Optimal Transport for Diffeomorphic Registration

MICCAI 2017 – September 12

Jean Feydy[1,2] Benjamin Charlier[3,5] François-Xavier Vialard[4,6] Gabriel Peyré[1,5]

[1]DMA – École Normale Supérieure, Paris, France [2]CMLA – ENS Cachan, Cachan, France

[3]Institut Montpelliérain Alexander Grothendieck, Univ. Montpellier, Montpellier, France

[4]Univ. Paris-Dauphine - PSL Research, Paris, France [5]CNRS, Paris, France [6]INRIA Mokaplan, Paris, France

The Diffeomorphic Registration problem

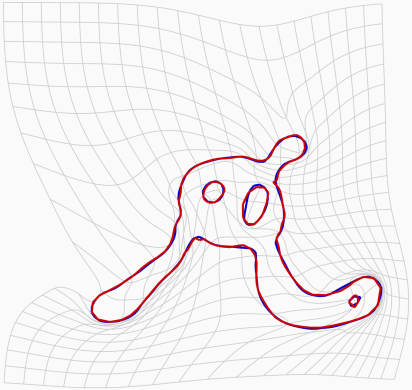


Figure: Matching a curve to another.

Registrate a shape A to a shape B through a (rigid, diffeomorphic, etc.) transformation φ:

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

The Diffeomorphic Registration problem

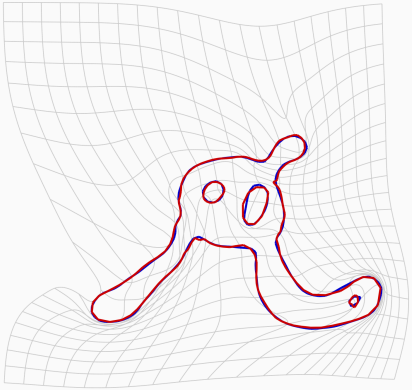


Figure: Matching a curve to another.

Registrate a shape A to a shape B through a (rigid, diffeomorphic, etc.) transformation φ:

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

Variational setting : minimize

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

A Diffeomorphic Registration pipeline

Iterative Matching Algorithm

Input : source A
target B

1: **while** updates $>$ tol **do**
2: Compute $\nabla_\varphi \mathrm{Reg}(\varphi)$.
3: Compute $\nabla_\varphi \left[d(\varphi(A) \to B)\right]$.
4: "$\varphi \leftarrow \varphi - \alpha \cdot \big($
$\nabla_\varphi \mathrm{Reg}(\varphi) + \nabla_\varphi \left[d(\varphi(A) \to B)\right] \big)$"
5: **return** φ

Output : matching transformation φ.

A Diffeomorphic Registration pipeline

Iterative Matching Algorithm

Input : source A
target B

1: **while** updates $>$ tol **do**
2: Compute $\nabla_\varphi \mathrm{Reg}(\varphi)$.
3: Compute $\nabla_\varphi \left[d(\varphi(A) \to B)\right]$.
4: "$\varphi \leftarrow \varphi - \alpha \cdot \big(\nabla_\varphi \mathrm{Reg}(\varphi) + \nabla_\varphi \left[d(\varphi(A) \to B)\right] \big)$"
5: **return** φ

Output : matching transformation φ.

Diffeomorphic registration **pipeline**:

- Optimization strategy: toolbox
- Regularizer: deformation model
- Data attachment: rating formula

A Diffeomorphic Registration pipeline

Iterative Matching Algorithm

Input : source A
target B

1: **while** updates $>$ tol **do**
2: Compute $\nabla_\varphi \mathrm{Reg}(\varphi)$.
3: Compute $\nabla_\varphi \left[d(\varphi(A) \to B)\right]$.
4: "$\varphi \leftarrow \varphi - \alpha \cdot \big(\nabla_\varphi \mathrm{Reg}(\varphi) + \nabla_\varphi \left[d(\varphi(A) \to B)\right] \big)$"
5: **return** φ

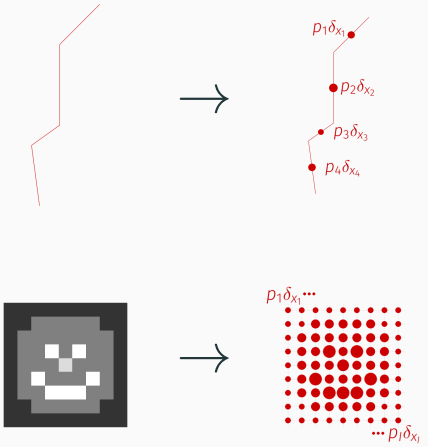
Output : matching transformation φ.

Diffeomorphic registration **pipeline**:

- Optimization strategy: toolbox
- Regularizer: deformation model
- Data attachment: rating formula

How do we define $d(\varphi(A) \to B)$?

Unlabeled shapes are encoded as measures



We will work with:

$$\varphi(A) \longleftrightarrow \mu = \sum_{i=1}^{I} p_i \delta_{x_i},$$

$$B \longleftrightarrow \nu = \sum_{j=1}^{J} q_j \delta_{y_j}.$$

In practice:

- Segmented surfaces:

  List of (x_i, p_i) and (y_j, q_j).

- Volumetric Data:

  p and q as grid images.

