Robust shape matching with Optimal Transport

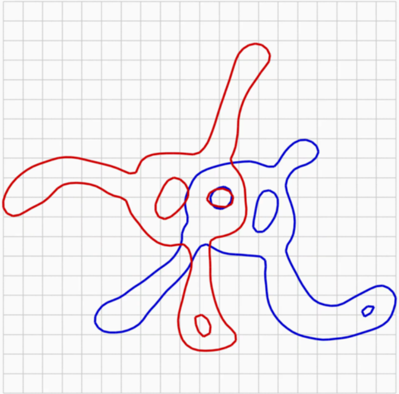
Jean Feydy

BIRS, Banff seminar 18w5151 – 13th December, 2018

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

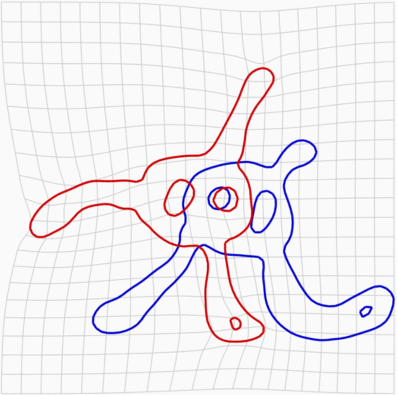
What is shape matching?

Source A, target B,



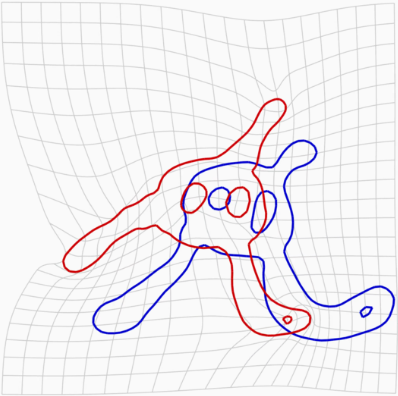
What is shape matching?

Source A, target B, mapping φ



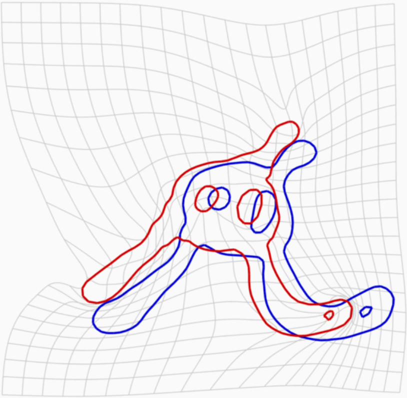
What is shape matching?

Source A, target B, mapping φ



What is shape matching?

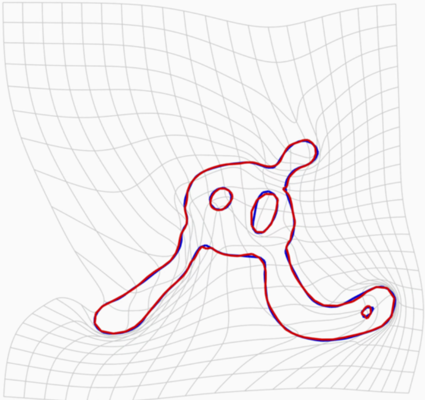
Source A, target B, mapping φ



What is shape matching?

Source A, target B, mapping φ

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



A good Loss function is a guarantee of robustness

Iterative Matching Algorithm

$A' \leftarrow A$

repeat

$\quad L, \nu \leftarrow \text{Loss}(A', B), \; -\partial_{A'}\text{Loss}(A', B)$

$\quad A' \leftarrow A' + \text{Model}(\nu)$

until $L < \text{tol}$

Output : deformed shape $A' = \varphi(A)$.

A good Loss function is a guarantee of robustness

Iterative Matching Algorithm

1: $A' \leftarrow A$
2: **repeat**
3: $\quad L\,,\ v \ \leftarrow\ \text{Loss}(A', B)\,,\ \ -\partial_{A'}\text{Loss}(A', B)$
4: $\quad A' \ \leftarrow\ A' \ +\ \text{Model}(\,v\,)$
5: **until** $\ L \ <\ \text{tol}$

Output : deformed shape $A' \ =\ \varphi(\,A\,)$.

