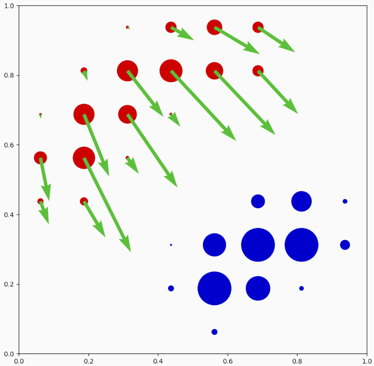
A baseline setting: density registration



$$\alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}\,, \quad \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}\,.$$

$$\sum_{i=1}^{N} \alpha_i \;=\; 1 \;=\; \sum_{j=1}^{M} \beta_j$$

A baseline setting: density registration

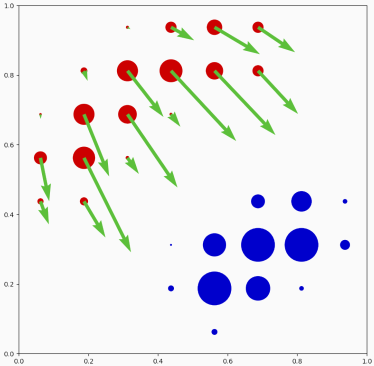


$$\alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}\,, \quad \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}\,.$$

$$\sum_{i=1}^{N} \alpha_i \;=\; 1 \;=\; \sum_{j=1}^{M} \beta_j$$

Display $\;v \;=\; -\nabla_{x_i}\text{Loss}(\alpha, \beta).$

A baseline setting: density registration



$$\alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}\,, \quad \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}\,.$$

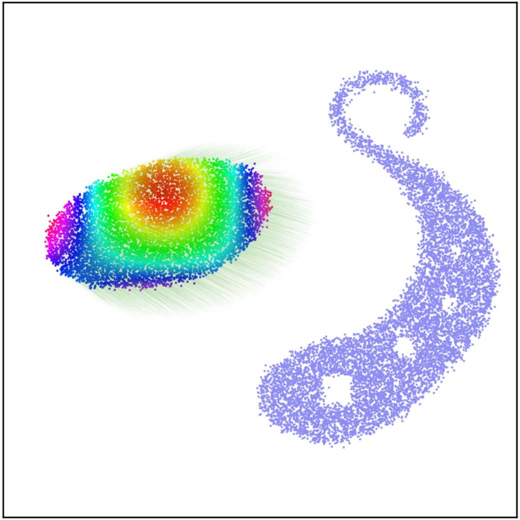
$$\sum_{i=1}^{N} \alpha_i \;=\; 1 \;=\; \sum_{j=1}^{M} \beta_j$$

Display $\;v \;=\; -\nabla_{x_i}\text{Loss}(\alpha, \beta)$.

Seamless extensions to:

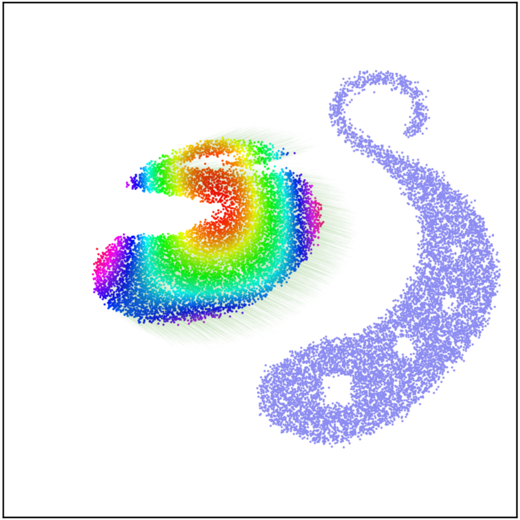
- $\sum_i \alpha_i \neq \sum_j \beta_j$, outliers [Chizat et al., 2018],
- curves and surfaces [Kaltenmark et al., 2017],
- variable weights α_i.

Gradient flow as a toy registration problem



$t = .00$

Gradient flow as a toy registration problem

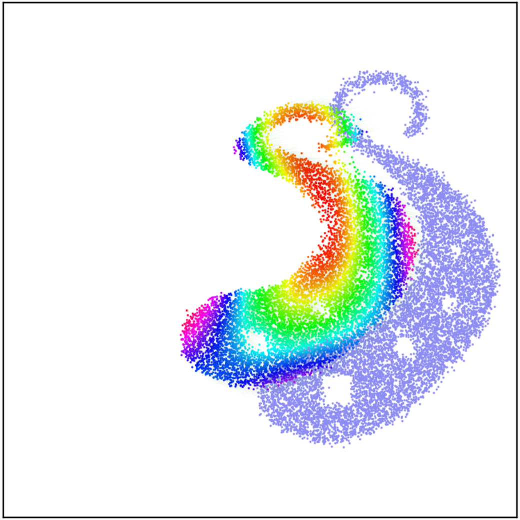


$t = .25$

Gradient flow as a toy registration problem

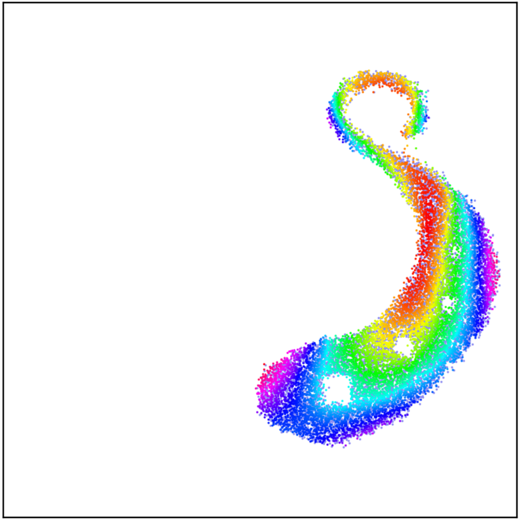
$t = .50$

Gradient flow as a toy registration problem



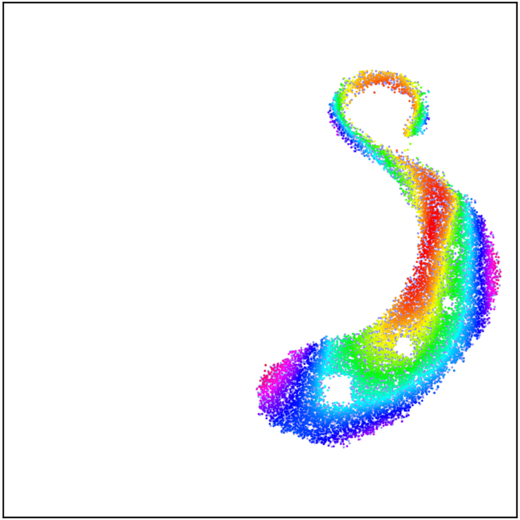
$t = 1.00$

Gradient flow as a toy registration problem



$t = 5.00$

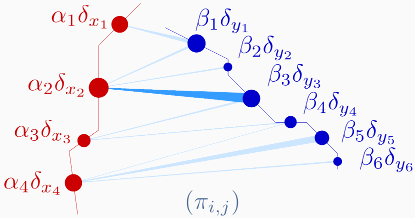
Gradient flow as a toy registration problem

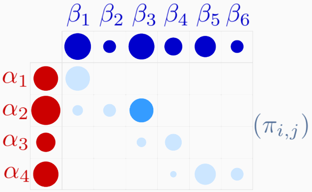


$t = 10.00$

The Wasserstein distance is a convenient baseline…
But will it scale to 3D meshes?

Introducing the Optimal Transport problem





Minimize over N-by-M matrices (transport plans) π :

$$\mathrm{OT}(\alpha, \beta) = \min_{\pi} \underbrace{\sum_{i,j} \pi_{i,j} \cdot \tfrac{1}{2}|x_i - y_j|^2}_{\text{transport cost}}$$

subject to $\quad \pi_{i,j} \geqslant 0,$

$$\sum_j \pi_{i,j} = \alpha_i, \quad \sum_i \pi_{i,j} = \beta_j.$$

Kantorovitch's dual formulation

$$\mathrm{OT}(\alpha, \beta) = \min_{\pi} \langle \pi , \mathrm{C} \rangle, \text{ with } \mathrm{C}(x_i, y_j) = \tfrac{1}{p}\|x_i - y_j\|^p \longrightarrow \text{Assignment}$$

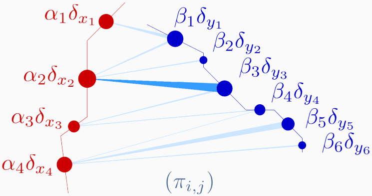
$$\text{s.t. } \pi \geqslant 0, \quad \pi \mathbf{1} = \alpha, \quad \pi^\intercal \mathbf{1} = \beta$$

Kantorovitch's dual formulation

$$\mathrm{OT}(\alpha, \beta) = \min_{\pi} \langle \pi , \mathrm{C} \rangle, \text{ with } \mathrm{C}(x_i, y_j) = \frac{1}{p}\|x_i - y_j\|^p \longrightarrow \text{Assignment}$$

$$\text{s.t. } \pi \geqslant 0, \quad \pi \mathbf{1} = \alpha, \quad \pi^\top \mathbf{1} = \beta$$

$\sum_{i,j} \pi_{i,j} \, \mathrm{C}(x_i, y_j)$

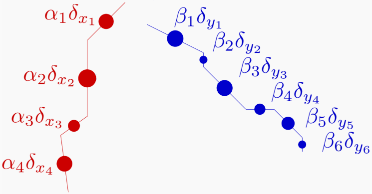


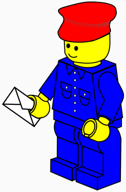
Kantorovitch's dual formulation

$$\mathrm{OT}(\alpha,\beta) = \min_{\pi} \langle \pi, \mathrm{C} \rangle, \text{ with } \mathrm{C}(x_i, y_j) = \frac{1}{p}\|x_i - y_j\|^p \longrightarrow \text{Assignment}$$

$$\text{s.t. } \pi \geqslant 0, \quad \pi\,\mathbf{1} = \alpha, \quad \pi^{\intercal}\mathbf{1} = \beta$$

$\sum_{i,j} \pi_{i,j}\, \mathrm{C}(x_i, y_j)$



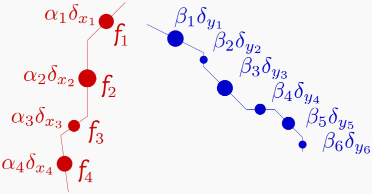


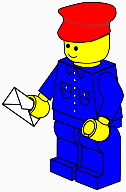
Kantorovitch's dual formulation

$$\mathrm{OT}(\alpha, \beta) = \min_{\pi} \langle \pi , \mathrm{C} \rangle, \text{ with } \mathrm{C}(x_i, y_j) = \frac{1}{p}\|x_i - y_j\|^p \longrightarrow \text{Assignment}$$

$$\text{s.t. } \pi \geqslant 0, \quad \pi \mathbf{1} = \alpha, \quad \pi^\top \mathbf{1} = \beta$$

$\sum_{i,j} \pi_{i,j} \, \mathrm{C}(x_i, y_j)$



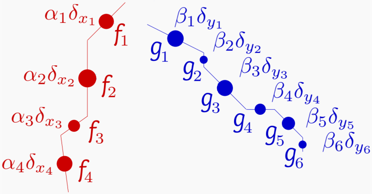


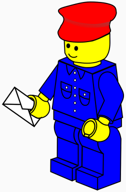
Kantorovitch's dual formulation

$$\mathrm{OT}(\alpha, \beta) = \min_{\pi} \langle \pi , \mathrm{C} \rangle, \text{ with } \mathrm{C}(x_i, y_j) = \frac{1}{p}\|x_i - y_j\|^p \longrightarrow \text{Assignment}$$

$$\text{s.t. } \pi \geqslant 0, \quad \pi \mathbf{1} = \alpha, \quad \pi^{\intercal} \mathbf{1} = \beta$$

$\sum_{i,j} \pi_{i,j} \, \mathrm{C}(x_i, y_j)$



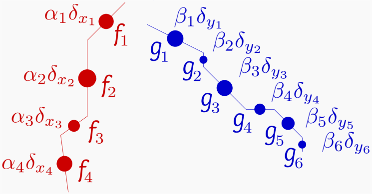


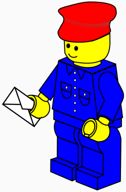
Kantorovitch's dual formulation

$$\mathrm{OT}(\alpha, \beta) = \min_{\pi} \langle \pi , \mathrm{C} \rangle, \text{ with } \mathrm{C}(x_i, y_j) = \frac{1}{p}\|x_i - y_j\|^p \longrightarrow \text{Assignment}$$

$$\text{s.t. } \pi \geqslant 0, \quad \pi \mathbf{1} = \alpha, \quad \pi^\top \mathbf{1} = \beta$$