Kernel fidelities: the simplest formula for $d(\mu \to \nu)$

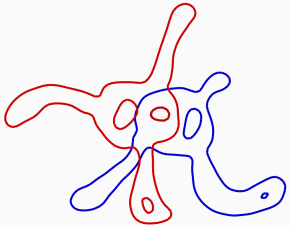
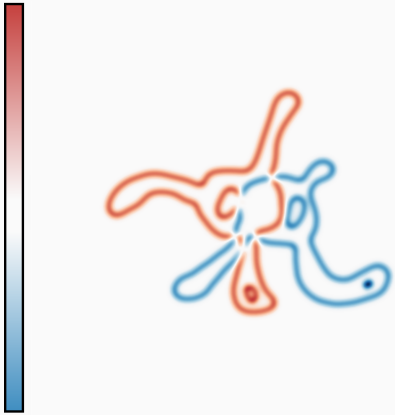


Figure: Raw signal ($\mu - \nu$).

Kernel fidelities: the simplest formula for $d(\mu \to \nu)$



Choose a blurring function g, use

$$d_k(\mu \to \nu) \;=\; \| g \star \mu - g \star \nu \|_{L^2}^2$$

Figure: Blurred signal $g \star (\mu - \nu)$.

Kernel fidelities: the simplest formula for $d(\mu \to \nu)$

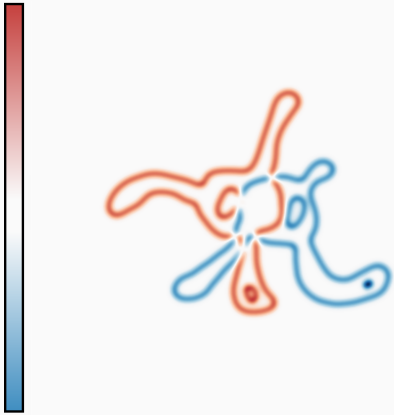


Figure: Blurred signal $g \star (\mu - \nu)$.

Choose a blurring function g, use

$$\begin{aligned} d_k(\mu \to \nu) &= \| g \star \mu - g \star \nu \|_{L^2}^2 \\ &= \langle \mu - \nu \mid k \star (\mu - \nu) \rangle , \end{aligned}$$

where $k = g \star g$ is the kernel function.

Kernel fidelities: the simplest formula for $d(\mu \to \nu)$

Figure: Blurred signal $g \star (\mu - \nu)$.

Choose a blurring function g, use

$$\begin{aligned} d_k(\mu \to \nu) &= \| g \star \mu - g \star \nu \|_{L^2}^2 \\ &= \langle \mu - \nu \mid k \star (\mu - \nu) \rangle , \end{aligned}$$

where $k = g \star g$ is the kernel function.

Kernel fidelities: the simplest formula for $d(\mu \to \nu)$

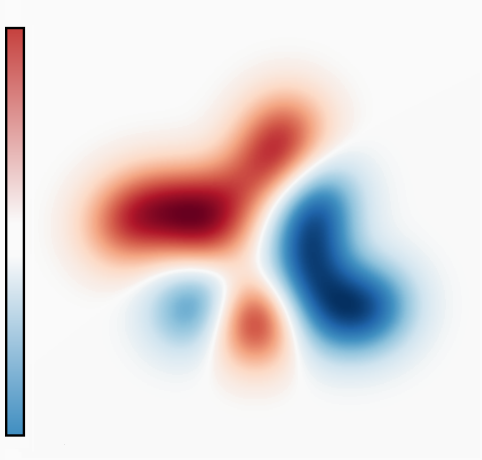


Figure: Blurred signal $g \star (\mu - \nu)$.

Choose a blurring function g, use

$$\begin{aligned} d_k(\mu \to \nu) &= \| g \star \mu - g \star \nu \|_{L^2}^2 \\ &= \langle \mu - \nu \mid k \star (\mu - \nu) \rangle , \end{aligned}$$

where $k = g \star g$ is the kernel function.

Kernel fidelities: the simplest formula for $d(\mu \to \nu)$

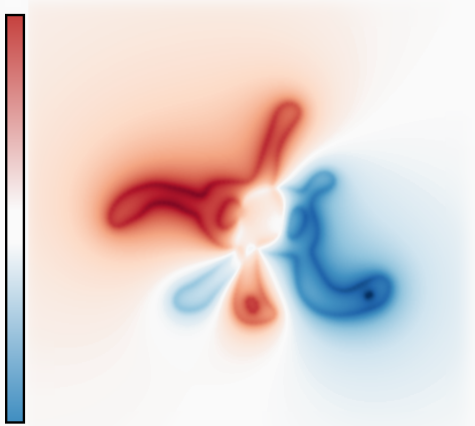


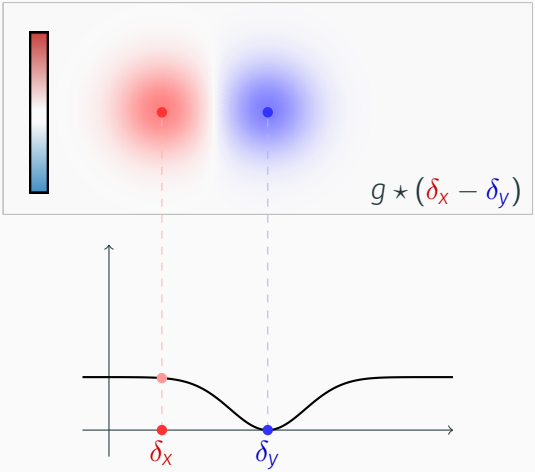
Figure: Blurred signal $g \star (\mu - \nu)$.

