Computational Optimal Transport – MVA

Numerical reports

Rodrigue Lelotte

1. Optimal transport and linear programming

This numerical tours details how to solve the discrete optimal transport problem using linear programming. Formally, given two sets of points $\{x_i\}_{i=1,\dots,n}$ and $\{y_j\}_{j=1,\dots,m}$ in $\mathbb{R}^d$ together with weights $a = (a_i)_{i=1,\dots,n} \in K^n$ and $b = (b_j)_{j=1,\dots,m} \in K^m$ where $K^l = \{(a_k)_{k=1,\dots,l} \in \mathbb{R}_+^l : \sum_{k=1}^l a_k = 1\}$, we consider the optimal transport problem between the discrete measures

$$\alpha := \sum_{i=1}^{n} a_i \delta_{x_i}, \quad \beta := \sum_{j=1}^{m} b_j \delta_{y_j}.$$

1.1. Theoretical background

Given the set $\mathcal{U}(\alpha, \beta)$ of *couplings* between α and β, *i.e.*

$$\mathcal{U}(\alpha, \beta) = \{P \in \mathbb{R}_+^{n \times m} : P\mathbb{1}_m = a \text{ and } \mathbb{1}_n^\intercal P = b\},$$

and a *cost matrix* $C \in \mathbb{R}_+^{n \times m}$, where $C_{ij} \geq 0$ expresses the cost of matching point x_i with point y_j (*e.g.* $C_{ij} = d(x_i, y_j)$ where d is a distance on $\mathbb{R}^d$) the Kantorovitch formulation of the optimal transport reads

$$P^\star \in \arg\min_{P \in \mathcal{U}(\alpha,\beta)} \langle P, C \rangle \triangleq \sum_{i,j} P_{i,j} C_{i,j}. \tag{1.1}$$

The set $\mathcal{U}(\alpha, \beta)$ is non-empty (*i.e.* $ab^\intercal \in \mathcal{U}(\alpha, \beta)$), convex and compact. Since the functional $P \mapsto \langle P, C \rangle$ is linear (therefore continuous), we know that the optimization problem (1.1) (which is a *linear program*) admits at least one solution. In the case where $n = m$ and the weights are uniform, *i.e.* $a = b = \mathbb{1}_n/n$, at least one of these solutions is a *permutation matrix* (and therefore a solution to the underlying Monge formulation).

1.2. Numerical experiments

Let $X = \{x_i\}_{i=1,\dots,n}$ be a sample of size $n = 60$ drawn from a uniform distribution on the unit ball $\{x \in \mathbb{R}^2 : \|x\| \leq 1\}$, and $Y = \{y_j\}_{j=1,\dots,m}$ a sample of size $m = 120$ drawn from a uniform distribution on the annulus $\{x \in \mathbb{R}^2 : 3 \leq \|x\|_2 \leq 4\}$ (fig. 1). The weights a (resp. b) are randomly selected in K^n (resp. K^m).

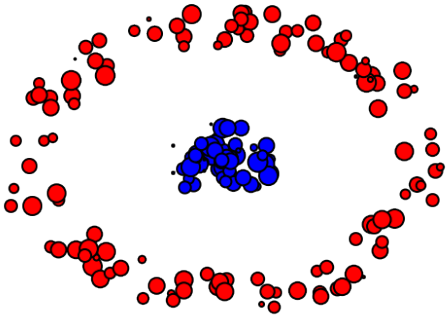

Figure 1: Datasets X (blue) and Y (red) .

An optimal transportation plan $P^\star$ (fig. 4) with respect to the cost $c(x, y) = \|x - y\|_2$ is found by solving the linear program (1.1) using `cvxpy` package. We display the (many-to-one) matching between α and β in fig. 9. More precisely, we draw an edge (x_i, y_j) if $P_{ij}^\star > \eta$, where η is some threshold value (*i.e.* $\eta = 1e^{-5}$).