$\sum_{i,j} \pi_{i,j} \mathrm{C}(x_i, y_j)$





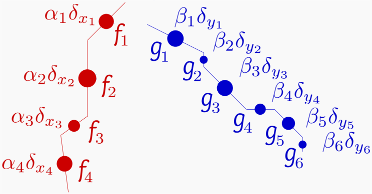
$\sum_i \alpha_i f_i + \sum_j \beta_j g_j$

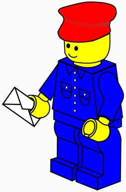
Kantorovitch's dual formulation

$$\mathrm{OT}(\alpha, \beta) \;=\; \min_{\pi} \langle \pi \,,\, \mathrm{C} \rangle, \;\text{ with } \mathrm{C}(x_i, y_j) \;=\; \tfrac{1}{p}\|x_i - y_j\|^p \quad \longrightarrow \text{ Assignment}$$

$$\text{s.t. } \pi \geqslant 0, \quad \pi\,\mathbf{1} = \alpha, \quad \pi^\top \mathbf{1} = \beta$$

$\sum_{i,j} \pi_{i,j}\, \mathrm{C}(x_i, y_j)$





$\sum_i \alpha_i f_i + \sum_j \beta_j g_j$

$$\max_{f,g} \quad \langle \alpha \,,\, f \rangle + \langle \beta \,,\, g \rangle \qquad \longrightarrow \text{ FedEx}$$

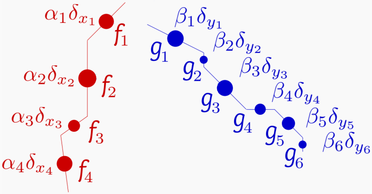
$$\text{s.t.} \quad f(x_i) + g(y_j) \leqslant \mathrm{C}(x_i, y_j),$$

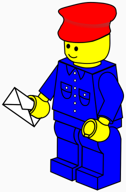
Kantorovitch's dual formulation

$$\mathrm{OT}(\alpha, \beta) \;=\; \min_{\pi} \langle \pi \,,\, \mathrm{C} \rangle, \;\; \text{with } \mathrm{C}(x_i, y_j) \;=\; \tfrac{1}{p}\|x_i - y_j\|^p \quad \longrightarrow \text{Assignment}$$

$$\text{s.t. } \pi \;\geqslant\; 0, \quad \pi\,\mathbf{1} \;=\; \alpha, \quad \pi^\top \mathbf{1} \;=\; \beta$$

$\sum_{i,j} \pi_{i,j}\, \mathrm{C}(x_i, y_j)$





$\sum_i \alpha_i f_i + \sum_j \beta_j g_j$

$$= \max_{f, g} \quad \langle \alpha \,,\, f \rangle \,+\, \langle \beta \,,\, g \rangle \qquad \longrightarrow \text{FedEx}$$

$$\text{s.t.} \quad f(x_i) \,+\, g(y_j) \;\leqslant\; \mathrm{C}(x_i, y_j),$$

How should we solve the dual problem?

Combinatorial, on the simplex $\implies$ Hungarian method in $O(N^3)$.

How should we solve the dual problem?

Combinatorial, on the simplex $\Longrightarrow$ Hungarian method in $\mathbf{O(N^3)}$.

$\alpha, \beta \geqslant 0,$ **separable** constraint $f(x_i) + g(y_j) \leqslant C(x_i, y_j)$:
Couldn't we **maximize the prices** f and g alternatively?

How should we solve the dual problem?

Combinatorial, on the simplex $\Longrightarrow$ Hungarian method in $\mathbf{O}(\mathbf{N}^3)$.

$\alpha, \beta \geqslant 0$, **separable** constraint $f(x_i) + g(y_j) \leqslant C(x_i, y_j)$:
Couldn't we **maximize the prices** f and g alternatively?

$$f_i \leftarrow \mathbf{0}_{\mathbb{R}^N} \; ; \quad g_j \leftarrow \mathbf{0}_{\mathbb{R}^M}$$

How should we solve the dual problem?

Combinatorial, on the simplex $\Longrightarrow$ Hungarian method in $\mathbf{O(N^3)}$.

$\alpha, \beta \geqslant 0$, **separable** constraint $f(x_i) + g(y_j) \leqslant C(x_i, y_j)$:
Couldn't we **maximize the prices** f and g alternatively?

$f_i \leftarrow \mathbf{0}_{\mathbb{R}^N}$; $g_j \leftarrow \mathbf{0}_{\mathbb{R}^M}$

Until convergence:

How should we solve the dual problem?

Combinatorial, on the simplex $\Longrightarrow$ Hungarian method in $\mathbf{O}(\mathbf{N}^3)$.

$\alpha,\ \beta\ \geqslant\ 0,$ **separable** constraint $f(x_i) + g(y_j) \leqslant C(x_i, y_j)$:
Couldn't we **maximize the prices** f and g alternatively?

$$f_i \leftarrow \mathbf{0}_{\mathbb{R}^N}\ ; \quad g_j \leftarrow \mathbf{0}_{\mathbb{R}^M}$$

Until convergence:

$$f_i = f(x_i) \leftarrow \min_{y_j \in \mathrm{Supp}(\beta)} \big[\, C(x_i, y_j) - g(y_j) \,\big]$$

How should we solve the dual problem?

Combinatorial, on the simplex $\Longrightarrow$ Hungarian method in $\mathbf{O}(\mathbf{N}^3)$.

$\alpha, \beta \geqslant 0$, **separable** constraint $f(x_i) + g(y_j) \leqslant C(x_i, y_j)$:
Couldn't we **maximize the prices** f and g alternatively?

$$f_i \leftarrow \mathbf{0}_{\mathbb{R}^N} \; ; \quad g_j \leftarrow \mathbf{0}_{\mathbb{R}^M}$$

Until convergence:

$$f_i = f(x_i) \leftarrow \min_{y_j \in \text{Supp}(\beta)} \big[\, C(x_i, y_j) - g(y_j) \,\big]$$
$$g_j = g(y_j) \leftarrow \min_{x_i \in \text{Supp}(\alpha)} \big[\, C(x_i, y_j) - f(x_i) \,\big]$$

How should we solve the dual problem?

Combinatorial, on the simplex $\Longrightarrow$ Hungarian method in $\mathbf{O(N^3)}$.

$\alpha, \beta \geqslant 0$, **separable** constraint $f(x_i) + g(y_j) \leqslant C(x_i, y_j)$:
Couldn't we **maximize the prices** f and g alternatively?

$$f_i \leftarrow \mathbf{0}_{\mathbb{R}^N} \; ; \quad g_j \leftarrow \mathbf{0}_{\mathbb{R}^M}$$

Until convergence:

$$f_i = f(x_i) \leftarrow \min_{y_j \in \text{Supp}(\beta)} \big[C(x_i, y_j) - g(y_j) \big]$$
$$g_j = g(y_j) \leftarrow \min_{x_i \in \text{Supp}(\alpha)} \big[C(x_i, y_j) - f(x_i) \big]$$

$\Longrightarrow$ Too **greedy**! We **get stuck** after two iterations.

An answer from Operational Research

Auction algorithm (Dimitri Bertsekas, 1980's):

$$f_i \leftarrow \mathbf{0}_{\mathbb{R}^N} \; ; \quad g_j \leftarrow \mathbf{0}_{\mathbb{R}^M}$$

Until convergence:

$$f_i = f(x_i) \leftarrow \min_{y_j \in \text{Supp}(\beta)} \big[\, C(x_i, y_j) - g(y_j) \, \big]$$

$$g_j = g(y_j) \leftarrow \min_{x_i \in \text{Supp}(\alpha)} \big[\, C(x_i, y_j) - f(x_i) \, \big] \text{ “} - \varepsilon \text{”}$$

An answer from Operational Research

Auction algorithm (Dimitri Bertsekas, 1980's):

$$f_i \leftarrow 0_{\mathbb{R}^N} \;; \quad g_j \leftarrow 0_{\mathbb{R}^M}$$

Until convergence:

$$f_i = f(x_i) \;\leftarrow\; \min_{y_j \in \mathrm{Supp}(\beta)} \big[\, C(x_i, y_j) - g(y_j) \,\big]$$

$$g_j = g(y_j) \;\leftarrow\; \min_{x_i \in \mathrm{Supp}(\alpha)} \big[\, C(x_i, y_j) - f(x_i) \,\big] \text{ “} - \varepsilon \text{”}$$

$\Longrightarrow$ ε-optimal solutions in $\mathbf{O}(N^2 \cdot \max_{\alpha \otimes \beta} C \,/\, \varepsilon)$.

An answer from Operational Research

Auction algorithm (Dimitri Bertsekas, 1980's):

$$f_i \leftarrow 0_{\mathbb{R}^N} \; ; \quad g_j \leftarrow 0_{\mathbb{R}^M}$$

Until convergence:

$$f_i = f(x_i) \leftarrow \min_{y_j \in \text{Supp}(\beta)} \big[C(x_i, y_j) - g(y_j) \big]$$

$$g_j = g(y_j) \leftarrow \min_{x_i \in \text{Supp}(\alpha)} \big[C(x_i, y_j) - f(x_i) \big] \text{ “}- \varepsilon\text{”}$$

$\Longrightarrow$ ε-optimal solutions in $\mathbf{O}(\mathbf{N}^2 \cdot \max_{\alpha \otimes \beta} C \, / \, \varepsilon)$.

$\Longrightarrow$ What about our **weights** α and β ?

$\Longrightarrow$ Can we symmetrize all this?

The SoftMin interpolates between a minimum and a sum

$$\log(e^c + e^d) = \max(c, d) + \log \big(\underbrace{e^{c-\max(c,d)} + e^{d-\max(c,d)}}_{\in [1,2]} \big)$$

The SoftMin interpolates between a minimum and a sum

$$\log(\, e^c \,+\, e^d \,) \;=\; \max(c,d) \,+\, \log\big(\underbrace{e^{c-\max(c,d)} + e^{d-\max(c,d)}}_{\in[1,2]}\big)$$

Building on this, for a **regularization** parameter $\varepsilon > 0$, we define

$$\min_{\varepsilon,\, y\sim\beta} \varphi(x,y) \;=\; -\varepsilon\,\log\sum_{j=1}^{M} \beta_j \,\exp\big[-\tfrac{1}{\varepsilon}\varphi(\,x, y_j\,)\big]$$

The SoftMin interpolates between a minimum and a sum

$$\log(e^c + e^d) = \max(c,d) + \log\big(\underbrace{e^{c-\max(c,d)} + e^{d-\max(c,d)}}_{\in [1,2]} \big)$$

Building on this, for a **regularization** parameter $\varepsilon > 0$, we define

$$\min_{\varepsilon,\, y\sim\beta} \varphi(x,y) = -\varepsilon \log \sum_{j=1}^{M} \beta_j \exp\big[-\tfrac{1}{\varepsilon}\varphi(x, y_j)\big]$$

The **IPFP–SoftAssign–Sinkhorn** algorithm:

$$f_i \leftarrow 0_{\mathbb{R}^N} ; \quad g_j \leftarrow 0_{\mathbb{R}^M}$$

Until convergence:

$$f_i = f(x_i) \leftarrow \min_{\varepsilon,\, y\sim\beta} \big[C(x_i,y_j) - g(y_j) \big]$$

$$g_j = g(y_j) \leftarrow \min_{\varepsilon,\, x\sim\alpha} \big[C(x_i,y_j) - f(x_i) \big]$$

The SoftMin interpolates between a minimum and a sum

$$\log(\, e^c \, + \, e^d \,) \;=\; \max(c,d) \, + \, \log\big(\underbrace{e^{c-\max(c,d)} \, + \, e^{d-\max(c,d)}}_{\in\,[1,2]}\big)$$

Building on this, for a **regularization** parameter $\varepsilon > 0$, we define

$$\min_{\varepsilon,\, y\sim\beta} \varphi(x,y) \;=\; -\varepsilon \, \log \sum_{j=1}^{M} \beta_j \, \exp\big[-\tfrac{1}{\varepsilon}\varphi(\,x,\, y_j\,)\big]$$

The **IPFP–SoftAssign–Sinkhorn** algorithm:

$$f_i \;\leftarrow\; 0_{\mathbb{R}^N} \;; \qquad g_j \;\leftarrow\; 0_{\mathbb{R}^M}$$

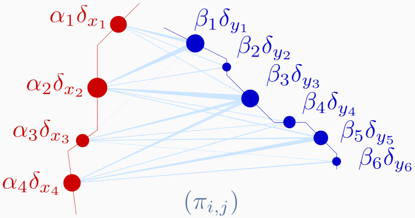
Until convergence:

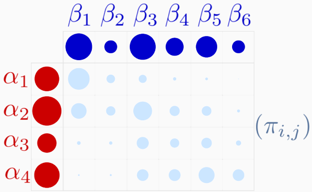
$$f_i \;=\; f(x_i) \;\;\leftarrow\;\; \min_{\varepsilon,\, y\sim\beta}\big[\, C(x_i,y_j) \;-\; g(y_j)\,\big]$$

$$g_j \;=\; g(y_j) \;\leftarrow\; \min_{\varepsilon,\, x\sim\alpha}\big[\, C(x_i,y_j) \;-\; f(x_i)\,\big]$$

$\Longrightarrow$ This **simple** algorithm works well!