Choose a blurring function g, use

$$\begin{aligned} d_k(\mu \to \nu) &= \| g \star \mu - g \star \nu \|_{L^2}^2 \\ &= \langle \mu - \nu \mid k \star (\mu - \nu) \rangle , \end{aligned}$$

where $k = g \star g$ is the kernel function.

Kernel fidelities and long-distance vision



Using a kernel k, we have

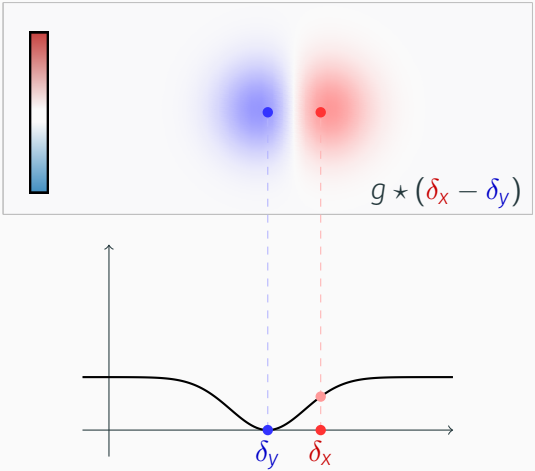
$$\begin{aligned} d_k(\delta_x \to \delta_y) &= \langle\, \delta_x - \delta_y \mid k \star (\delta_x - \delta_y) \,\rangle \\ &= k(x-x) - k(x-y) \\ &- k(y-x) + k(y-y) \\ &= 2\,[k(0) - k(x-y)]\,. \end{aligned}$$

Wouldn't we prefer to use

$$d(\delta_x \to \delta_y) = \|x-y\|^2$$

instead ?

Kernel fidelities and long-distance vision



Using a kernel k, we have

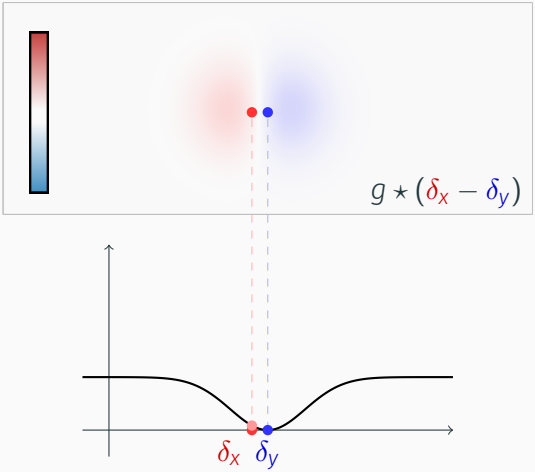
$$\begin{aligned} d_k(\delta_x \to \delta_y) &= \langle\, \delta_x - \delta_y \mid k \star (\delta_x - \delta_y) \,\rangle \\ &= k(x-x) - k(x-y) \\ &- k(y-x) + k(y-y) \\ &= 2\,[k(0) - k(x-y)]\,. \end{aligned}$$

Wouldn't we prefer to use

$$d(\delta_x \to \delta_y) = \|x-y\|^2$$

instead ?

Kernel fidelities and long-distance vision



Using a kernel k, we have

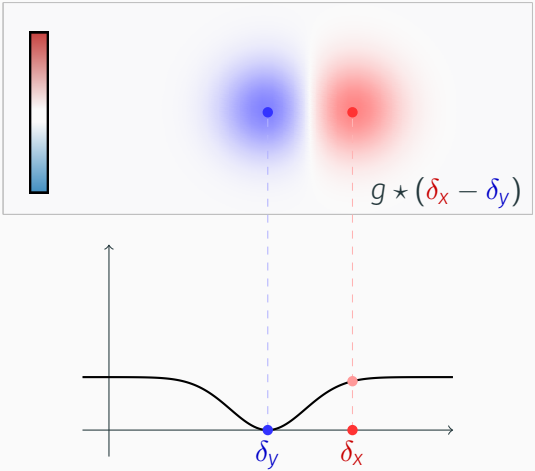
$$\begin{aligned} d_k(\delta_x \to \delta_y) &= \langle\, \delta_x - \delta_y \mid k \star (\delta_x - \delta_y) \,\rangle \\ &= k(x-x) - k(x-y) \\ &- k(y-x) + k(y-y) \\ &= 2\,[k(0) - k(x-y)]\,. \end{aligned}$$

Wouldn't we prefer to use

$$d(\delta_x \to \delta_y) = \|x-y\|^2$$

instead ?

Kernel fidelities and long-distance vision



Using a kernel k, we have

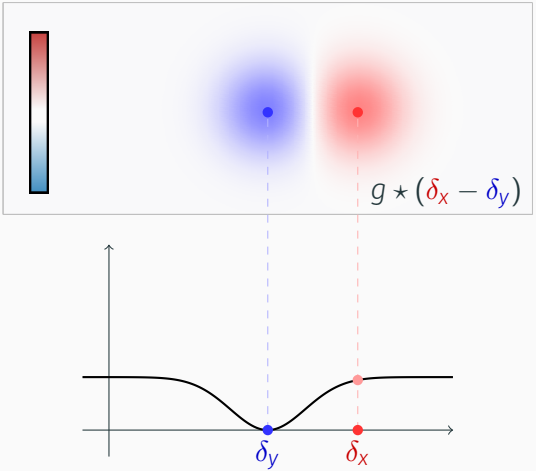
$$\begin{aligned} d_k(\delta_x \to \delta_y) &= \langle\, \delta_x - \delta_y \mid k \star (\delta_x - \delta_y) \,\rangle \\ &= k(x-x) - k(x-y) \\ &- k(y-x) + k(y-y) \\ &= 2\,[k(0) - k(x-y)]\,. \end{aligned}$$

Wouldn't we prefer to use

$$d(\delta_x \to \delta_y) = \|x-y\|^2$$

instead ?