"Model" encodes the **prior knowledge** on admissible deformations:

- *smoothing* convolution
- LDDMM/SVF *backprop* + regularization + *shooting*
- *trained* neural network

A good Loss function is a guarantee of robustness

Iterative Matching Algorithm

1: $A' \leftarrow A$
2: **repeat**
3: $\quad L, v \leftarrow \text{Loss}(A', B), \; -\partial_{A'}\text{Loss}(A', B)$
4: $\quad A' \leftarrow A' + \text{Model}(v)$
5: **until** $L < \text{tol}$

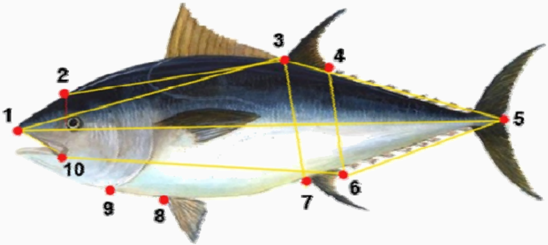
Output : deformed shape $A' = \varphi(A)$.

"Model" encodes the **prior knowledge** on admissible deformations:

- *smoothing* convolution
- LDDMM/SVF *backprop* + regularization + *shooting*
- *trained* neural network

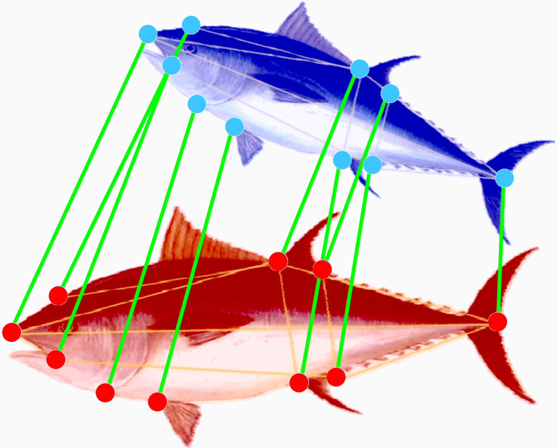
$\Rightarrow$ The *raw* Loss gradient v is what **drives** the registration

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

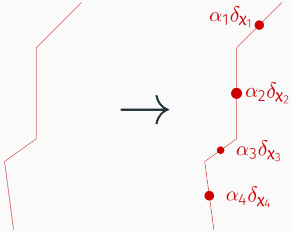
Let's enforce sampling invariance:

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

Encoding unlabeled shapes as measures

Let's enforce sampling invariance:

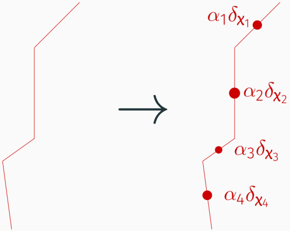
$$A \longrightarrow \alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i}, \qquad B \longrightarrow \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j}.$$

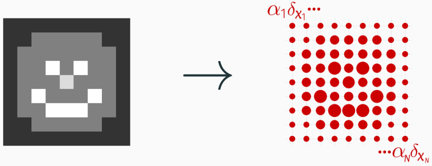


Encoding unlabeled shapes as measures

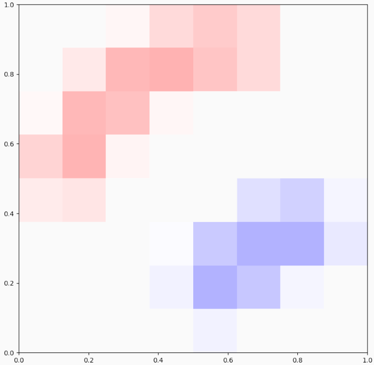
Let's enforce sampling invariance:

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

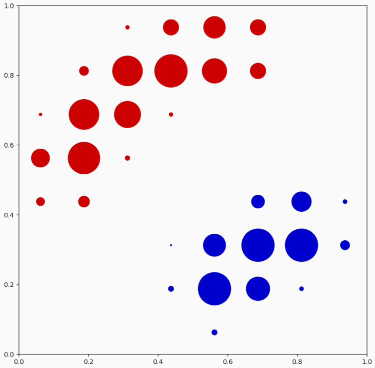




A baseline setting: density registration

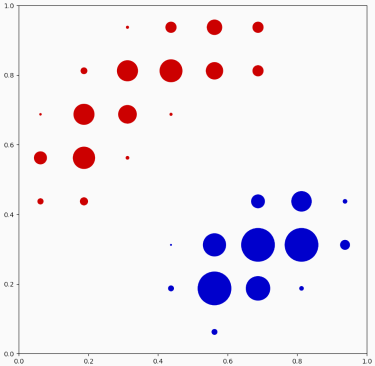


A baseline setting: density registration



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

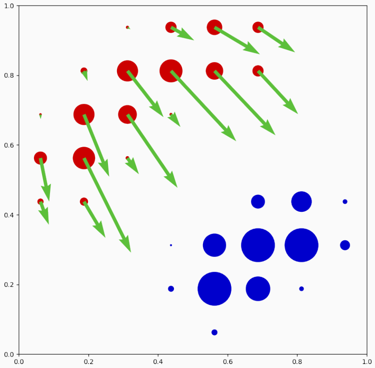
A baseline setting: density registration



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

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

A baseline setting: density registration

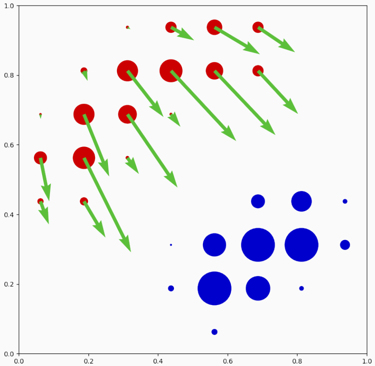


$$\alpha = \sum_{i=1}^{N} \alpha_i \delta_{x_i} \,, \quad \beta = \sum_{j=1}^{M} \beta_j \delta_{y_j} \,.$$

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

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

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 $\;v \;=\; -\nabla_{x_i} \mathrm{d}(\alpha, \beta)$.

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.

Overview

1. Computing fidelities between **measures**

Overview

1. Computing fidelities between **measures**
2. What's **new**, in 2018?

Overview

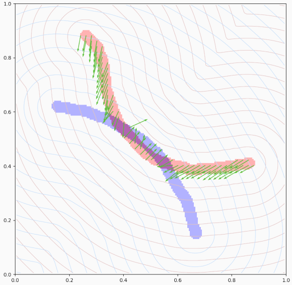
1. Computing fidelities between **measures**
2. What's **new**, in 2018?
3. Efficient GPU routines: **KeOps**

A simple formula: Hausdorff distance (aka. ICP, $\simeq$ GMM-MLE)

Define the fields

$$a(x) = \mathrm{d}(x, \mathrm{supp}(\alpha)) = \min_i \|x_i - x\|,$$

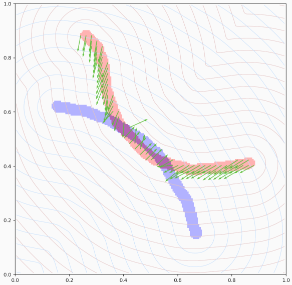
$$b(x) = \mathrm{d}(x, \mathrm{supp}(\beta)) = \min_j \|x - y_j\|,$$



A simple formula: Hausdorff distance (aka. ICP, $\simeq$ GMM-MLE)

Define the fields
$$a(x) = \mathrm{d}(x, \mathrm{supp}(\alpha)) = \min_i \|x_i - x\|,$$
$$b(x) = \mathrm{d}(x, \mathrm{supp}(\beta)) = \min_j \|x - y_j\|,$$

$$\begin{aligned}\mathrm{Loss}(\alpha, \beta) &= \tfrac{1}{2}\langle \alpha, b \rangle + \tfrac{1}{2}\langle \beta, a \rangle \\ &= \tfrac{1}{2}\sum_i \alpha_i \cdot \min_j \|x_i - y_j\| + \tfrac{1}{2}\sum_j \beta_j \cdot \min_i \|x_i - y_j\|\end{aligned}$$



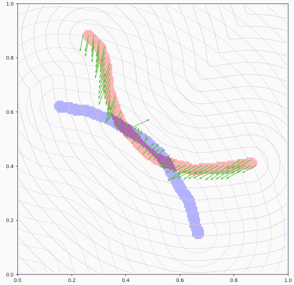
A simple formula: Hausdorff distance (aka. ICP, $\simeq$ GMM-MLE)