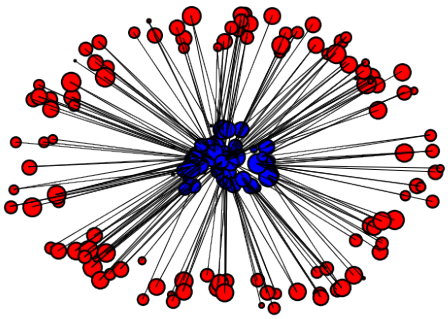

Figure 2: Matching between α and β

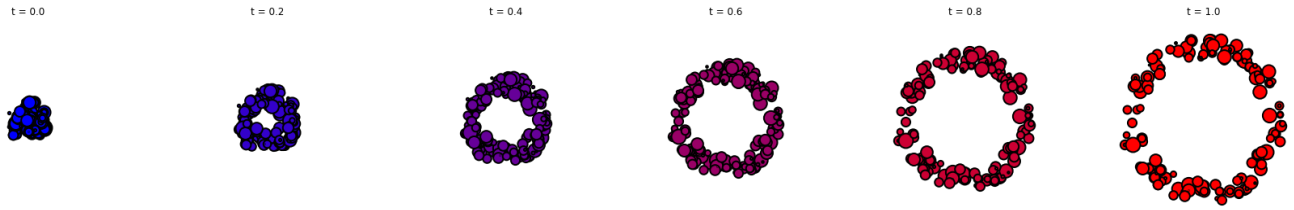

Figure 3: Interpolation between α and β along the geodesic in the Wasserstein space.

We can visualize the matching in a more dynamical way. Indeed, recall that the *Wasserstein space* $\mathcal{W}_2(Y)$ on a Polish metric space (Y, d) is defined as the set of Borel probability measures μ with finite second moment endowed with the *Wasserstein metric* W_2, *i.e.*

$$W_2(\mu,\nu) = \inf_{\pi\in\Pi(\mu,\nu)} \int_{Y\times Y} d^2(x,y)\pi(\mathrm{d}x,\mathrm{d}y),$$

where $\Pi(\mu,\nu)$ is the set of couplings between μ and ν. This space has several nice properties, among which one states that $\mathcal{W}_2(Y)$ is a *geodesic space* as soon as Y is.

In the case where (Y, d) is $\mathbb{R}^d$ endowed with the euclidian distance, the (unique) geodesic between δ_x and δ_y reads $[0,1] \ni t \mapsto \delta_{(1-t)x+ty}$. As such, the (unique) geodesic between α and β reads

$$[0,1] \ni t \mapsto \sum_{i,j} P^\star_{i,j}\delta_{(1-t)x_i+ty_j}.$$

The interpolation obtained in the case of our datasets is displayed (fig. 3) for different values of t.

Remark 1.1. *Along the way, to validate the overall correctness of the optimization, we can check that the number of non-zeros entries (*i.e. *greater that some threshold value η) in the optimal plan $P^\star$ is at most $n+m-1$ (see [3, Proposition 3.3]).*

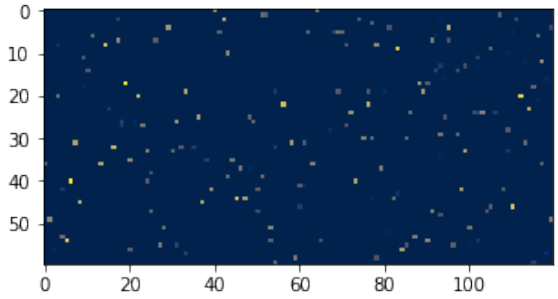

Figure 4: A color-map of $P^\star$.

As mentioned earlier, in the case where $n = m$ and $a = b = \mathbb{1}_n/n$, we expect $P^\star$ to be a permutation matrix, meaning that there exists a permutation $\sigma^\star \in \mathfrak{S}_n$ such that $P^\star_{ij} = 1$ if $j = \sigma^\star(i)$ and 0 otherwise.

To vary a little bit the flavors, let us work with another example using the *repulsive cost* $c(x,y) = \|x-y\|_2^{-1}$. Given α the uniform discrete measure on a sample $X = \{x_i\}_{i=1,\dots,n}$ of size $n = 100$ drawn from a uniform distribution on the annulus $\{x \in \mathbb{R}^2 : 5 \le \|x\|_2 \le 5.5\}$, we want to send α back *onto itself*, *i.e.* find the permutation $\sigma^\star : X \to X$ such that

$$\sigma^\star \in \arg\min_{\sigma:X\to X} \sum_{i=1}^{n} \frac{1}{\|x_i - \sigma(x_i)\|_2}. \qquad \text{(M)}$$

Qualitatively, we want to send each point as *farther out* as possible from itself (note that $\sigma^\star$ cannot have a fixed point, since $c(x,x) = +\infty$ for all x). Therefore, we roughly expect antipodal points to be matched together.