Kernel fidelities and long-distance vision



Using a kernel k, we have

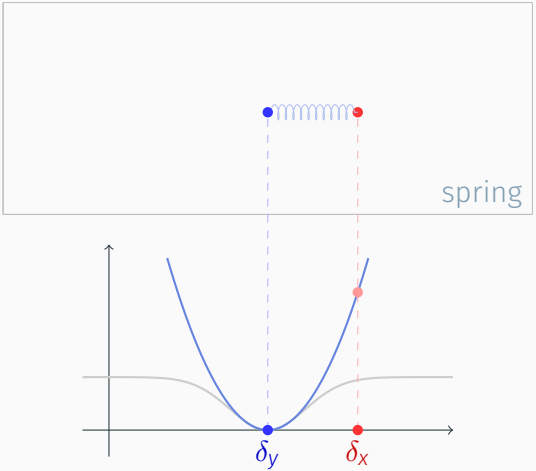
$$\begin{aligned} d_k(\delta_x \to \delta_y) &= \langle\, \delta_x - \delta_y \mid k \star (\delta_x - \delta_y) \,\rangle \\ &= k(x-x) - k(x-y) \\ &- k(y-x) + k(y-y) \\ &= 2\,[k(0) - k(x-y)]\,. \end{aligned}$$

Wouldn't we prefer to use

$$d(\delta_x \to \delta_y) = \|x-y\|^2$$

instead ?

Kernel fidelities and long-distance vision



Using a kernel k, we have

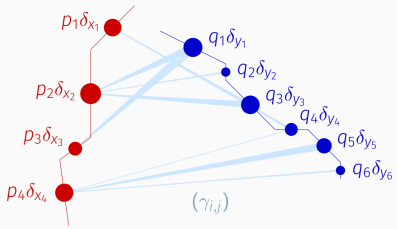
$$\begin{aligned} d_k(\delta_x \to \delta_y) &= \langle\, \delta_x - \delta_y \mid k \star (\delta_x - \delta_y) \,\rangle \\ &= k(x-x) - k(x-y) \\ &- k(y-x) + k(y-y) \\ &= 2\,[k(0) - k(x-y)]\,. \end{aligned}$$

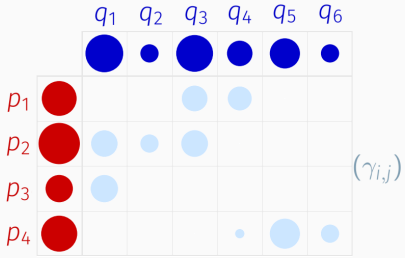
Wouldn't we prefer to use

$$d(\delta_x \to \delta_y) = \|x-y\|^2$$

instead ?

Introducing the Optimal Transport problem





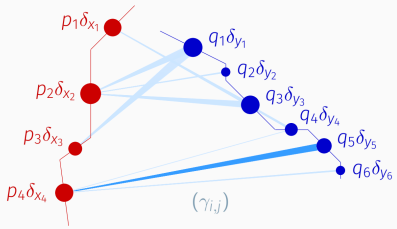
In the simplest setting, assume that μ and ν have the same total mass. Define the OT fidelity through a minimization on I-by-J matrices – called transport plans Γ:

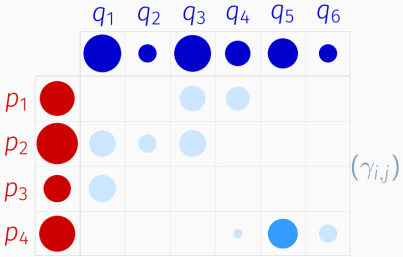
$$\min_{\Gamma} \underbrace{\sum_{i,j} \gamma_{i,j} \cdot |x_i - y_j|^2}_{\text{transport cost}} + \underbrace{\varepsilon \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j}}_{\text{entropic regularization}}$$

under the constraint that

$$\gamma_{i,j} \geqslant 0, \quad \sum_j \gamma_{i,j} = p_i, \quad \sum_i \gamma_{i,j} = q_j.$$

Introducing the Optimal Transport problem





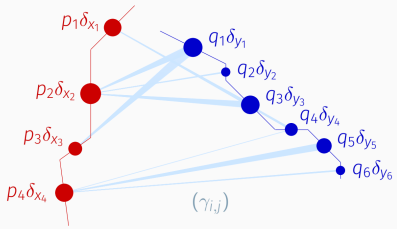
In the simplest setting, assume that μ and ν have the same total mass. Define the OT fidelity through a minimization on I-by-J matrices – called transport plans Γ:

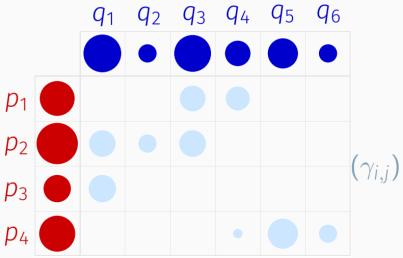
$$\min_{\Gamma} \underbrace{\sum_{i,j} \gamma_{i,j} \cdot |x_i - y_j|^2}_{\text{transport cost}} + \underbrace{\varepsilon \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j}}_{\text{entropic regularization}}$$

under the constraint that

$$\gamma_{i,j} \geqslant 0, \quad \sum_j \gamma_{i,j} = p_i, \quad \sum_i \gamma_{i,j} = q_j.$$

Introducing the Optimal Transport problem





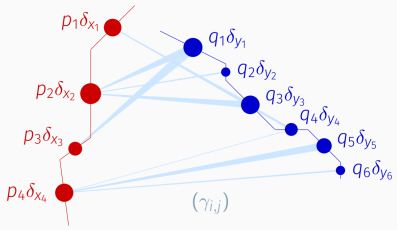
In the simplest setting, assume that μ and ν have the same total mass. Define the OT fidelity through a minimization on I-by-J matrices – called transport plans Γ:

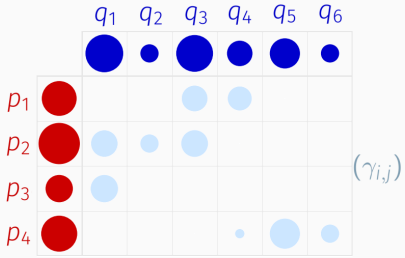
$$\min_{\Gamma} \underbrace{\sum_{i,j} \gamma_{i,j} \cdot |x_i - y_j|^2}_{\text{transport cost}} + \underbrace{\varepsilon \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j}}_{\text{entropic regularization}}$$

under the constraint that

$$\gamma_{i,j} \geqslant 0, \quad \sum_j \gamma_{i,j} = p_i, \quad \sum_i \gamma_{i,j} = q_j.$$

Introducing the Optimal Transport problem





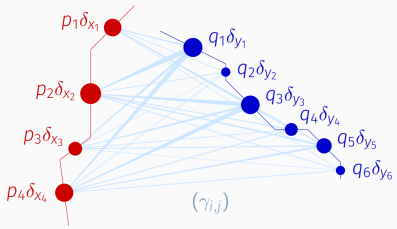
In the simplest setting, assume that μ and ν have the same total mass. Define the OT fidelity through a minimization on I-by-J matrices – called transport plans Γ:

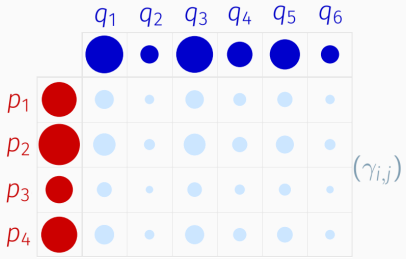
$$\min_{\Gamma} \underbrace{\sum_{i,j} \gamma_{i,j} \cdot |x_i - y_j|^2}_{\text{transport cost}} + \underbrace{\varepsilon \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j}}_{\text{entropic regularization}}$$

under the constraint that

$$\gamma_{i,j} \geqslant 0, \quad \sum_j \gamma_{i,j} = p_i, \quad \sum_i \gamma_{i,j} = q_j.$$

Introducing the Optimal Transport problem





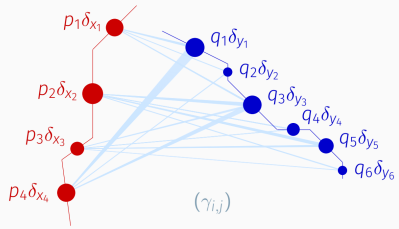
In the simplest setting, assume that μ and ν have the same total mass. Define the OT fidelity through a minimization on I-by-J matrices – called transport plans Γ:

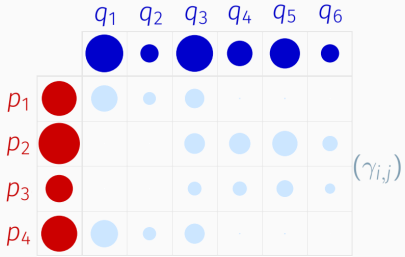
$$\min_{\Gamma} \underbrace{\sum_{i,j} \gamma_{i,j} \cdot |x_i - y_j|^2}_{\text{transport cost}} + \underbrace{\varepsilon \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j}}_{\text{entropic regularization}}$$

under the constraint that

$$\gamma_{i,j} \geqslant 0, \quad \sum_j \gamma_{i,j} = p_i, \quad \sum_i \gamma_{i,j} = q_j.$$

Introducing the Optimal Transport problem





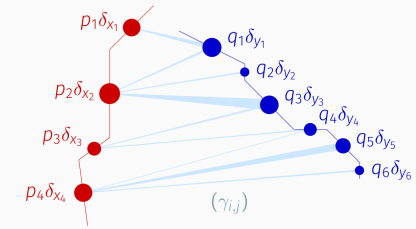
In the simplest setting, assume that μ and ν have the same total mass. Define the OT fidelity through a minimization on I-by-J matrices – called transport plans Γ:

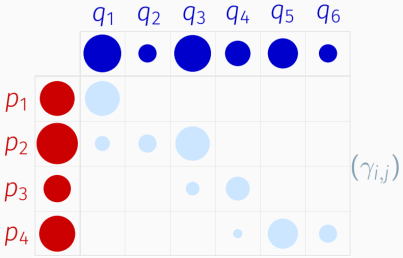
$$\min_{\Gamma} \underbrace{\sum_{i,j} \gamma_{i,j} \cdot |x_i - y_j|^2}_{\text{transport cost}} + \underbrace{\varepsilon \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j}}_{\text{entropic regularization}}$$

under the constraint that

$$\gamma_{i,j} \geqslant 0, \quad \sum_j \gamma_{i,j} = p_i, \quad \sum_i \gamma_{i,j} = q_j.$$

Introducing the Optimal Transport problem





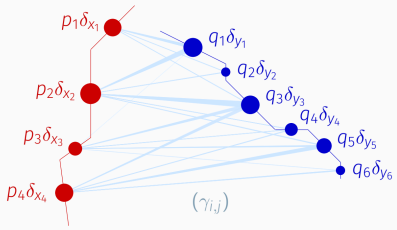
In the simplest setting, assume that μ and ν have the same total mass. Define the OT fidelity through a minimization on I-by-J matrices – called transport plans Γ:

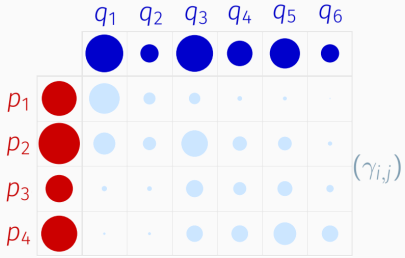
$$\min_{\Gamma} \underbrace{\sum_{i,j} \gamma_{i,j} \cdot |x_i - y_j|^2}_{\text{transport cost}} + \underbrace{\varepsilon \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j}}_{\text{entropic regularization}}$$

under the constraint that

$$\gamma_{i,j} \geqslant 0, \quad \sum_j \gamma_{i,j} = p_i, \quad \sum_i \gamma_{i,j} = q_j.$$

Introducing the Optimal Transport problem





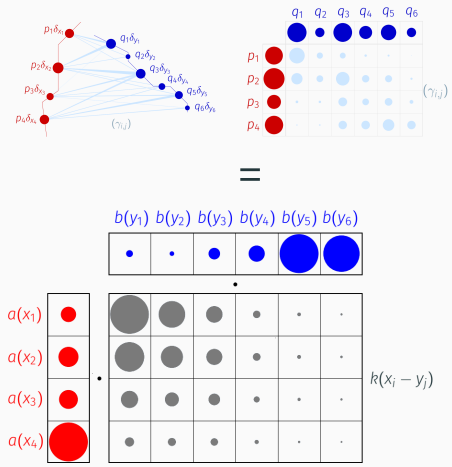
In the simplest setting, assume that μ and ν have the same total mass. Define the OT fidelity through a minimization on I-by-J matrices – called transport plans Γ:

$$\min_{\Gamma} \underbrace{\sum_{i,j} \gamma_{i,j} \cdot |x_i - y_j|^2}_{\text{transport cost}} + \underbrace{\varepsilon \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j}}_{\text{entropic regularization}}$$

under the constraint that

$$\gamma_{i,j} \geqslant 0, \quad \sum_j \gamma_{i,j} = p_i, \quad \sum_i \gamma_{i,j} = q_j.$$

The regularized OT problem is tractable through a kernel factorization



Optimality conditions show that the OT plan can be written as a product

$$\gamma_{i,j} \;=\; \Gamma(x_i \to y_j) \;=\; a(x_i)\, k(x_i - y_j)\, b(y_j),$$

where:

- The kernel function k is given by

$$k(x_i - y_j) \;=\; e^{-|x_i - y_j|^2/\varepsilon}.$$

- a and $b \geqslant 0$ are functions supported by $\{x_i\}$ and $\{y_j\}$.

Computing the OT fidelity at a cost of 100-1000 convolutions

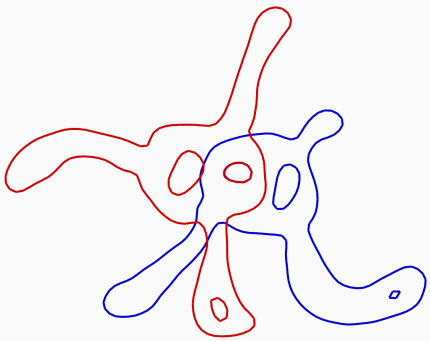


Figure: Source and target measures.

Sinkhorn Iterative Algorithm

Input : source $\mu = \sum_i p_i \delta_{x_i}$

target $\nu = \sum_j q_j \delta_{y_j}$

Parameter : $k : x \mapsto e^{-|x|^2/\varepsilon}$

$a \leftarrow ones(size(p))$

$b \leftarrow ones(size(q))$

while updates $>$ tol **do**

  $a \leftarrow p \,/\, (k \star b)$

  $b \leftarrow q \,/\, (k \star a)$

return $\varepsilon \cdot \big(\langle p, \log(a) + 1/2 \rangle + \langle q, \log(b) + 1/2 \rangle \big)$

Output : fidelity $W_\varepsilon(\mu, \nu)$

Computing the OT fidelity at a cost of 100-1000 convolutions

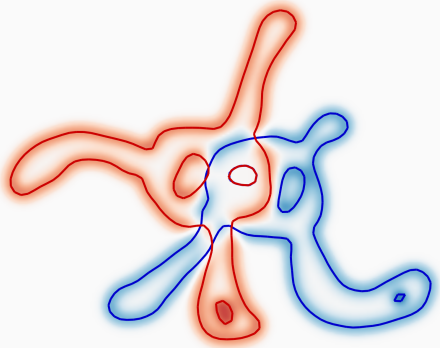


Figure: Data as seen by the kernel k.