Define the fields

$$a(x) = \mathrm{d}(x, \mathrm{supp}(\alpha)) = \min_i \|x_i - x\|,$$

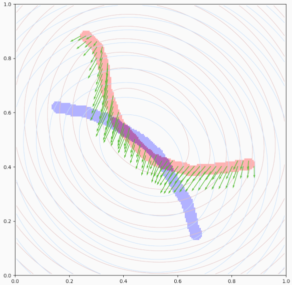
$$b(x) = \mathrm{d}(x, \mathrm{supp}(\beta)) = \min_j \|x - y_j\|,$$

$$\begin{aligned}\mathrm{Loss}(\alpha, \beta) &= \tfrac{1}{2}\langle \alpha, b \rangle + \tfrac{1}{2}\langle \beta, a \rangle = \tfrac{1}{2}\langle \alpha - \beta, b - a \rangle \\ &= \tfrac{1}{2}\sum_i \alpha_i \cdot \min_j \|x_i - y_j\| + \tfrac{1}{2}\sum_j \beta_j \cdot \min_i \|x_i - y_j\|\end{aligned}$$



A simple formula: Kernel norms (aka. MMD)

Define the fields
$$a(x) = \sum_i \alpha_i \|x - x_i\| = (\|\cdot\| \star \alpha)(x),$$
$$b(x) = \sum_j \beta_j \|x - y_j\| = (\|\cdot\| \star \beta)(x),$$



A simple formula: Kernel norms (aka. MMD)

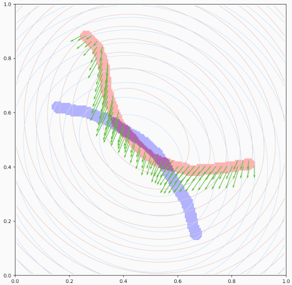
Define the fields

$$a(x) = \sum_i \alpha_i \|x - x_i\| = (\|\cdot\| \star \alpha)(x),$$

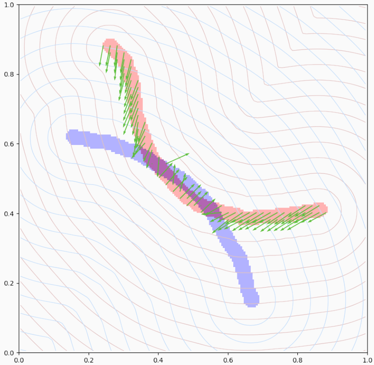
$$b(x) = \sum_j \beta_j \|x - y_j\| = (\|\cdot\| \star \beta)(x),$$

$$\text{Loss}(\alpha, \beta) = \tfrac{1}{2}\langle \alpha - \beta , b - a \rangle = \sum_i \sum_j \alpha_i \beta_j \|x_i - y_j\|$$

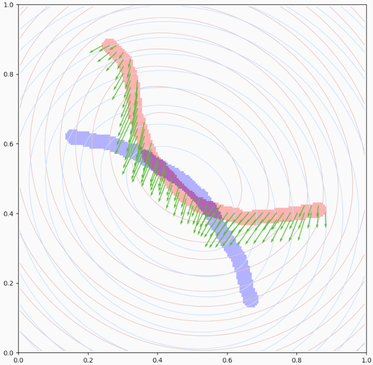
$$- \tfrac{1}{2} \sum_i \sum_j \alpha_i \alpha_j \|x_i - x_j\| - \tfrac{1}{2} \sum_i \sum_j \beta_i \beta_j \|y_i - y_j\|$$



The Hausdorff distance is local, the Energy Distance is global

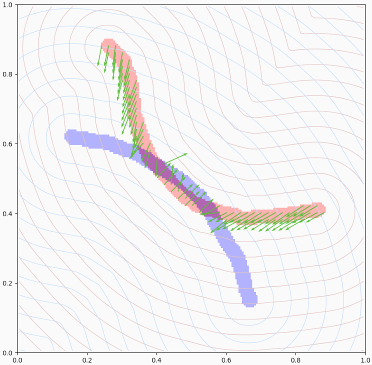


Hausdorff, min

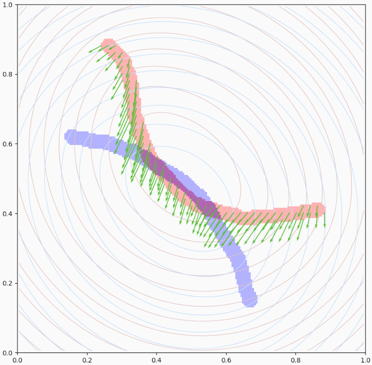


Kernel, $\sum$

The Hausdorff distance is local, the Energy Distance is global



Hausdorff, min

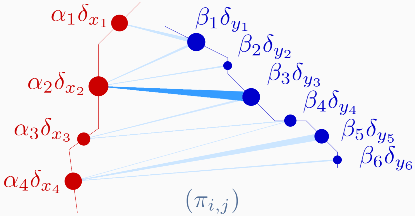


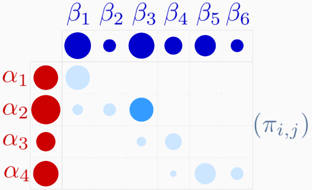
Kernel, $\sum$

$\Longrightarrow$ Can we get the best of both worlds?