The Kantorovitch relaxation of the (M) reads as in (1.1) with $C_{ij} = \|x_i - x_j\|_2^{-1} \in \mathbb{R}_+ \cup \{+\infty\}$. From a numerical perspective, `cvxpy` does not seem to handle infinite quantities well. To bypass this small issue, we manually add in the constraints of the optimization problem that $P_{ii} = 0$ for all $i = 1,\dots,n$ (i.e. `cp.diag(P) == 0`). Remark that the feasible set remains non-empty, convex and compact.

Remark 1.2. *Because of the newly added constraint, we had to switch to the solver `ECOS`, for the default one was unable to achieve a proper minimization.*

As expected, the optimal plan $P^\star$ found by the solver is a permutation matrix (fig. 5). The associated optimal matching $\sigma^\star$ (fig. 6) is seen to confirm our intuition, namely that $\sigma^\star$ is a product of transpositions matching « antipodal » points together, *i.e.* $\sigma^\star = \prod_{i=1}^{n/2}(i\sigma_i)$ where x_i and x_{σ_i} are roughly antipodal.

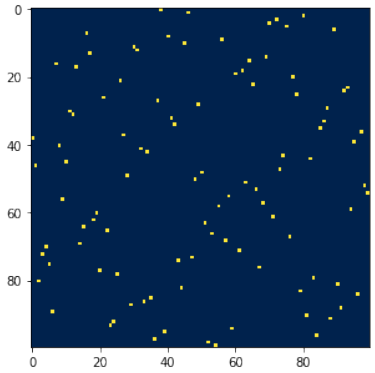

Figure 5: A color-map of $P^\star$

Remark 1.3. *Contrary to the preceding case, there are no such thing as a geodesic between two discrete measures when we consider the repulsive cost $c(x,y) = \|x-y\|_2^{-1}$. In fact, the associated optimal transport doesn't even define a quasi-metric, for c fails to verify the triangular inequality (see [3, Proposition 2.2]).*

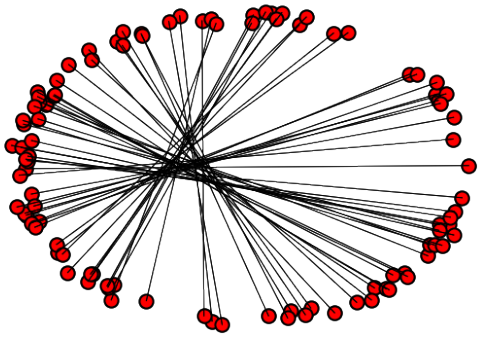

Figure 6: Optimal matching $\sigma^\star$

2. Entropic regularization of optimal transport

This numerical tours exposes the general methodology of regularizing the optimal transport linear program using entropy. This allows to derive fast computation algorithms based on iterative projections according to a Kullback-Leiber divergence.

2.1. Theoretical background

We modify the Kantorovitch problem (1.1) by adding an entropic regularization, *i.e.*

$$P^\varepsilon \in \arg\min_{P\in\mathcal{U}(\alpha,\beta)} \langle P, C\rangle - \varepsilon \mathrm{Ent}(P), \tag{2.1}$$

where $\mathrm{Ent}(P) := -\sum_{i,j} P_{i,j}(\log(P_{i,j})-1)$ is the entropy of the coupling P. This makes the objective ε-strongly convex, which in turn makes the solution to (2.1) unique. This unique solution P^ε converges to the optimal solution with maximal entropy within the set of all optimal solutions of the Kantorovich problem (1.1) as $\varepsilon \to 0$ ([3, Proposition 4.1]).