Sinkhorn Iterative Algorithm

Input : source $\mu = \sum_i p_i \delta_{x_i}$

target $\nu = \sum_j q_j \delta_{y_j}$

Parameter : $k : x \mapsto e^{-|x|^2/\varepsilon}$

$a \leftarrow ones(size(p))$

$b \leftarrow ones(size(q))$

while updates $>$ tol **do**

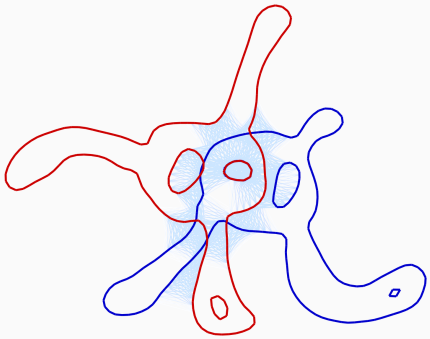
  $a \leftarrow p \,/\, (k \star b)$

  $b \leftarrow q \,/\, (k \star a)$

return $\varepsilon \cdot \big(\langle p, \log(a) + 1/2 \rangle + \langle q, \log(b) + 1/2 \rangle \big)$

Output : fidelity $W_\varepsilon(\mu, \nu)$

Computing the OT fidelity at a cost of 100-1000 convolutions



Sinkhorn Iteration 000

Figure: Starting estimate.

Sinkhorn Iterative Algorithm

Input : source $\mu = \sum_i p_i \delta_{x_i}$

target $\nu = \sum_j q_j \delta_{y_j}$

Parameter : $k : x \mapsto e^{-|x|^2/\varepsilon}$

$a \leftarrow ones(size(p))$

$b \leftarrow ones(size(q))$

while updates $>$ tol **do**

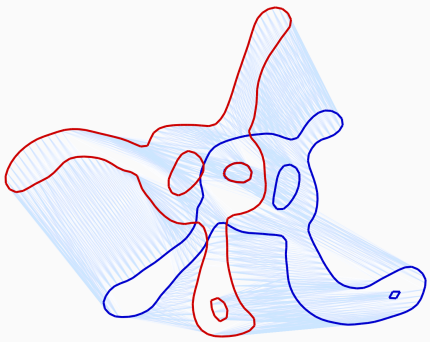
  $a \leftarrow p \,/\, (k \star b)$

  $b \leftarrow q \,/\, (k \star a)$

return $\varepsilon \cdot \big(\langle p, \log(a) + 1/2 \rangle + \langle q, \log(b) + 1/2 \rangle \big)$

Output : fidelity $W_\varepsilon(\mu, \nu)$

Computing the OT fidelity at a cost of 100-1000 convolutions



Sinkhorn Iteration 250

Figure: Computing the OT plan.

Sinkhorn Iterative Algorithm

Input : source $\mu = \sum_i p_i \delta_{x_i}$

target $\nu = \sum_j q_j \delta_{y_j}$

Parameter : $k : x \mapsto e^{-|x|^2/\varepsilon}$

$a \leftarrow ones(size(p))$

$b \leftarrow ones(size(q))$

while updates $>$ tol **do**

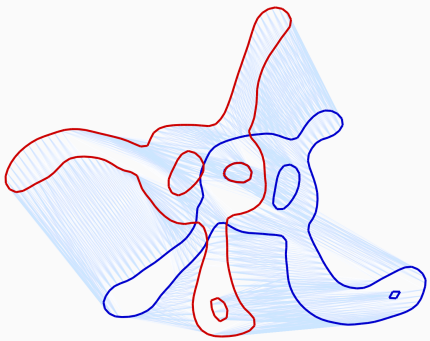
  $a \leftarrow p \,/\, (k \star b)$

  $b \leftarrow q \,/\, (k \star a)$

return $\varepsilon \cdot \big(\langle p, \log(a) + 1/2 \rangle + \langle q, \log(b) + 1/2 \rangle \big)$

Output : fidelity $W_\varepsilon(\mu, \nu)$

Computing the OT fidelity at a cost of 100-1000 convolutions



Sinkhorn Iteration 250

Figure: Computing the OT plan.

Sinkhorn Iterative Algorithm

Input : source $\mu = \sum_i p_i \delta_{x_i}$

target $\nu = \sum_j q_j \delta_{y_j}$

Parameter : $k : x \mapsto e^{-|x|^2/\varepsilon}$

$a \leftarrow ones(size(p))$

$b \leftarrow ones(size(q))$

while updates $>$ tol **do**

  $a \leftarrow p \,/\, (k \star b)$

  $b \leftarrow q \,/\, (k \star a)$

return $\varepsilon \cdot \big(\langle p, \log(a) + 1/2 \rangle + \langle q, \log(b) + 1/2 \rangle \big)$