Computational Optimal Transport

The Optimal Transport problem





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

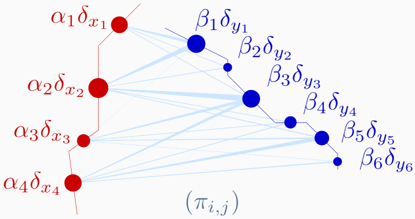
$$\mathrm{OT}(\alpha, \beta) \;=\; \min_{\pi} \underbrace{\sum_{i,j} \pi_{i,j} \cdot |x_i - y_j|^2}_{\text{transport cost}}$$

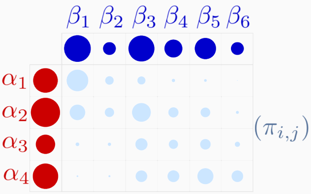
subject to $\quad \pi_{i,j} \geqslant 0$,

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

$\Rightarrow$ Hungarian method in $\mathbf{O}(\mathbf{N}^3)$.

Entropic regularization = add temperature, blur the transport plan





For $\varepsilon > 0$:

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

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

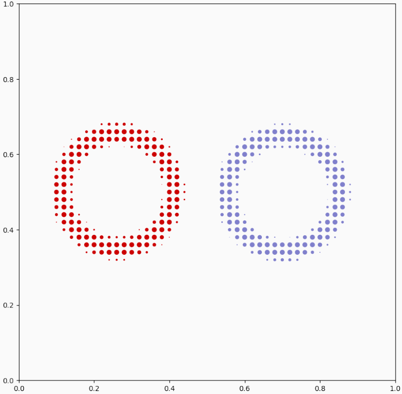
subject to

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

$\Rightarrow$ **Sinkhorn algorithm (GPU).**

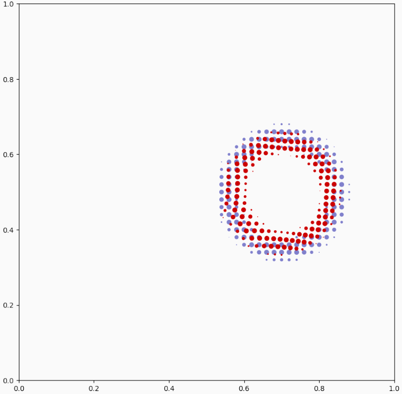
Problem : if $\varepsilon > 0$, OT_ε is *not* a valid divergence

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



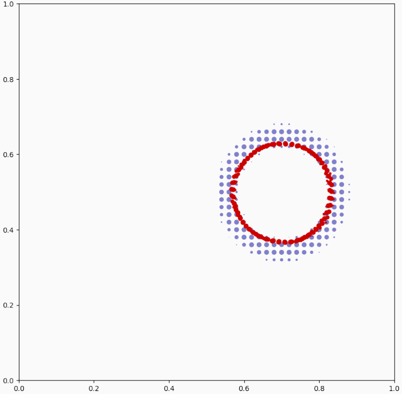
Problem : if $\varepsilon > 0$, OT_ε is *not* a valid divergence

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



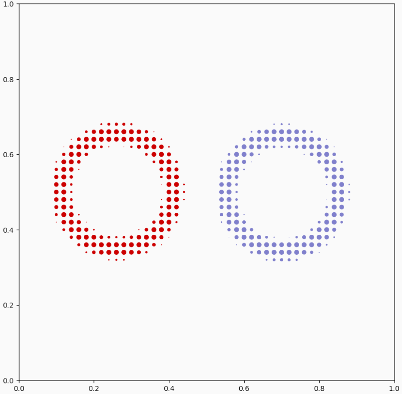
Problem : if $\varepsilon > 0$, OT_ε is *not* a valid divergence