By defining the *Gibbs kernel* $K^\varepsilon = (K^\varepsilon_{i,j})_{i,j}$ with $K^\varepsilon_{i,j} := \exp(-C_{i,j}/\varepsilon)$, the problem (2.1) rewrites as

$$\min_{P\in\mathcal{U}(\alpha,\beta)} \mathrm{KL}(P\|K^\varepsilon), \tag{2.2}$$

where $\mathrm{KL}(P\|Q) := -\sum_{i,j} P_{i,j}\left(\log\frac{P_{i,j}}{Q_{i,j}}-1\right)$ is the Kullback-Leibler divergence. It can be proven [3, Proposition 4.3] that there exist two scaling vectors $(u,v)\in\mathbb{R}_+^n\times\mathbb{R}_+^m$ such that P^ε the unique solution to (2.2) reads $P^\varepsilon = \Delta(u)K^\varepsilon\Delta(v)$, where $\Delta(w)$ is the diagonal matrix with diagonal w. The variables (u,v) must satisfy the mass conservation constraints, *i.e.*

$$u\odot(Kv) = a, \quad v\odot(K^\intercal u) = b,$$

where $\odot$ corresponds to the entry-wise multiplication. Solving both equalities iteratively leads to the so-called *Sinkhorn's algorithm*, *i.e.*

$$u^{(k+1)} = \frac{a}{Kv^{(k)}}, \quad v^{(k+1)} = \frac{b}{K^\intercal u^{(k+1)}}.$$

This iterative procedure can be seen as particular case of *Bregman iterative projections*, and will therefore converge to a solution $(u^\star, v^\star)$ that verifies the mass conservation constraints ([3, Remark 4.8]).

2.2. Numerical experiments

Let $X = \{x_i\}_{i=1,\dots,n}$ be a sample of size $n = 400$ drawn from a uniform distribution on the union of five balls $\cup_{i=1}^5 B(c_i, \frac{1}{5})$ where the c_i's are the origin and the four vertices of the unit square, and $Y = \{y_j\}_{j=1,\dots,m}$ a sample of size $m = n = 400$ drawn from a uniform distribution of the annulus $\{x\in\mathbb{R}^2 : \frac{3}{5}\le\|x\|_2\le\frac{4}{5}\}$ (fig. 7). The weights are selected uniformly, *i.e.* $a = b = \mathbb{1}_n/n$.

We implement the Sinkhorn's algorithm with $\varepsilon = 0.005$. We plot the ℓ^1-errors $\|\overline{P}^{(k)}\mathbb{1}_m - a\|_1$ and $\|P^{(k),\intercal}\mathbb{1}_n - b\|_1$ along the iterations (fig. 8), where $P^{(k)} = \Delta(u^{(k)})K\Delta(v^{(k)})$ and $\overline{P}^{(k)} = \Delta(u^{(k)})K\Delta(v^{(k+1)})$.

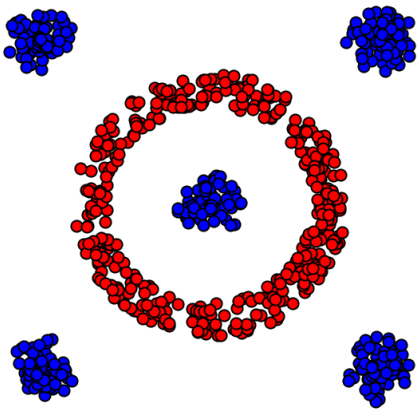

Figure 7: Datasets X (blue) and Y (red).

Note that we don't need to compute these matrices at each iteration, since the errors rewrite as

$$\begin{aligned}\mathrm{err}_a^{(k)} &:= u^{(k)} \odot K v^{(k+1)} - a,\\ \mathrm{err}_b^{(k)} &:= v^{(k)} \odot K^\intercal u^{(k)} - b.\end{aligned}$$

Remark 2.1. *These quantities provide useful stopping criteria to monitor the convergence. Indeed, one has*

$$\|\log(P^{(k)}) - \log(P^\varepsilon)\|_\infty \leq C_k \left(\|err_a^{(k)}\|_1 + \|err_b^{(k)}\|_1 \right),$$

where $C_k = \frac{2\widetilde{C}_k}{1-\lambda(K)^2}$ *with* $\lambda(K) \in (0,1)$ *and* $\widetilde{C}_k = \mathcal{O}(1)$. *See [3, Remark 4.14] for explicit derivation.*