Output : fidelity $W_\varepsilon(\mu, \nu)$

Bonus features, resulting in a flexible iterative framework

Minimize under the constraints that

$$\Gamma \mathbf{1} = p \qquad \text{and} \qquad \Gamma^{\mathsf{T}} \mathbf{1} = q.$$

$$\Downarrow$$

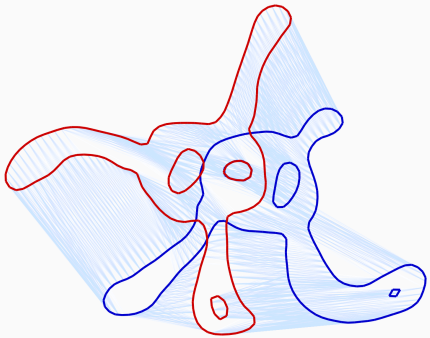
$$\min_{\Gamma \geqslant 0} \underbrace{\left\langle \Gamma \,,\, |x_i - y_j|^2 \right\rangle - \varepsilon H(\Gamma)}_{\text{Objective}}$$

$$+ \underbrace{\rho \left[\mathrm{KL}(\Gamma \mathbf{1} \,|\, p) + \mathrm{KL}(\Gamma^{\mathsf{T}} \mathbf{1} \,|\, q) \right]}_{\text{Kullback regularization}}$$

In the paper, we show how to use:

- unbalanced measures,
- Nesterov acceleration,
- local features.

Bonus features, resulting in a flexible iterative framework



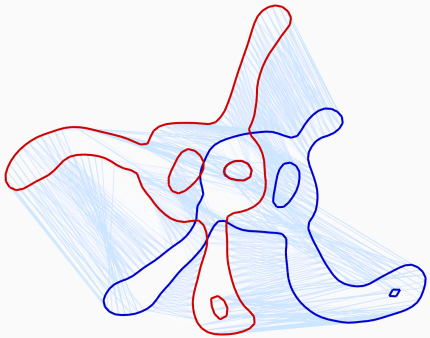
Sinkhorn Iteration 191

Figure: Accelerated Sinkhorn loop.

In the paper, we show how to use:

- unbalanced measures,
- Nesterov acceleration,
- local features.

Bonus features, resulting in a flexible iterative framework



Sinkhorn Iteration 190

Figure: “Position + Orientation” OT.

In the paper, we show how to use:

- unbalanced measures,
- Nesterov acceleration,
- local features.

Benefits of the entropic regularization

Adding a $+\varepsilon \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j}$ regularization:

- Linear $\longrightarrow$ Strictly convex problem.
- Compute the OT data attachment at the cost of 100-1000 convolutions with a (separable) gaussian kernel

$$k : x \mapsto e^{-x^2/\varepsilon}. \tag{1}$$

In the paper, we show how to fine-tune the underlying spring system Γ, which is a:

- non-smooth
- global

correspondence map between source and target.

OT (global coverage) + Diffeomorphic registration (smoothness) $=$?

Benefits of the entropic regularization

Adding a $+\varepsilon \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j}$ regularization:

- Linear $\longrightarrow$ Strictly convex problem.
- Compute the OT data attachment at the cost of 100-1000 convolutions with a (separable) gaussian kernel

$$k : x \mapsto e^{-x^2/\varepsilon}. \tag{1}$$

In the paper, we show how to fine-tune the underlying spring system Γ, which is a:

- non-smooth
- global

correspondence map between source and target.

OT (global coverage) + Diffeomorphic registration (smoothness) $=$?

Benefits of the entropic regularization

Adding a $+\varepsilon \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j}$ regularization:

- Linear $\longrightarrow$ Strictly convex problem.
- Compute the OT data attachment at the cost of 100-1000 convolutions with a (separable) gaussian kernel

$$k : x \mapsto e^{-x^2/\varepsilon}. \tag{1}$$

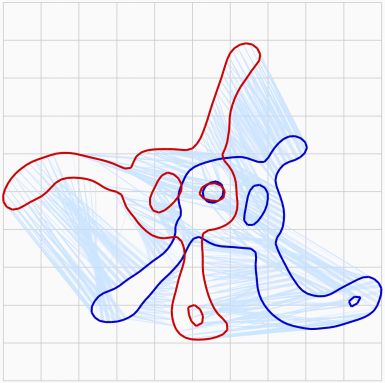
In the paper, we show how to fine-tune the underlying spring system Γ, which is a:

- non-smooth
- global

correspondence map between source and target.

OT (global coverage) + Diffeomorphic registration (smoothness) $=$?

Using OT plans as spring systems driving a registration routine



L-BFGS Iteration 01

Figure: LDDMM + OT registration.

Derivatives of the OT fidelity can be computed easily: plug it in any standard registration toolbox.

The eventual registration is both smooth and global.

Conclusion and practical use

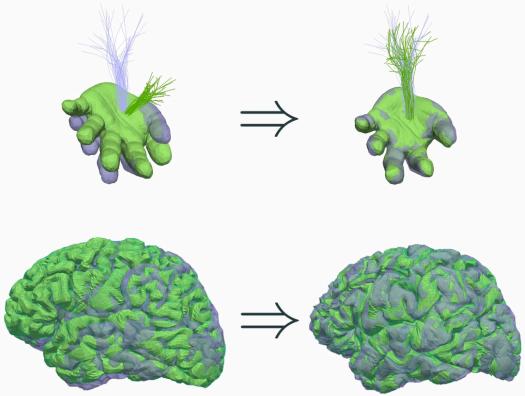


Figure: Examples of LDDMM+OT matchings.

- The 5-line **Sinkhorn Algorithm** provides kernel methods with long-range vision.
- As no target data is "out-of-sight", this idea should improve the robustness of your registration pipeline.

Thank you for your attention.