Entropic regularization: introducing Schrödinger's problem





For $\varepsilon > 0$:

$$\mathrm{OT}_\varepsilon(\alpha, \beta) = \min_\pi \underbrace{\sum_{i,j} \pi_{i,j} \cdot \tfrac{1}{2}|x_i - y_j|^2}_{\text{transport cost}}$$

$$+ \; \varepsilon \underbrace{\sum_{i,j} \pi_{i,j} \cdot \log \frac{\pi_{i,j}}{\alpha_i \beta_j}}_{\text{entropic barrier}}$$

subject to

$$\sum_j \pi_{i,j} = \alpha_i, \quad \sum_i \pi_{i,j} = \beta_j.$$

Fenchel-Rockafellar to the rescue

$$\mathrm{OT}_\varepsilon(\alpha, \beta) \;=\; \min_{\pi} \,\langle\, \pi \,,\, \mathrm{C} \,\rangle \;+\; \varepsilon\, \mathrm{KL}(\, \pi \,,\, \alpha \otimes \beta \,) \quad \longrightarrow \text{Fuzzy assignment}$$

$$\text{s.t.} \qquad \pi\, \mathbf{1} \;=\; \alpha, \qquad \pi^\intercal \mathbf{1} \;=\; \beta$$

Fenchel-Rockafellar to the rescue

$$\mathrm{OT}_\varepsilon(\alpha,\beta) = \min_{\pi} \langle \pi , \mathrm{C} \rangle + \varepsilon\,\mathrm{KL}(\pi , \alpha\otimes\beta) \quad \longrightarrow \text{Fuzzy assignment}$$

$$\text{s.t.} \quad \pi\,\mathbf{1} = \alpha, \qquad \pi^{\mathsf{T}}\mathbf{1} = \beta$$

$$= \max_{f,g} \langle \alpha , f \rangle + \langle \beta , g \rangle \quad \longrightarrow \text{Cheeky FedEx}$$

$$- \underbrace{\varepsilon \langle \alpha\otimes\beta , e^{(f\oplus g - \mathrm{C})/\varepsilon} - 1 \rangle}_{\text{soft constraint } f\oplus g \leqslant \mathrm{C}}$$

Fenchel-Rockafellar to the rescue

$$\mathrm{OT}_\varepsilon(\alpha, \beta) = \min_{\pi} \langle \pi , \mathrm{C} \rangle + \varepsilon\, \mathrm{KL}(\pi , \alpha \otimes \beta) \quad \longrightarrow \text{Fuzzy assignment}$$

$$\text{s.t.} \qquad \pi \mathbf{1} = \alpha, \qquad \pi^\intercal \mathbf{1} = \beta$$

$$= \max_{f, g} \langle \alpha , f \rangle + \langle \beta , g \rangle \quad \longrightarrow \text{Cheeky FedEx}$$

$$- \underbrace{\varepsilon \langle \alpha \otimes \beta , e^{(f \oplus g - \mathrm{C})/\varepsilon} - 1 \rangle}_{\text{soft constraint } f \oplus g \leqslant \mathrm{C}}$$

At the optimum, $\pi = e^{(f \oplus g - \mathrm{C})/\varepsilon} \cdot \alpha \otimes \beta$

i.e. $\pi_{i,j} = \alpha_i \, e^{f_i/\varepsilon} \, e^{-\mathrm{C}(x_i, y_j)/\varepsilon} \, e^{g_j/\varepsilon} \, \beta_j.$

Sinkhorn algorithm = coordinate ascent on the dual problem

$$\mathrm{OT}_\varepsilon(\alpha, \beta) = \max_{f,g} \langle \alpha , f \rangle + \langle \beta , g \rangle \qquad \longrightarrow \text{Cheeky FedEx}$$

$$- \underbrace{\varepsilon \langle \alpha \otimes \beta , e^{(f \oplus g - \mathrm{C})/\varepsilon} - 1 \rangle}_{\text{soft constraint } f \oplus g \leqslant \mathrm{C}}$$

Sinkhorn algorithm = coordinate ascent on the dual problem

$$\mathrm{OT}_\varepsilon(\alpha,\beta) \;=\; \max_{f,g} \langle \alpha , f \rangle + \langle \beta , g \rangle \qquad \longrightarrow \text{Cheeky FedEx}$$

$$- \underbrace{\varepsilon \langle \alpha \otimes \beta , e^{(f \oplus g - \mathrm{C})/\varepsilon} - 1 \rangle}_{\text{soft constraint } f \oplus g \leqslant \mathrm{C}}$$

Equivalent to the constraints on π, the optimality conditions read:

$$f(x) \;=\; \min_{\varepsilon \atop y\sim\beta} \big[\, \mathrm{C}(x,y) - g(y) \,\big],$$

$$g(y) \;=\; \min_{\varepsilon \atop x\sim\alpha} \big[\, \mathrm{C}(x,y) - f(x) \,\big].$$

Sinkhorn algorithm = coordinate ascent on the dual problem

$$\mathrm{OT}_\varepsilon(\alpha, \beta) \;=\; \max_{f,g} \langle \alpha , f \rangle + \langle \beta , g \rangle \qquad \longrightarrow \text{Cheeky FedEx}$$

$$- \underbrace{\varepsilon \langle \alpha \otimes \beta , e^{(f \oplus g - \mathrm{C})/\varepsilon} - 1 \rangle}_{\text{soft constraint } f \oplus g \leqslant \mathrm{C}}$$

Equivalent to the constraints on π, the optimality conditions read:

$$f(x) \;=\; \min_{y \sim \beta}{}_\varepsilon \left[\mathrm{C}(x,y) - g(y) \right],$$

$$g(y) \;=\; \min_{x \sim \alpha}{}_\varepsilon \left[\mathrm{C}(x,y) - f(x) \right].$$

Sinkhorn algorithm = coordinate ascent on the dual problem

$$\mathrm{OT}_\varepsilon(\alpha, \beta) \;=\; \max_{f,g} \langle \alpha , f \rangle + \langle \beta , g \rangle \qquad \longrightarrow \text{ Cheeky FedEx}$$

$$- \underbrace{\varepsilon \langle \alpha \otimes \beta , e^{(f \oplus g - \mathrm{C})/\varepsilon} - 1 \rangle}_{\text{soft constraint } f \oplus g \leqslant \mathrm{C}}$$

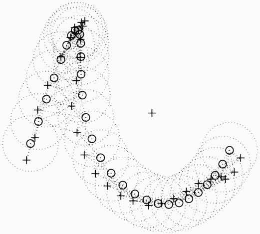
Equivalent to the constraints on π, the optimality conditions read:

$$f(x) \;=\; \min_{y \sim \beta}{}_{\varepsilon} \left[\mathrm{C}(x, y) - g(y) \right],$$

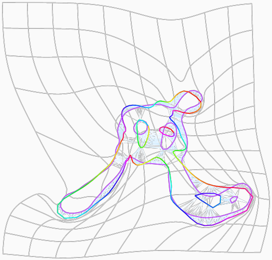
$$g(y) \;=\; \min_{x \sim \alpha}{}_{\varepsilon} \left[\mathrm{C}(x, y) - f(x) \right].$$

$\Longrightarrow$ Let's enforce them alternatively!

Re-inventing the wheel, every twenty years or so



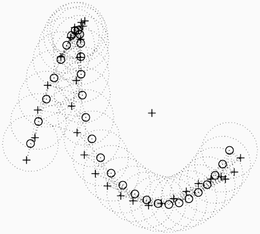
$\simeq$



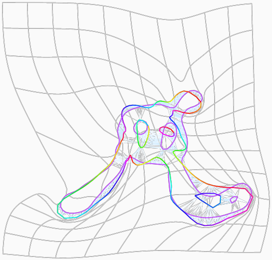
TPS-RPM algorithm,
Chui and Rangarajan, CVPR **2000**

Optimal Transport for diffeomorphic registration, Feydy et al., MICCAI **2017**

Re-inventing the wheel, every twenty years or so



$\simeq$



TPS-RPM algorithm,
Chui and Rangarajan, CVPR **2000**

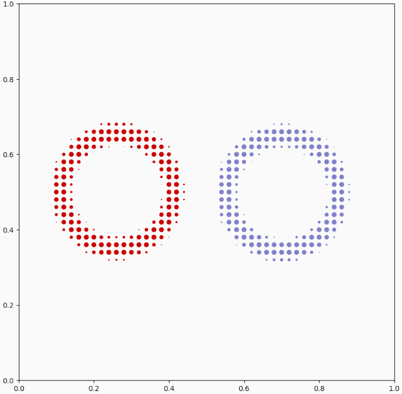
Optimal Transport for diffeomorphic registration, Feydy et al., MICCAI **2017**

$\Longrightarrow$ We've added weights, orientations, convergence analysis...
But shouldn't we go a bit **further**?

It's 2019 now:
What's new?

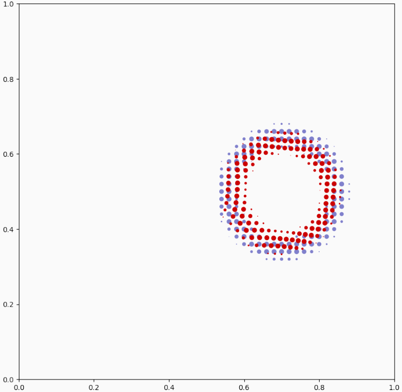
Unfortunately, if $\varepsilon > 0$, OT_ε is *not* a valid divergence

Registrating circles, $C(x,y) = \|x-y\|^2$, $\sqrt{\varepsilon} = 0.1$:



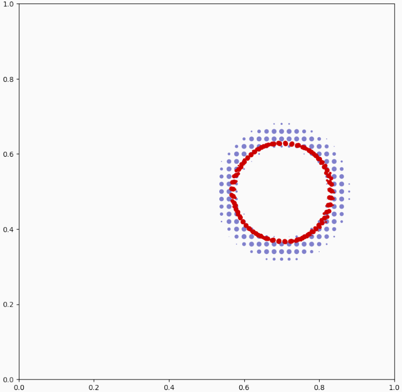
Unfortunately, if $\varepsilon > 0$, OT_ε is *not* a valid divergence

Registrating circles, $C(x, y) = \|x - y\|^2$, $\sqrt{\varepsilon} = 0.1$:



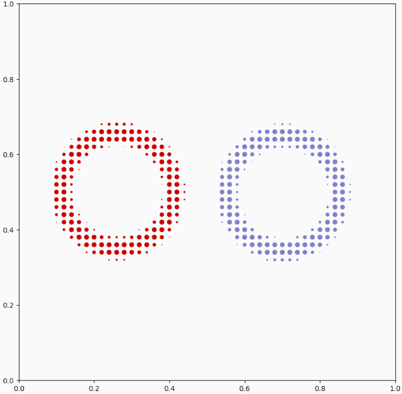
Unfortunately, if $\varepsilon > 0$, OT_ε is *not* a valid divergence

Registrating circles, $C(x, y) = \|x - y\|^2, \quad \sqrt{\varepsilon} = 0.1$:



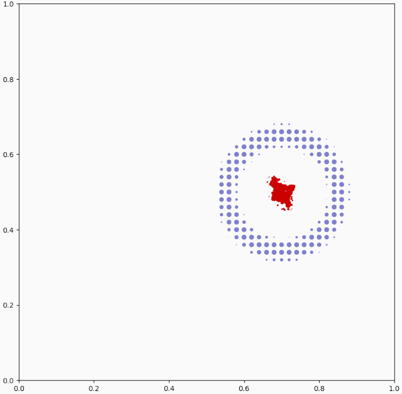
Unfortunately, if $\varepsilon > 0$, OT_ε is *not* a valid divergence

Registrating circles, $C(x, y) = \|x - y\|^2, \quad \sqrt{\varepsilon} = 0.2$:



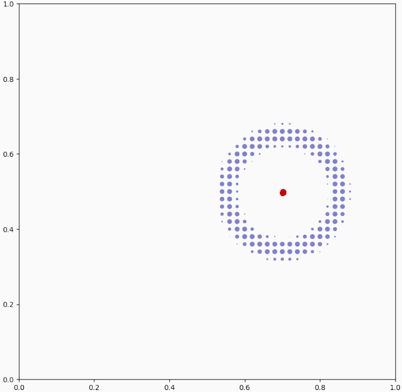
Unfortunately, if $\varepsilon > 0$, OT_ε is *not* a valid divergence

Registrating circles, $C(x, y) = \|x - y\|^2$, $\sqrt{\varepsilon} = 0.2$:



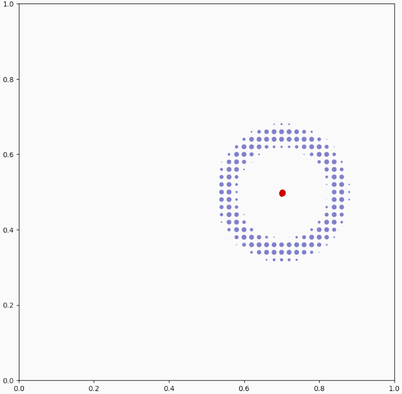
Unfortunately, if $\varepsilon > 0$, OT_ε is *not* a valid divergence

Registrating circles, $C(x, y) = \|x - y\|^2$, $\sqrt{\varepsilon} = 0.2$:



Unfortunately, if $\varepsilon > 0$, OT_ε is *not* a valid divergence

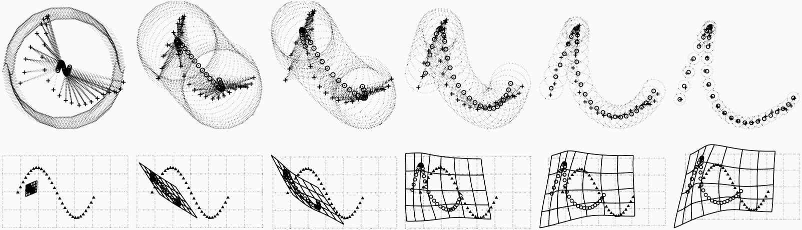
Registrating circles, $C(x, y) = \|x - y\|^2$, $\sqrt{\varepsilon} = 0.2$:



Bad news: for $0 < \varepsilon \leqslant +\infty$, we converge towards α such that

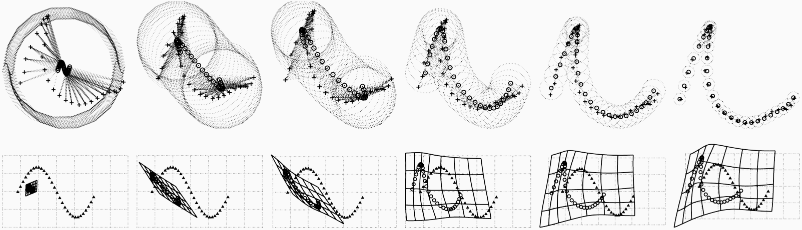
$$\mathrm{OT}_\varepsilon(\alpha, \beta) \;<\; \mathrm{OT}_\varepsilon(\beta, \beta).$$

Standard solution: use an annealing scheme in the descent



TPS-RPM algorithm, Chui and Rangarajan, CVPR 2000

Standard solution: use an annealing scheme in the descent



TPS-RPM algorithm, Chui and Rangarajan, CVPR 2000

$\Longrightarrow$ **Cumbersome** and **brittle** workaround,
with parameters to tune.

A new idea in 2015-17 : de-biased Sinkhorn divergences

$$\text{OT}_\varepsilon(\alpha, \beta) \;=\; \min_{\pi} \,\langle\, \pi \,,\, \text{C} \,\rangle \;+\; \varepsilon\, \text{KL}(\, \pi \,,\, \alpha \otimes \beta \,) \quad \longrightarrow \text{Fuzzy assignment}$$

$$\text{s.t.} \qquad \pi\, \mathbf{1} \;=\; \alpha, \qquad \pi^{\intercal} \mathbf{1} \;=\; \beta$$

A new idea in 2015-17 : de-biased Sinkhorn divergences

$$\mathrm{OT}_\varepsilon(\alpha, \beta) = \min_\pi \langle \pi , \mathrm{C} \rangle + \varepsilon \, \mathrm{KL}(\pi , \alpha \otimes \beta) \longrightarrow \text{Fuzzy assignment}$$

$$\text{s.t.} \quad \pi \mathbf{1} = \alpha, \quad \pi^\intercal \mathbf{1} = \beta$$

$$\mathrm{OT}_\varepsilon(\alpha, \beta) \xrightarrow{\varepsilon \to +\infty} \langle \alpha \otimes \beta , \mathrm{C} \rangle = \langle \alpha , \mathrm{C} \star \beta \rangle$$

A new idea in 2015-17 : de-biased Sinkhorn divergences

$$\text{OT}_\varepsilon(\alpha, \beta) = \min_\pi \langle \pi , \text{C} \rangle + \varepsilon \, \text{KL}(\pi , \alpha \otimes \beta) \quad \longrightarrow \text{Fuzzy assignment}$$

$$\text{s.t.} \qquad \pi \, \mathbf{1} = \alpha, \qquad \pi^\intercal \mathbf{1} = \beta$$

$$\text{OT}_\varepsilon(\alpha, \beta) \qquad \xrightarrow{\varepsilon \to +\infty} \qquad \langle \alpha \otimes \beta , \text{C} \rangle = \langle \alpha , \text{C} \star \beta \rangle$$

Define the **Sinkhorn divergence** [Ramdas et al., 2017]:

$$\text{S}_\varepsilon(\alpha, \beta) = \text{OT}_\varepsilon(\alpha, \beta) - \tfrac{1}{2}\text{OT}_\varepsilon(\alpha, \alpha) - \tfrac{1}{2}\text{OT}_\varepsilon(\beta, \beta)$$

A new idea in 2015-17 : de-biased Sinkhorn divergences

$$\mathrm{OT}_\varepsilon(\alpha,\beta) = \min_\pi \langle \pi , \mathrm{C} \rangle + \varepsilon\, \mathrm{KL}(\pi , \alpha \otimes \beta) \longrightarrow \text{Fuzzy assignment}$$

$$\text{s.t.} \quad \pi\,\mathbf{1} = \alpha, \qquad \pi^\intercal \mathbf{1} = \beta$$

$$\mathrm{OT}_\varepsilon(\alpha,\beta) \xrightarrow{\varepsilon \to +\infty} \langle \alpha \otimes \beta , \mathrm{C} \rangle = \langle \alpha , \mathrm{C} \star \beta \rangle$$

Define the **Sinkhorn divergence** [Ramdas et al., 2017]:

$$\mathrm{S}_\varepsilon(\alpha,\beta) = \mathrm{OT}_\varepsilon(\alpha,\beta) - \tfrac{1}{2}\mathrm{OT}_\varepsilon(\alpha,\alpha) - \tfrac{1}{2}\mathrm{OT}_\varepsilon(\beta,\beta)$$

$$\text{Wasserstein}_{+\mathrm{C}}(\alpha,\beta) \xleftarrow{\varepsilon \to 0} \mathrm{S}_\varepsilon(\alpha,\beta) \xrightarrow{\varepsilon \to +\infty} \tfrac{1}{2}\langle \alpha - \beta, -\mathrm{C} \star (\alpha - \beta) \rangle$$

A new idea in 2015-17 : de-biased Sinkhorn divergences

$$\mathrm{OT}_\varepsilon(\alpha,\beta) = \min_\pi \langle \pi , \mathrm{C} \rangle + \varepsilon \, \mathrm{KL}(\pi , \alpha\otimes\beta) \quad \longrightarrow \text{Fuzzy assignment}$$

$$\text{s.t.} \qquad \pi \mathbf{1} = \alpha, \qquad \pi^\intercal \mathbf{1} = \beta$$

$$\mathrm{OT}_\varepsilon(\alpha,\beta) \quad \xrightarrow{\varepsilon \to +\infty} \quad \langle \alpha\otimes\beta , \mathrm{C} \rangle = \langle \alpha , \mathrm{C} \star \beta \rangle$$

Define the **Sinkhorn divergence** [Ramdas et al., 2017]:

$$\mathrm{S}_\varepsilon(\alpha,\beta) = \mathrm{OT}_\varepsilon(\alpha,\beta) - \tfrac{1}{2}\mathrm{OT}_\varepsilon(\alpha,\alpha) - \tfrac{1}{2}\mathrm{OT}_\varepsilon(\beta,\beta)$$

$$\mathrm{Wasserstein}_{+\mathrm{C}}(\alpha,\beta) \xleftarrow{\varepsilon\to 0} \mathrm{S}_\varepsilon(\alpha,\beta) \xrightarrow{\varepsilon\to+\infty} \tfrac{1}{2}\langle \alpha-\beta, -\mathrm{C}\star(\alpha-\beta) \rangle$$

In practice, S_ε is “good enough” for ML applications [Genevay et al., 2018, Salimans et al., 2018, Sanjabi et al., 2018].

In our papers: theoretical guarantees

Theorem (F., Séjourné, Vialard, Amari, Trouvé, Peyré; 2018)

For all probability measures α, β *and regularization* $\varepsilon > 0$ *:*

In our papers: theoretical guarantees

Theorem (F., Séjourné, Vialard, Amari, Trouvé, Peyré; 2018)

For all probability measures α, β *and regularization* $\varepsilon > 0$ *:*

$$0 \leqslant S_\varepsilon(\alpha, \beta) \quad \text{with equality iff.} \quad \alpha = \beta$$

In our papers: theoretical guarantees

Theorem (F., Séjourné, Vialard, Amari, Trouvé, Peyré; 2018)

For all probability measures α, β *and regularization* $\varepsilon > 0$ *:*

$$0 \leqslant S_\varepsilon(\alpha, \beta) \quad \text{with equality iff.} \quad \alpha = \beta$$

$$\alpha \mapsto S_\varepsilon(\alpha, \beta) \quad \text{is convex and differentiable}$$

In our papers: theoretical guarantees

Theorem (F., Séjourné, Vialard, Amari, Trouvé, Peyré; 2018)

For all probability measures α, β *and regularization* $\varepsilon > 0$:

$$0 \leqslant S_\varepsilon(\alpha, \beta) \quad \text{with equality iff.} \quad \alpha = \beta$$

$$\alpha \mapsto S_\varepsilon(\alpha, \beta) \quad \text{is convex and differentiable}$$

These results can be generalized to ***unbalanced OT*** *and arbitrary* ***feature*** *spaces – e.g. (position, orientation)* $\simeq \mathbb{R}^3 \times \mathbb{S}^2$.

In our papers: theoretical guarantees

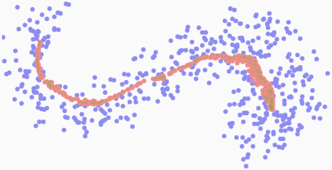
Theorem (F., Séjourné, Vialard, Amari, Trouvé, Peyré; 2018)

For all probability measures α, β and regularization $\varepsilon > 0$:

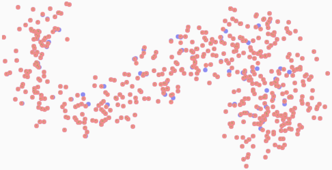
$$0 \leqslant S_\varepsilon(\alpha, \beta) \quad \textit{with equality iff.} \quad \alpha = \beta$$

$$\alpha \mapsto S_\varepsilon(\alpha, \beta) \quad \textit{is convex and differentiable}$$

*These results can be generalized to **unbalanced OT** and arbitrary **feature** spaces – e.g. (position, orientation)* $\simeq \mathbb{R}^3 \times \mathbb{S}^2$.



Loss $= \mathrm{OT}_\varepsilon$



Loss $= \mathrm{S}_\varepsilon$

Sinkhorn divergence $\iff$ de-biased entropic dual potentials

The optimality conditions read:

$$f(x) \;=\; \min_{y\sim\beta}{}_{\varepsilon} \left[\, C(x,y) \;-\; g(y) \,\right],$$

$$g(y) \;=\; \min_{x\sim\alpha}{}_{\varepsilon} \left[\, C(x,y) \;-\; f(x) \,\right].$$

Sinkhorn divergence $\iff$ de-biased entropic dual potentials

The optimality conditions read:

$$f(x) = \min_{y\sim\beta}{}_{\varepsilon} \big[\, C(x,y) - g(y) \,\big],$$

$$g(y) = \min_{x\sim\alpha}{}_{\varepsilon} \big[\, C(x,y) - f(x) \,\big].$$

Final cost:

$$\mathrm{OT}_\varepsilon(\alpha,\beta) = \langle\, \alpha \,,\, f \,\rangle + \langle\, \beta \,,\, g \,\rangle,$$

Sinkhorn divergence $\Longleftrightarrow$ de-biased entropic dual potentials

The optimality conditions read:

$$f(x) = \min_{y \sim \beta}{}_{\varepsilon} \left[\mathrm{C}(x,y) - g(y) \right],$$

$$g(y) = \min_{x \sim \alpha}{}_{\varepsilon} \left[\mathrm{C}(x,y) - f(x) \right].$$

Final cost:

$$\begin{aligned} \mathrm{OT}_\varepsilon(\alpha, \beta) &= \langle \alpha \,, f \rangle + \langle \beta \,, g \rangle, \\ \mathrm{S}_\varepsilon(\alpha, \beta) &= \mathrm{OT}_\varepsilon(\alpha, \beta) - \tfrac{1}{2}\mathrm{OT}_\varepsilon(\alpha, \alpha) - \tfrac{1}{2}\mathrm{OT}_\varepsilon(\beta, \beta) \\ &= \langle \alpha \,, \underbrace{f^{\beta \to \alpha} - f^{\alpha \leftrightarrow \alpha}}_{F} \rangle + \langle \beta \,, \underbrace{g^{\alpha \to \beta} - g^{\beta \leftrightarrow \beta}}_{G} \rangle. \end{aligned}$$

Sinkhorn divergence $\iff$ de-biased entropic dual potentials

The optimality conditions read:

$$f(x) \;=\; \min_{y\sim\beta}{}_{\varepsilon} \left[\, \mathrm{C}(x,y) \;-\; g(y) \,\right],$$

$$g(y) \;=\; \min_{x\sim\alpha}{}_{\varepsilon} \left[\, \mathrm{C}(x,y) \;-\; f(x) \,\right].$$

Final cost:

$$\begin{aligned}
\mathrm{OT}_\varepsilon(\alpha,\beta) \;&=\; \langle\, \alpha \,,\, f \,\rangle \;+\; \langle\, \beta \,,\, g \,\rangle, \\
\mathrm{S}_\varepsilon(\alpha,\beta) \;&=\; \mathrm{OT}_\varepsilon(\alpha,\beta) \;-\; \tfrac{1}{2}\mathrm{OT}_\varepsilon(\alpha,\alpha) \;-\; \tfrac{1}{2}\mathrm{OT}_\varepsilon(\beta,\beta) \\
&=\; \langle\, \alpha \,,\, \underbrace{f^{\beta\to\alpha} \;-\; f^{\alpha\leftrightarrow\alpha}}_{F} \,\rangle \;+\; \langle\, \beta \,,\, \underbrace{g^{\alpha\to\beta} \;-\; g^{\beta\leftrightarrow\beta}}_{G} \,\rangle.
\end{aligned}$$

Is **Sinkhorn** the optimal way of computing
the **de-biased** potentials F and G?