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



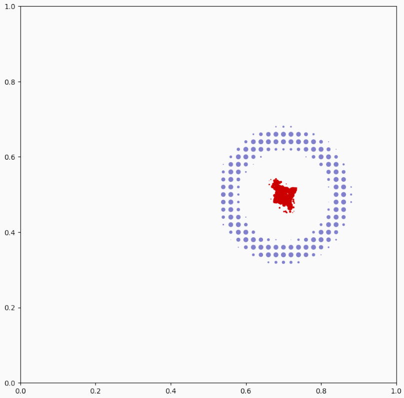
Problem : if $\varepsilon > 0$, OT_ε is *not* a valid divergence

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



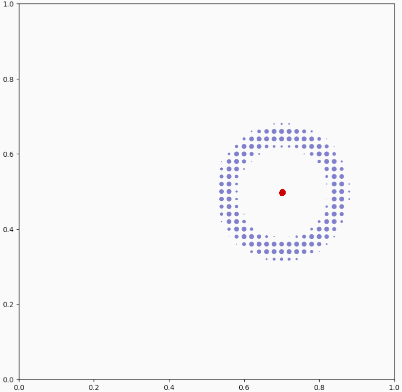
Problem : if $\varepsilon > 0$, OT_ε is *not* a valid divergence

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



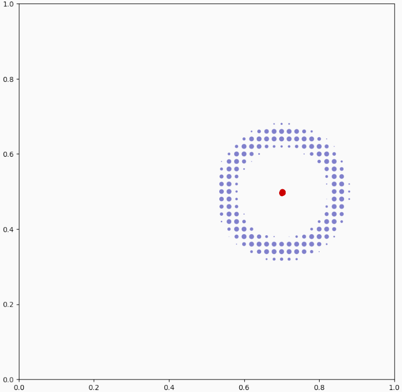
Problem : if $\varepsilon > 0$, OT_ε is *not* a valid divergence

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



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

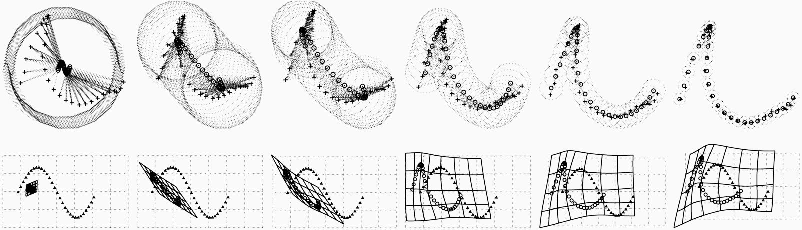
Registrating circles, $C(x,y) = \|x-y\|^2$, $\sqrt{\varepsilon} = 0.2$:



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

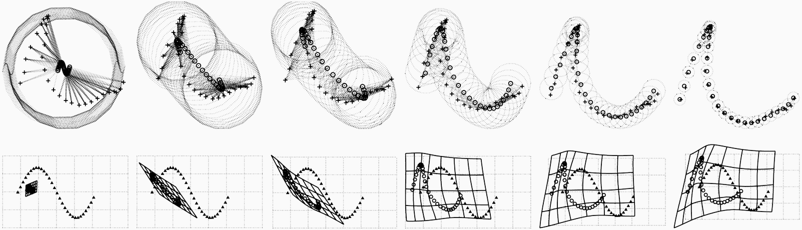
$$\mathrm{OT}_\varepsilon(\alpha, \beta) \;<\; \mathrm{OT}_\varepsilon(\beta, \beta).$$

Standard solution: use an annealing scheme



TPS-RPM algorithm, Chui and Rangarajan, CVPR 2000

Standard solution: use an annealing scheme



TPS-RPM algorithm, Chui and Rangarajan, CVPR 2000

$\Longrightarrow$ **Expensive** and cumbersome workaround,
with parameters to tune.

A new idea in 2017 : un-biased Sinkhorn divergences

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

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

A new idea in 2017 : un-biased Sinkhorn divergences

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

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

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

A new idea in 2017 : un-biased Sinkhorn divergences

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

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

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

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

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

A new idea in 2017 : un-biased Sinkhorn divergences

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

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

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

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

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

$$\text{Wasserstein}_{+\text{C}}(\alpha,\beta) \;\xleftarrow{\varepsilon\to 0}\; \text{S}_\varepsilon(\alpha,\beta) \;\xrightarrow{\varepsilon\to+\infty}\; \text{Kernel}_{-\text{C}}(\alpha,\beta)$$

A new idea in 2017 : un-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]:

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

$$\mathrm{Wasserstein}_{+\mathrm{C}}(\alpha,\beta) \xleftarrow{\varepsilon\to 0} \mathrm{S}_\varepsilon(\alpha,\beta) \xrightarrow{\varepsilon\to+\infty} \mathrm{Kernel}_{-\mathrm{C}}(\alpha,\beta)$$

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

In our paper(s): theoretical guarantees

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

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

In our paper(s): 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 paper(s): theoretical guarantees

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

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

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

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

In our paper(s): theoretical guarantees

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

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

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

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

*These results can be generalized to arbitrary **feature** spaces*
– e.g. (position,orientation,curvature).