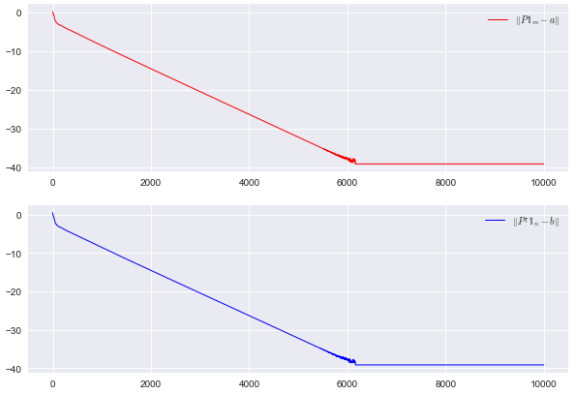

Figure 8: Errors $\|\overline{P}^{(k)}\mathbb{1}_m - a\|_1$ (red) and $\|P^{(k),\intercal}\mathbb{1}_n - b\|_1$ (blue) in log-plot for $k = 1, \ldots, 10000$.

To numerically access the role of ε, we compute P^ε for several values of $\varepsilon > 0$ (fig. 10). As excepted, the bigger ε is the *blurrier* the optimal coupling matrix is (from a theoretical perspective, we have $P^\varepsilon \to ab^\intercal$ as $\varepsilon \to \infty$ [3, Proposition 4.3]), and conversely, the smaller ε is the *sparser* the optimal coupling matrix gets.

Remark 2.2. *We don't necessarily have that* P^ε *converges to the optimal* permutation *as* $\varepsilon \to 0$, *for this permutation might have lower entropy that some optimal transportation plans with mass-splitting.*

Remark 2.3. *When* ε *gets to small, the entries in the scaling vectors along the iterations become so small that we experience* underflow. *A way to prevent this is to store the log-values instead of the values (see section 4).*

We can display the approximate optimal transport plan for $\varepsilon = 0.005$ (fig. 9). Plain lines corresponds strong connections, *i.e.* edges (x_i, y_j) such that $P_{ij}^\varepsilon > \eta_M$ and dashed lines corresponds to weaker connections, *i.e.* edges (x_i, y_j) such that $P_{ij}^\varepsilon > \eta_m$ with $\eta_M > \eta_m$.

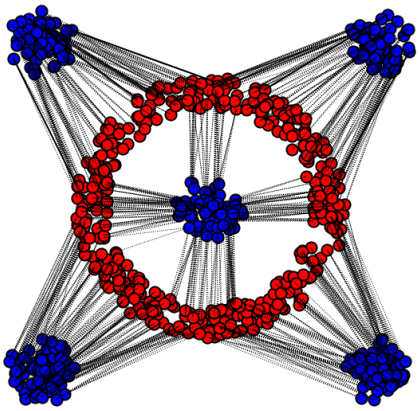

Figure 9: Approximate optimal matching ($\varepsilon = 0.005$) with $\eta_M = \frac{7}{10}\xi$ and $\eta_m = \frac{1}{5}\xi$ where $\xi = \max_{ij} P_{ij}^\varepsilon$.

A natural question that now arises is the following : is the trade-off between computational burden and accuracy a bargain ? By computing the (exact) optimal permutation $\sigma^\star$ using linear programming as before, we find that the overall execution time is roughly 4 to 5 times higher with this strategy compared to the iterative procedure. When it comes to accuracy, Sinkhorn's algorithm achieves a rather relatively good performance. For instance, the number of points that are correctly matched, *i.e.* $\mathrm{Card}(\{i : \arg\max_j P_{ij}^\varepsilon = \sigma^\star(i)\})$, is roughly equal to $n/2$, and the mismatched points are nonetheless « good-enough » solutions.