Use a *coarse-to-fine* strategy [Kosowsky and Yuille, 1994]

Dual $\text{OT}_\varepsilon(\alpha, \beta)$ problem: high-dimensional, concave maximization.

Use a *coarse-to-fine* strategy [Kosowsky and Yuille, 1994]

Dual $\mathrm{OT}_\varepsilon(\alpha, \beta)$ problem: high-dimensional, concave maximization.

Unfortunately, “standard” acceleration schemes are inefficient:
the gradient is **highly un-informative**.

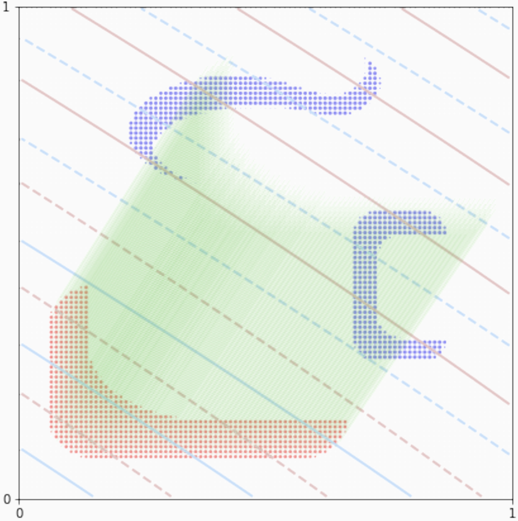
Use a *coarse-to-fine* strategy [Kosowsky and Yuille, 1994]

Dual $\text{OT}_\varepsilon(\alpha, \beta)$ problem: high-dimensional, concave maximization.

Unfortunately, "standard" acceleration schemes are inefficient:
the gradient is **highly un-informative**.

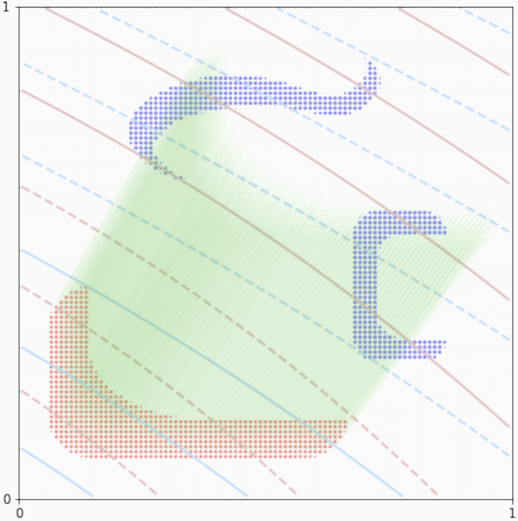
$\Longrightarrow$ **Simulated annealing:** let ε decrease across iterations,
to **leverage the structure of the problem**
in a **coarse-to-fine** fashion.

Visualizing F, G and the Brenier map $-\frac{1}{\alpha_i}\partial_{x_i}\mathrm{S}_\varepsilon(\alpha,\beta)$



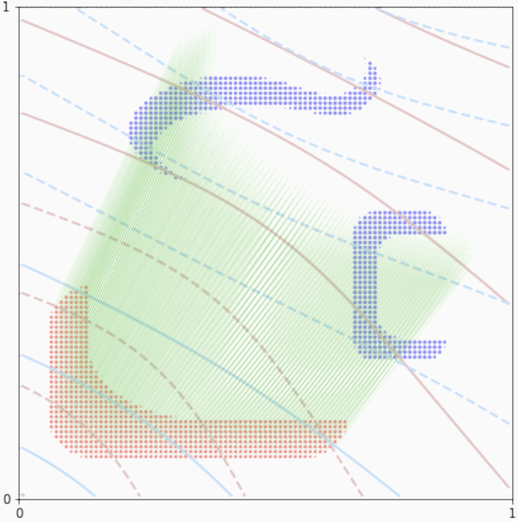
Iteration 0, $\sqrt{\varepsilon} = 2^0$

Visualizing F, G and the Brenier map $-\frac{1}{\alpha_i}\partial_{x_i}\mathrm{S}_\varepsilon(\alpha, \beta)$



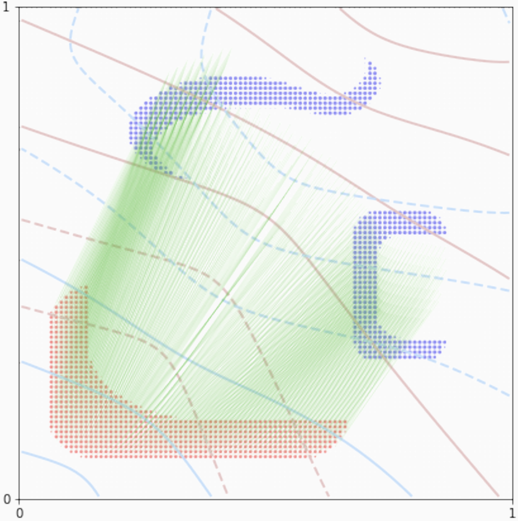
Iteration 1, $\sqrt{\varepsilon} = 2^{-1}$

Visualizing F, G and the Brenier map $-\frac{1}{\alpha_i}\partial_{x_i}\mathrm{S}_\varepsilon(\alpha,\beta)$



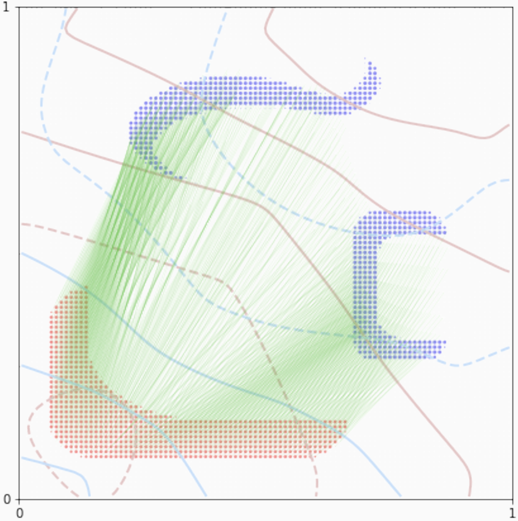
Iteration 2, $\sqrt{\varepsilon} = 2^{-2}$

Visualizing F, G and the Brenier map $-\frac{1}{\alpha_i}\partial_{x_i}\mathrm{S}_\varepsilon(\alpha, \beta)$



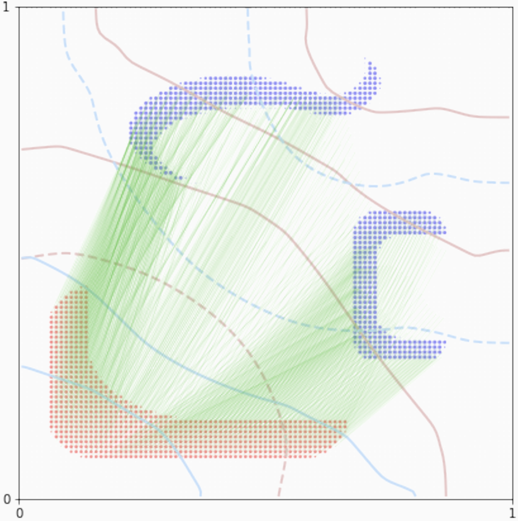
Iteration 3, $\sqrt{\varepsilon} = 2^{-3}$

Visualizing F, G and the Brenier map $-\frac{1}{\alpha_i}\partial_{x_i}\mathrm{S}_\varepsilon(\alpha,\beta)$



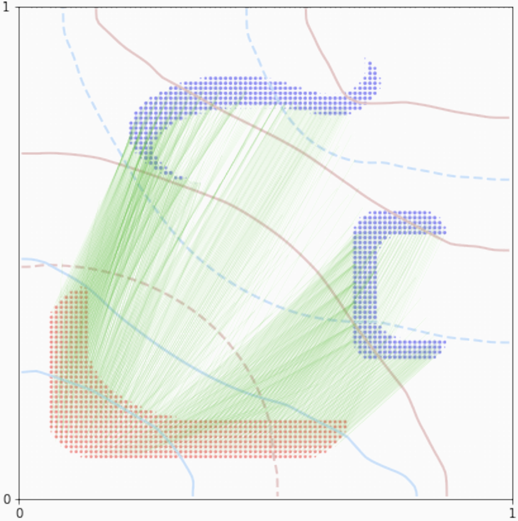
Iteration 4, $\sqrt{\varepsilon} = 2^{-4}$

Visualizing F, G and the Brenier map $-\frac{1}{\alpha_i}\partial_{x_i}\mathrm{S}_\varepsilon(\alpha, \beta)$



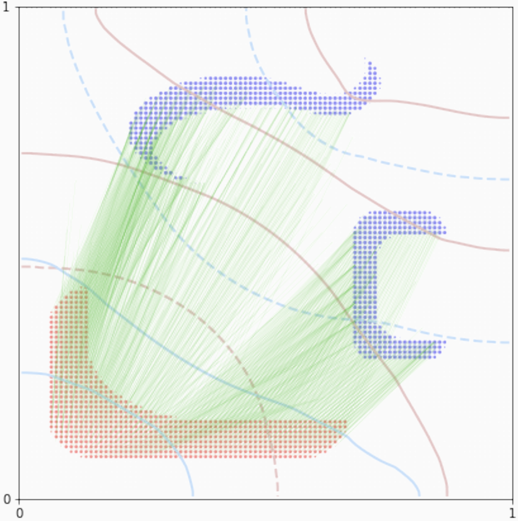
Iteration 5, $\sqrt{\varepsilon} = 2^{-5}$

Visualizing F, G and the Brenier map $-\frac{1}{\alpha_i}\partial_{x_i}\mathrm{S}_\varepsilon(\alpha,\beta)$



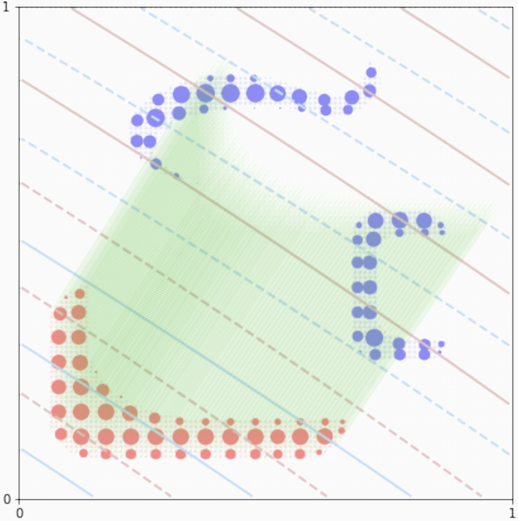
Iteration 6, $\sqrt{\varepsilon} = 2^{-6}$

Visualizing F, G and the Brenier map $-\frac{1}{\alpha_i}\partial_{x_i}\mathrm{S}_\varepsilon(\alpha,\beta)$