In our paper(s): theoretical guarantees

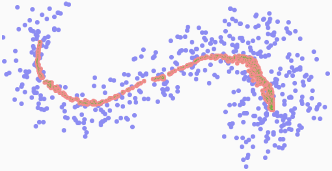
Theorem (F., Séjourné, Vialard, Amari, Trouvé, Peyré; 2018)

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

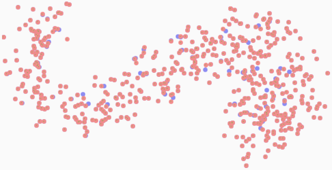
$$0 \leqslant S_\varepsilon(\alpha, \beta) \quad \text{with equality iff.} \quad \alpha = \beta$$

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

These results can be generalized to arbitrary ***feature*** *spaces*
– e.g. (position,orientation,curvature).



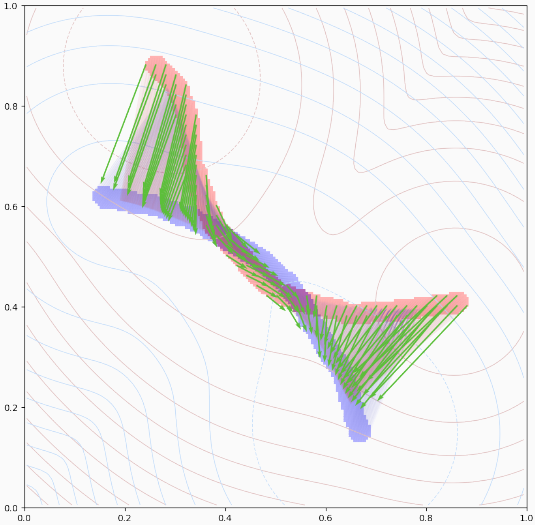
Loss $= \mathrm{OT}_\varepsilon$



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

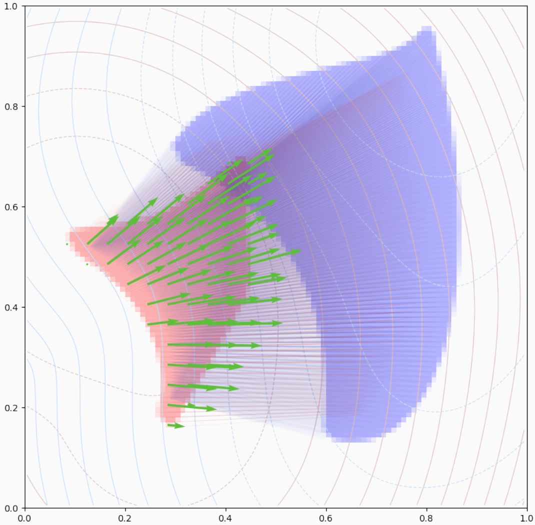
In practice

The ε-Sinkhorn divergence; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



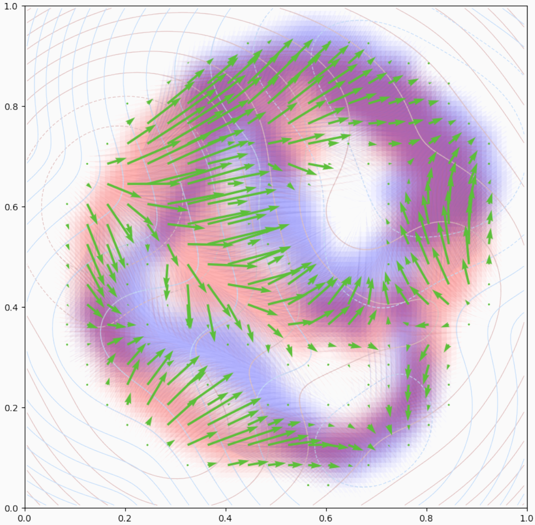
A high-quality gradient.