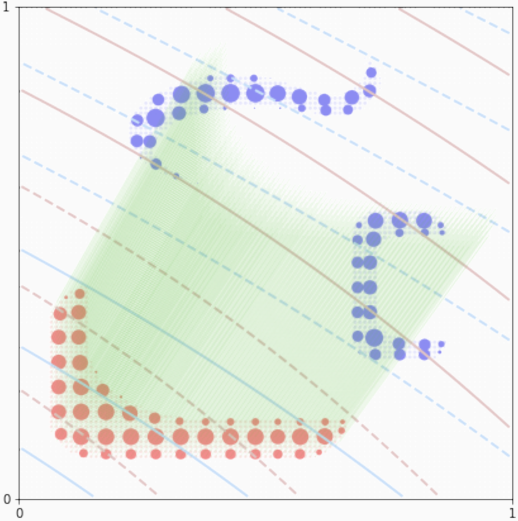


Iteration 7, $\sqrt{\varepsilon}$ = `.01`

Combining ε-scaling with a multiscale scheme [Schmitzer, 2016]

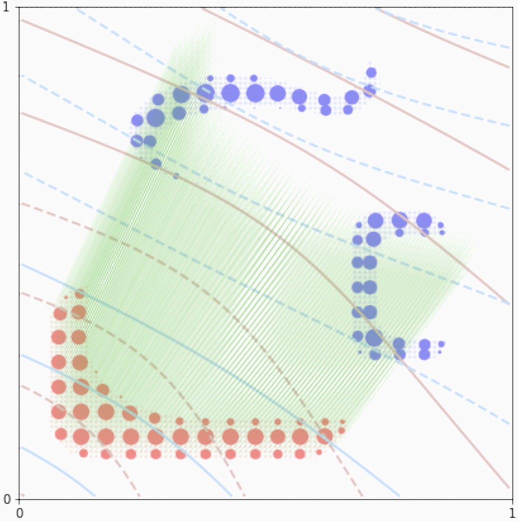


Iteration 0, $\sqrt{\varepsilon} = 2^0$



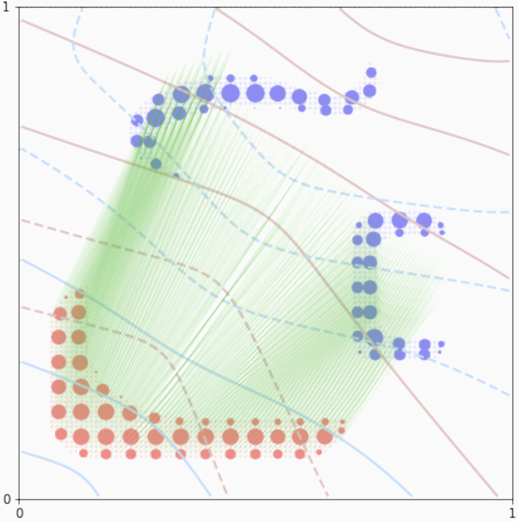
Iteration 1, $\sqrt{\varepsilon} = 2^{-1}$

Combining ε-scaling with a multiscale scheme [Schmitzer, 2016]



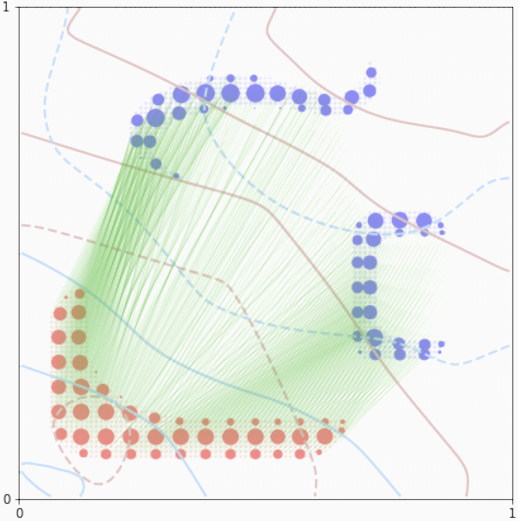
Iteration 2, $\sqrt{\varepsilon} = 2^{-2}$

Combining ε-scaling with a multiscale scheme [Schmitzer, 2016]



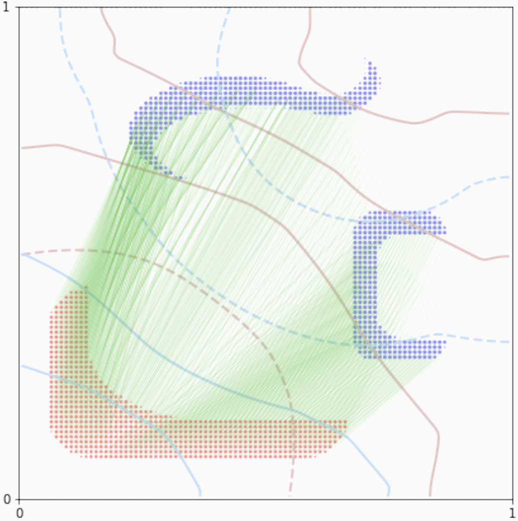
Iteration 3, $\sqrt{\varepsilon} = 2^{-3}$

Combining ε-scaling with a multiscale scheme [Schmitzer, 2016]



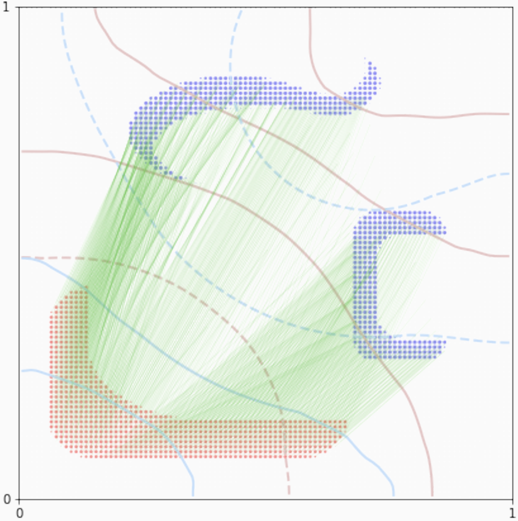
Iteration 4, $\sqrt{\varepsilon} = 2^{-4}$

Combining ε-scaling with a multiscale scheme [Schmitzer, 2016]



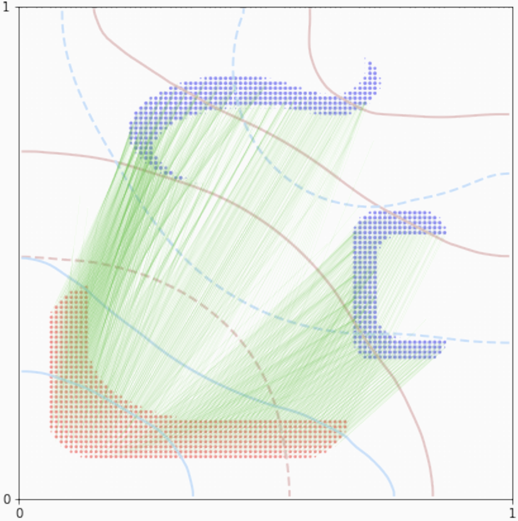
Iteration 5, $\sqrt{\varepsilon} = 2^{-5}$

Combining ε-scaling with a multiscale scheme [Schmitzer, 2016]



Iteration 6, $\sqrt{\varepsilon} = 2^{-6}$

Combining ε-scaling with a multiscale scheme [Schmitzer, 2016]



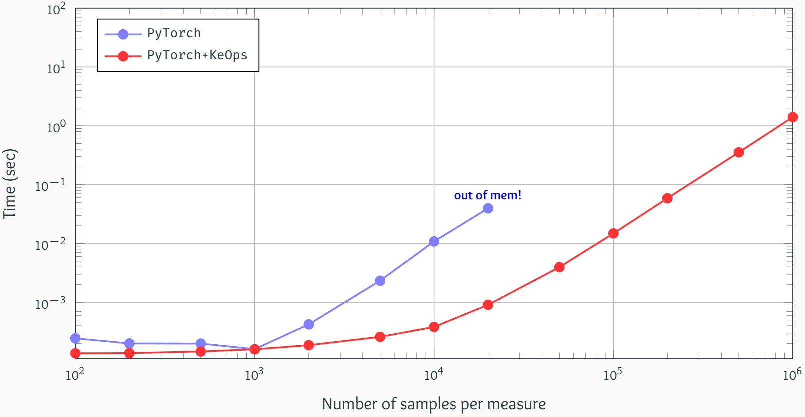
Iteration 7, $\sqrt{\varepsilon}$ = `.01`

GeomLoss: a new, super-fast GPU implementation

Leverages the **KeOps** library [Charlier, F., Glaunès, 2018]:

$\Longrightarrow$ `pip install pykeops` $\Longleftarrow$

Gaussian dot product in 3D (RTX 2080 GPU)

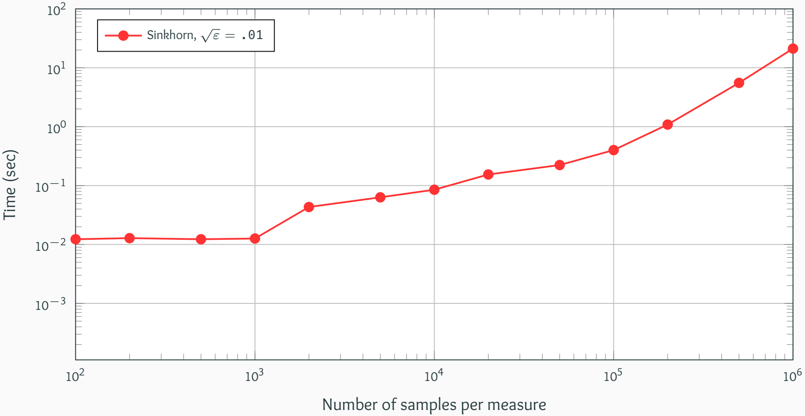


GeomLoss: a new, super-fast GPU implementation

Our website: `www.kernel-operations.io/geomloss`

$\Longrightarrow$ `pip install geomloss` $\Longleftarrow$

Loss + gradient in the unit cube, on **Google Colab** (Tesla T4 GPU)



Conclusion

Wasserstein distance = Multi-dimensional sorting problem ?

The **three regimes** of Optimal Transport:

Wasserstein distance = Multi-dimensional sorting problem ?

The **three regimes** of Optimal Transport:

- α, β live in **dimension 1**
  $\Longrightarrow$ Simple sorting problem
  $\Longrightarrow$ Quicksort in $\mathbf{O}(\mathbf{N} \log \mathbf{N})$.

Wasserstein distance = Multi-dimensional sorting problem ?

The **three regimes** of Optimal Transport:

- α, β live in **dimension 1**
  $\Longrightarrow$ Simple sorting problem
  $\Longrightarrow$ Quicksort in $\mathbf{O}(\mathbf{N}\log\mathbf{N})$.

- α, β live in dimension **10+**
  $\Longrightarrow$ $C(x_i, y_j)$ has very little structure
  $\Longrightarrow$ Compute all pairs in $\geqslant \mathbf{O}(\mathbf{N}^2)$.

Wasserstein distance = Multi-dimensional sorting problem ?

The **three regimes** of Optimal Transport:

- α, β live in **dimension 1**
  $\Longrightarrow$ Simple sorting problem
  $\Longrightarrow$ Quicksort in $\mathbf{O}(\mathbf{N} \log \mathbf{N})$.

- α, β have a **small** intrinsic **dimension**

- α, β live in dimension **10+**
  $\Longrightarrow$ $C(x_i, y_j)$ has very little structure
  $\Longrightarrow$ Compute all pairs in $\geqslant \mathbf{O}(\mathbf{N}^2)$.

Wasserstein distance = Multi-dimensional sorting problem ?

The **three regimes** of Optimal Transport:

- α, β live in **dimension 1**
  $\Longrightarrow$ Simple sorting problem
  $\Longrightarrow$ Quicksort in $\mathbf{O}(\mathbf{N} \log \mathbf{N})$.

- α, β have a **small** intrinsic **dimension**
  $\Longrightarrow$ Rely on multiscale strategies

- α, β live in dimension **10+**
  $\Longrightarrow$ $\text{C}(x_i, y_j)$ has very little structure
  $\Longrightarrow$ Compute all pairs in $\geqslant \mathbf{O}(\mathbf{N}^2)$.

Wasserstein distance = Multi-dimensional sorting problem ?

The **three regimes** of Optimal Transport:

- α, β live in **dimension 1**
  $\Longrightarrow$ Simple sorting problem
  $\Longrightarrow$ Quicksort in $\mathbf{O(N \log N)}$.

- α, β have a **small** intrinsic **dimension**
  $\Longrightarrow$ Rely on multiscale strategies
  $\Longrightarrow$ Multiscale Sinkhorn in $\mathbf{O(N \log N)}$ **on the GPU.**

- α, β live in dimension **10+**
  $\Longrightarrow$ $C(x_i, y_j)$ has very little structure
  $\Longrightarrow$ Compute all pairs in $\geqslant \mathbf{O(N^2)}$.

Wasserstein distance = Multi-dimensional sorting problem ?

The **three regimes** of Optimal Transport:

- α, β live in **dimension 1**
  $\Longrightarrow$ Simple sorting problem
  $\Longrightarrow$ Quicksort in $\mathbf{O(N \log N)}$.

- α, β have a **small** intrinsic **dimension**
  $\Longrightarrow$ Rely on multiscale strategies
  $\Longrightarrow$ Multiscale Sinkhorn in $\mathbf{O(N \log N)}$ **on the GPU.**

- α, β live in dimension **10+**
  $\Longrightarrow$ $C(x_i, y_j)$ has very little structure
  $\Longrightarrow$ Compute all pairs in $\geqslant \mathbf{O(N^2)}$.

$\Longrightarrow$ Multiscale Sinkhorn algorithm $\simeq$ Multi-dimensional **Quicksort.**

Key points

For **users**: reliable, efficient `python` toolboxes:

- Fluid mechanics: `github.com/sd-ot/pysdot`
- Machine Learning: `pot.readthedocs.io`
- Graphics, large-scale ML:
  `www.kernel-operations.io/geomloss`

Key points

For **users**: reliable, efficient `python` toolboxes:

- Fluid mechanics: `github.com/sd-ot/pysdot`
- Machine Learning: `pot.readthedocs.io`
- Graphics, large-scale ML:
  `www.kernel-operations.io/geomloss`

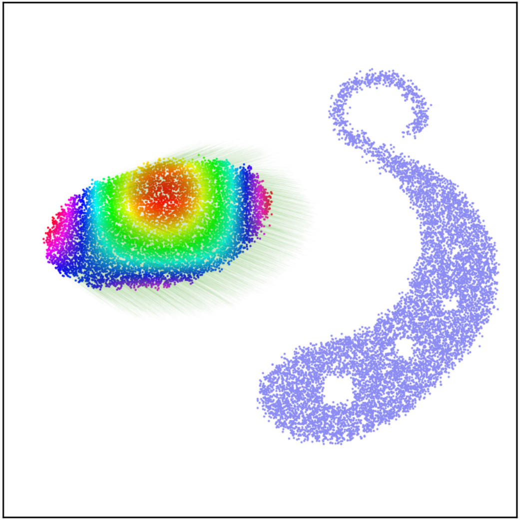
For **us**: new interesting questions:

- How should we quantify the **convergence** of ε-scaling?
- Link between S_ε and a **blurred Wasserstein** distance?

Thank you for your attention.

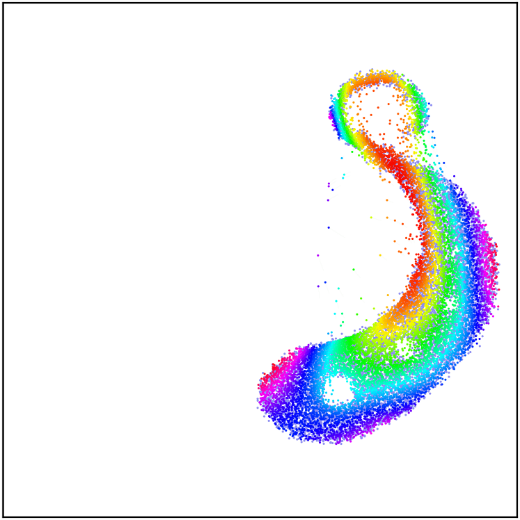
Any questions ?

Gradient descent on S_ε: cheap'n easy registration?



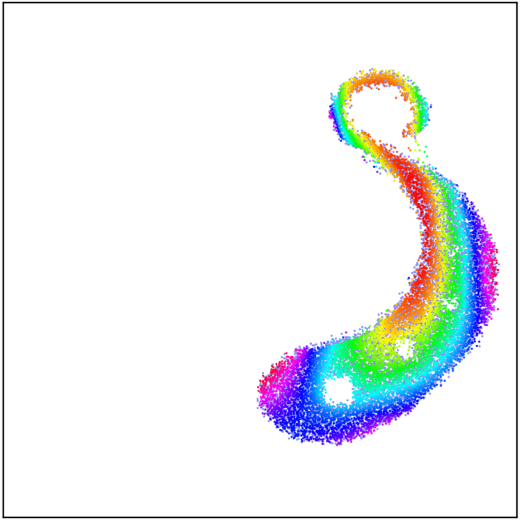
Iteration 0

Gradient descent on S_ε: cheap'n easy registration?



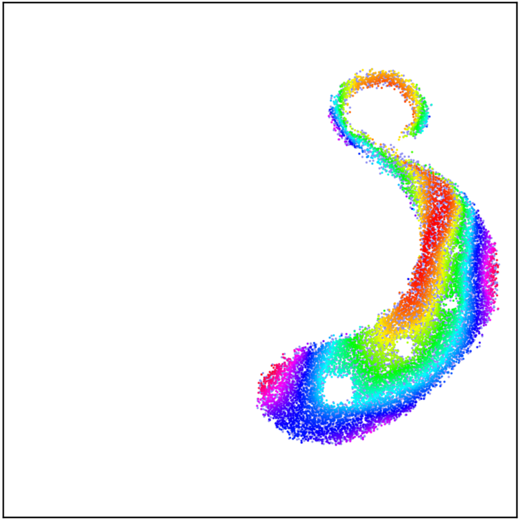
Iteration 1

Gradient descent on S_ε: cheap'n easy registration?



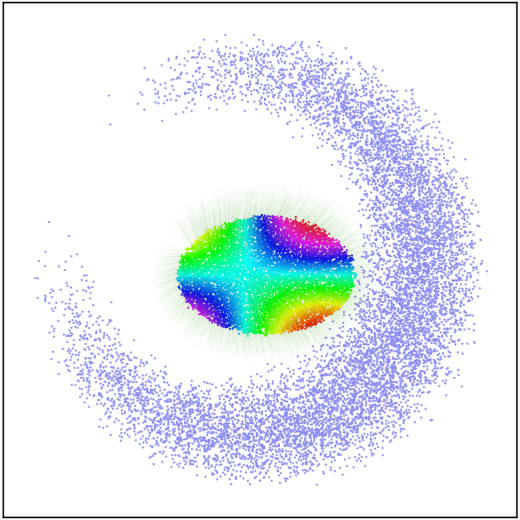
Iteration 2

Gradient descent on S_ε: cheap'n easy registration?



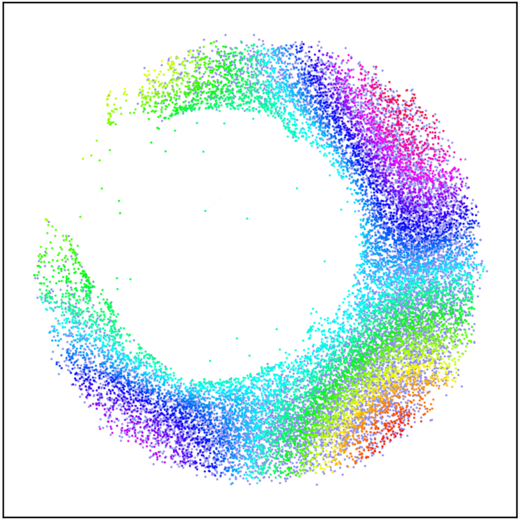
Iteration 10

Gradient descent on S_ε: cheap'n easy registration?



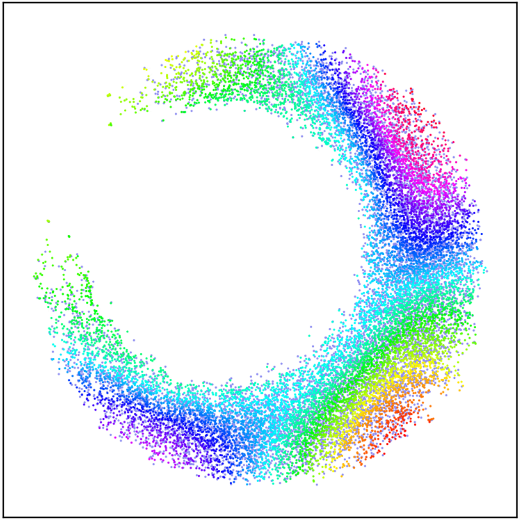
Iteration 0

Gradient descent on S_ε: cheap'n easy registration?



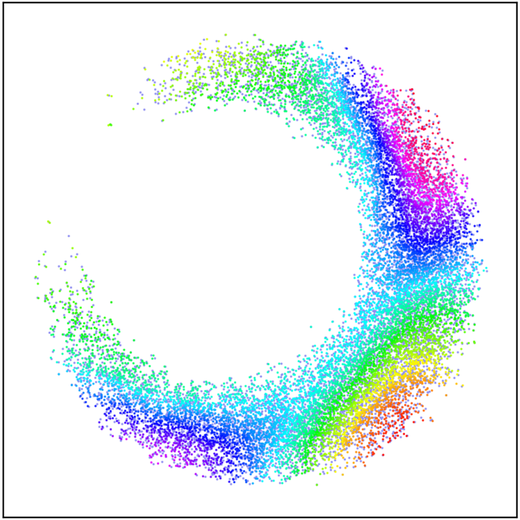
Iteration 1

Gradient descent on S_ε: cheap'n easy registration?



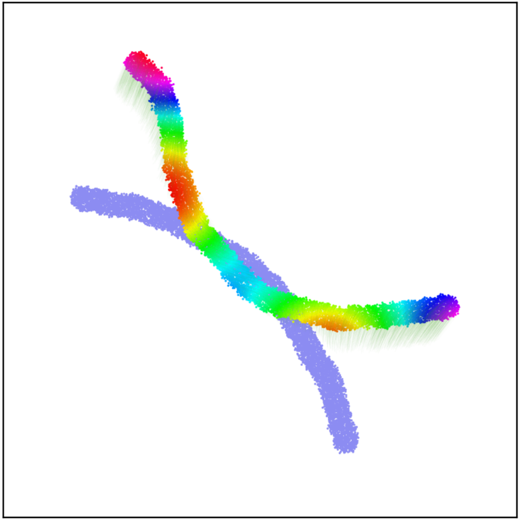
Iteration 2

Gradient descent on S_ε: cheap'n easy registration?



Iteration 10

Gradient descent on S_ε: cheap'n easy registration?



Iteration 0

Gradient descent on S_ε: cheap'n easy registration?

Iteration 1

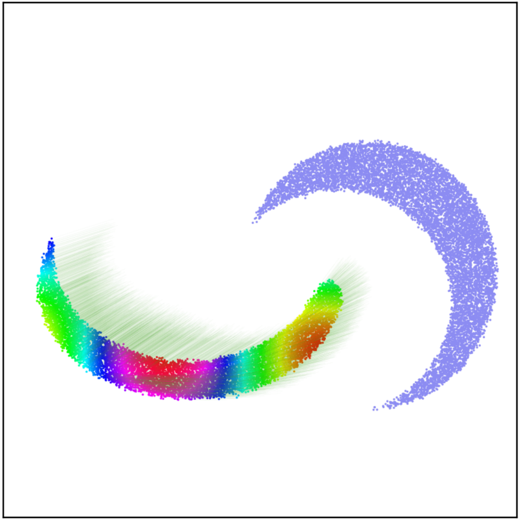
Gradient descent on S_ε: cheap'n easy registration?

Iteration 2

Gradient descent on S_ε: cheap'n easy registration?

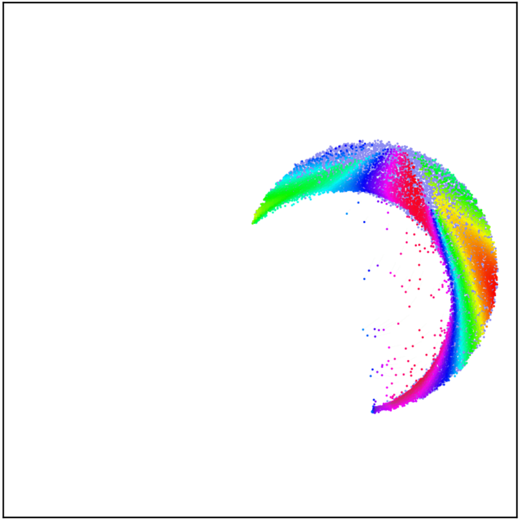
Iteration 10

Gradient descent on S_ε: cheap'n easy registration? Beware!



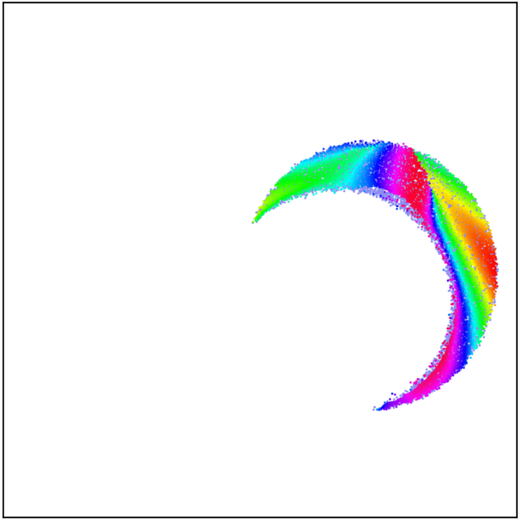
Iteration 0

Gradient descent on S_ε: cheap'n easy registration? Beware!