The ε-Sinkhorn divergence; with $\|x-y\|^2$ and $\sqrt{\varepsilon} = .1$



A high-quality gradient.

The ε-Sinkhorn divergence; with $\|x - y\|^2$ and $\sqrt{\varepsilon} = .1$



A high-quality gradient?

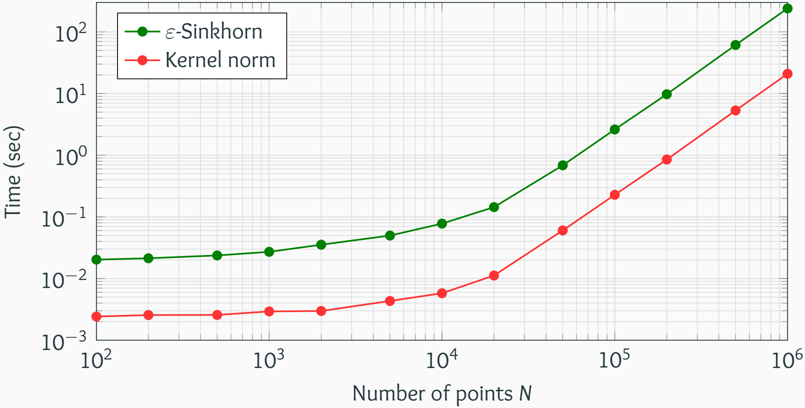
(Data from the Spectral Log-Demons paper.)

KErnel OPerationS, with autodiff, without memory overflows

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

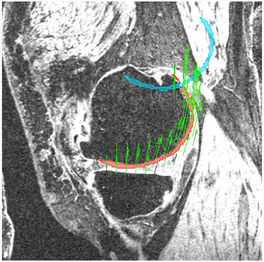
(Thanks Benjamin and Joan!)

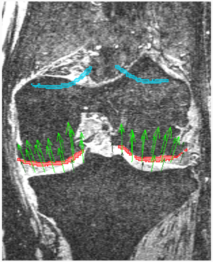
Fidelity + gradient with N vertices on a **high-end GPU** (Tesla P100)

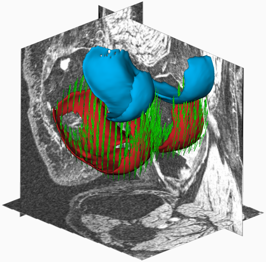


We provide a reference PyTorch implementation

github.com/jeanfeydy/global-divergences.







Gradient of the Energy Distance, computed in 0.5s on my laptop.
Data from the OsteoArthritris Initiative:
52,319 and 34,966 voxels out of a 192-192-160 volume.

Conclusion

Global, **geometry-aware** loss functions are easy to compute.

Conclusion

Global, **geometry-aware** loss functions are easy to compute.

- Try using $k(x, y) \; = \; -\|x - y\|$!

Conclusion

Global, **geometry-aware** loss functions are easy to compute.

- Try using $k(x, y) = -\|x - y\|$!
- Remove the **entropic bias** from the SoftAssign algorithm!

Conclusion

Global, **geometry-aware** loss functions are easy to compute.

- Try using $k(x, y) = -\|x - y\|$!
- Remove the **entropic bias** from the SoftAssign algorithm!
- Sinkhorn $=$ Hausdorff + mass **spreading** constraint
  $\simeq$ best you can do without topology or landmarks
  $\simeq$ 20-50 convolutions through the data
  $\rightarrow$ Is it worth it?

Conclusion

Open questions:

- Rigorous link with the **auction algorithm**?

Conclusion

Open questions:

- Rigorous link with the **auction algorithm**?
- Link between S_ε and Sobolev **distances**?

Conclusion

Open questions:

- Rigorous link with the **auction algorithm**?
- Link between S_ε and Sobolev **distances**?
- What about **multiscale** schemes?

Conclusion

Open questions:

- Rigorous link with the **auction algorithm**?
- Link between S_ε and Sobolev **distances**?
- What about **multiscale** schemes?
- Interest in the **CVPR/SIGGRAPH** communities?

Thank you for your attention.

Any questions ?

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

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.

Cuturi, M. (2013).
Sinkhorn distances: Lightspeed computation of optimal transport.
In *Advances in neural information processing systems*, pages 2292–2300.

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)*.

References iii

Peyré, G. and Cuturi, M. (2018).
Computational optimal transport.
arXiv preprint arXiv:1803.00567.

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.