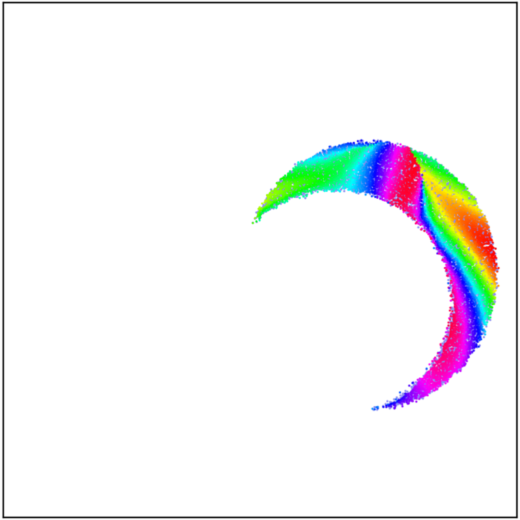
Iteration 1

Gradient descent on S_ε: cheap'n easy registration? Beware!



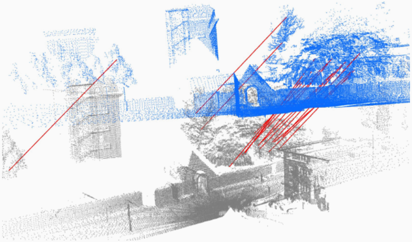
Iteration 2

Gradient descent on S_ε: cheap'n easy registration? Beware!



Iteration 10

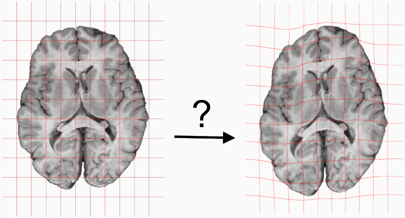
First setting: processing of point clouds



From the documentation of the Point Cloud Library.

- φ is **rigid** or affine
- Occlusions
- Outliers

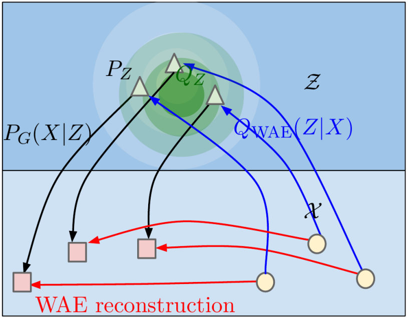
Second setting: medical imaging



From Marc Niethammer's Quicksilver slides.

- φ is a spline or a **diffeomorphism**
- Ill-posed problem
- Some occlusions

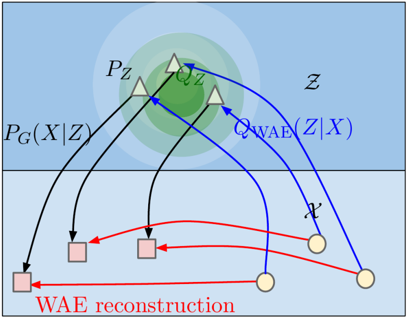
Third setting: training a generative model



Wasserstein Auto-Encoders,
Tolstikhin et al., 2018.

- φ is a **neural network**
- Very weak regularization
- High-dimensional space

Third setting: training a generative model



Wasserstein Auto-Encoders, Tolstikhin et al., 2018.

- φ is a **neural network**
- Very weak regularization
- High-dimensional space

Which **Loss** function should we use?

Dual norms - link with the GANs literature

$$\text{Loss}(\alpha, \beta) = \max_{f \in B} \langle \alpha - \beta , f \rangle,$$

$$\text{look for } \theta^* = \arg\min_{\theta} \max_{f \in B} \langle \alpha(\theta) - \beta , f \rangle$$

Dual norms - link with the GANs literature

$$\text{Loss}(\alpha, \beta) \;=\; \max_{f \in B} \langle\, \alpha - \beta \,,\, f \rangle,$$

$$\text{look for } \theta^* \;=\; \arg\min_{\theta} \max_{f \in B} \langle\, \alpha(\theta) - \beta \,,\, f \rangle$$

- $B = \{\, \|f\|_\infty \leqslant 1 \,\} \implies \text{Loss} = \text{TV norm:}$
  - zero geometry
  - **too many** test functions

Dual norms - link with the GANs literature

$$\text{Loss}(\alpha, \beta) \;=\; \max_{f \in B} \langle\, \alpha - \beta \,,\, f \,\rangle,$$

$$\text{look for } \theta^* \;=\; \arg\min_{\theta} \max_{f \in B} \langle\, \alpha(\theta) - \beta \,,\, f \,\rangle$$

- $B = \{ \, \| f \|_\infty \leqslant 1 \, \} \implies$ Loss $=$ TV norm:
    - zero geometry
    - **too many** test functions

- $B = \{ \, \| f \|_2^2 + \| \nabla f \|_2^2 + \cdots \leqslant 1 \, \} \implies$ Loss $=$ kernel norm:
    - may saturate at infinity
    - **screening** artifacts

Dual norms - link with the GANs literature

$$\text{Loss}(\alpha, \beta) = \max_{f \in B} \langle \alpha - \beta, f \rangle,$$

$$\text{look for } \theta^* = \arg\min_{\theta} \max_{f \in B} \langle \alpha(\theta) - \beta, f \rangle$$

- $B = \{ f \text{ is 1-Lipschitz} \} \implies$ Loss = Wasserstein-1 (OT_0):

Dual norms - link with the GANs literature

$$\text{Loss}(\alpha, \beta) \; = \; \max_{f \in B} \langle \, \alpha - \beta \, , \, f \rangle,$$

$$\text{look for } \; \theta^* \; = \; \arg\min_{\theta} \max_{f \in B} \langle \, \alpha(\theta) - \beta \, , \, f \rangle$$

- $B = \{ \, f \text{ is 1-Lipschitz} \, \} \implies$ Loss $=$ Wasserstein-1 (OT_0):
    - S_ε is nearly as efficient as a **closed formula**

Dual norms - link with the GANs literature

$$\text{Loss}(\alpha, \beta) \;=\; \max_{f \in B} \langle\, \alpha - \beta \,,\, f \,\rangle,$$

$$\text{look for } \;\theta^* \;=\; \arg\min_{\theta} \max_{f \in B} \langle\, \alpha(\theta) - \beta \,,\, f \,\rangle$$

- $B = \{\, f \text{ is 1-Lipschitz} \,\} \implies$ Loss $=$ Wasserstein-1 (OT_0):
    - S_ε is nearly as efficient as a **closed formula**
    - relevant in **low dimensions**
    - **useless** in $(\mathbb{R}^{512\times 512}, \|\cdot\|_2)$: the ground cost makes no sense

Dual norms - link with the GANs literature

$$\text{Loss}(\alpha, \beta) \;=\; \max_{f \in B} \langle\, \alpha - \beta\, , f \,\rangle,$$

$$\text{look for } \theta^* \;=\; \arg\min_{\theta} \max_{f \in B} \langle\, \alpha(\theta) - \beta\, , f \,\rangle$$

- $B = \{\, f \text{ is 1-Lipschitz} \,\} \implies$ Loss $=$ Wasserstein-1 (OT_0):
  - S_ε is nearly as efficient as a **closed formula**
  - relevant in **low dimensions**
  - **useless** in $(\mathbb{R}^{512\times 512}, \|\cdot\|_2)$: the ground cost makes no sense

- $B \simeq \{\, f \text{ is 1-Lipschitz} \,\} \bigcap \{\, f \text{ is a } \textbf{CNN} \,\}$

  $\implies$ Loss $=$ Wasserstein GAN :

Dual norms - link with the GANs literature

$$\text{Loss}(\alpha, \beta) = \max_{f \in B} \langle \alpha - \beta, f \rangle,$$

$$\text{look for } \theta^* = \arg\min_{\theta} \max_{f \in B} \langle \alpha(\theta) - \beta, f \rangle$$

- $B = \{ f \text{ is 1-Lipschitz} \} \implies$ Loss $=$ Wasserstein-1 (OT_0):
  - S_ε is nearly as efficient as a **closed formula**
  - relevant in **low dimensions**
  - **useless** in $(\mathbb{R}^{512\times 512}, \|\cdot\|_2)$: the ground cost makes no sense

- $B \simeq \{ f \text{ is 1-Lipschitz} \} \bigcap \{ f \text{ is a } \textbf{CNN} \}$
  $\implies$ Loss $=$ Wasserstein GAN :
  - use **perceptually sensible** test functions

Dual norms - link with the GANs literature

$$\text{Loss}(\alpha, \beta) = \max_{f \in B} \langle \alpha - \beta , f \rangle,$$

$$\text{look for } \theta^* = \arg\min_{\theta} \max_{f \in B} \langle \alpha(\theta) - \beta , f \rangle$$

- $B = \{ f \text{ is 1-Lipschitz} \} \implies \text{Loss} = \text{Wasserstein-1 } (\text{OT}_0)$:
  - S_ε is nearly as efficient as a **closed formula**
  - relevant in **low dimensions**
  - **useless** in $(\mathbb{R}^{512 \times 512}, \| \cdot \|_2)$: the ground cost makes no sense

- $B \simeq \{ f \text{ is 1-Lipschitz} \} \bigcap \{ f \text{ is a } \textbf{CNN} \}$
  $\implies \text{Loss} = \text{Wasserstein GAN}$:
  - use **perceptually sensible** test functions
  - no simple formula: use **gradient ascent**

Dual norms - link with the GANs literature

$$\text{Loss}(\alpha, \beta) = \max_{f \in B} \langle \alpha - \beta, f \rangle,$$

$$\text{look for } \theta^* = \arg\min_{\theta} \max_{f \in B} \langle \alpha(\theta) - \beta, f \rangle$$

- $B = \{ f \text{ is 1-Lipschitz} \} \implies \text{Loss} = \text{Wasserstein-1 } (\text{OT}_0)$:
  - S_ε is nearly as efficient as a **closed formula**
  - relevant in **low dimensions**
  - **useless** in $(\mathbb{R}^{512 \times 512}, \| \cdot \|_2)$: the ground cost makes no sense

- $B \simeq \{ f \text{ is 1-Lipschitz} \} \bigcap \{ f \text{ is a } \textbf{CNN} \}$
  $\implies \text{Loss} = \text{Wasserstein GAN}$:
  - use **perceptually sensible** test functions
  - no simple formula: use **gradient ascent**
  - can we provide relevant **insights** to the ML community?

References

Our papers:

- *Global divergences between measures: from Hausdorff distance to Optimal Transport*, F., Trouvé, 2018

References

Our papers:

- *Global divergences between measures: from Hausdorff distance to Optimal Transport*, F., Trouvé, 2018
- *Sinkhorn entropies and divergences*, F., Séjourné, Vialard, Amari, Trouvé, Peyré, 2018

References

Our papers:

- *Global divergences between measures: from Hausdorff distance to Optimal Transport*, F., Trouvé, 2018
- *Sinkhorn entropies and divergences*,
  F., Séjourné, Vialard, Amari, Trouvé, Peyré, 2018
- *Optimal Transport for diffeomorphic registration*,
  F., Charlier, Vialard, Peyré, 2017

Charlier, B., Feydy, J., and Glaunès, J. (2018).
Kernel operations on the gpu, with autodiff, without memory overflows.
http://www.kernel-operations.io.
Accessed: 2019-01-20.

Chizat, L., Peyré, G., Schmitzer, B., and Vialard, F.-X. (2018).
Unbalanced optimal transport: Dynamic and kantorovich formulations.
Journal of Functional Analysis, 274(11):3090–3123.

References ii

Genevay, A., Peyre, G., and Cuturi, M. (2018).
Learning generative models with sinkhorn divergences.
In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1608–1617. PMLR.

Kaltenmark, I., Charlier, B., and Charon, N. (2017).
A general framework for curve and surface comparison and registration with oriented varifolds.
In *Computer Vision and Pattern Recognition (CVPR)*.

Kosowsky, J. and Yuille, A. L. (1994).
The invisible hand algorithm: Solving the assignment problem with statistical physics.
Neural networks, 7(3):477–490.

Ramdas, A., Trillos, N. G., and Cuturi, M. (2017).
On wasserstein two-sample testing and related families of nonparametric tests.
Entropy, 19(2).

Salimans, T., Zhang, H., Radford, A., and Metaxas, D. (2018).
Improving GANs using optimal transport.
arXiv preprint arXiv:1803.05573.

- Sanjabi, M., Ba, J., Razaviyayn, M., and Lee, J. D. (2018).
  On the convergence and robustness of training GANs with regularized optimal transport.
  arXiv preprint arXiv:1802.08249.
- Schmitzer, B. (2016).
  Stabilized sparse scaling algorithms for entropy regularized transport problems.
  arXiv preprint arXiv:1610.06519.