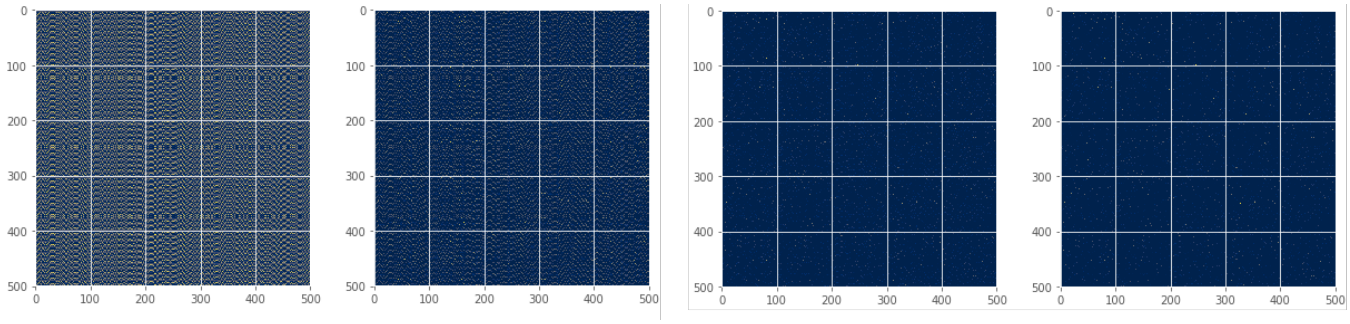

Figure 10: Color-maps of P^ε for $\varepsilon \in \{1, 0.1, 0.01, 0.005\}$ (with 5000 inner iterations).

Let us now experiment with *histograms*. To motivate what follows, consider the MRI images of a brain (fig. 11). The left one I_b represents a « baseline » sagittal section, and the right one I_m is another section for which we have artificially lowered the *contrast*. This kind of modification pertains to concrete issues in medical imaging (*e.g. bias field correction* for MRI images). We want to correct the contrast of the right image so that it matches the one from the baseline.

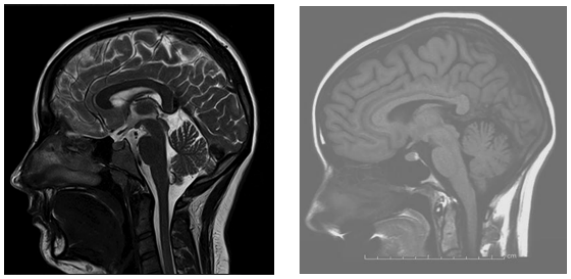

Figure 11: MRI images. Baseline (left), and modified (right).

To do so, we want to compute an optimal transport plan between the *normalized histograms* of I_b and I_m, where in this context the normalized histogram of an gray-scaled image I of size $w \times h$ is defined as the discrete measure

$$h_I = \frac{1}{wh} \sum_{i=0}^{255} \text{Card}\{(x, y) : I(x, y) = i\}\delta_i.$$

We run the Sinkhorn's algorithm with $\varepsilon = 0.0005$ and cost $c(i, j) = [(i - j)/255]^2$ (the histograms and the plots of the log-ℓ_1-errors are available in the notebook). Since P^ε is not (necessarily) a permutation (*i.e.* it is possibly *one-to-many* correspondence), we need to find a « natural » candidate for the transformation $i \mapsto T(i)$ such that $h_{T(I_m)} \approx h_{I_b}$. For instance, we can take $T(i) = \arg\max_j P^\varepsilon_{ij}$. Here, we rather take the mean, *i.e.*

$$\overline{T}(i) = \sum_j P^\varepsilon_{ij} j / (\sum_j P^\varepsilon_{ij}).$$

We display the optimal P^ε together with the superimposed mean (*i.e.* *barycentric projection map*) in red (fig. 12).

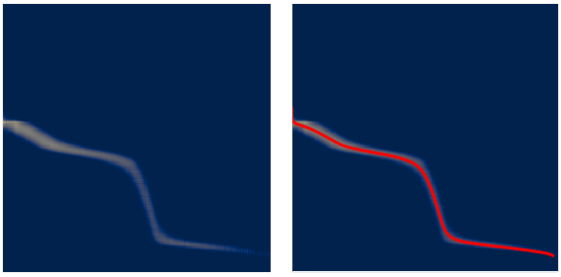

Figure 12: A color-map of P^ε together with $i \mapsto \overline{T}(i)$.

We then apply the transformation $\overline{T}$ to the image I_m (more precisely, we apply $[\overline{T}]$ where $[\cdot]$ is the nearest integer). We see that the overall contrast of the resulting image in very similar to the baseline (fig. 13).

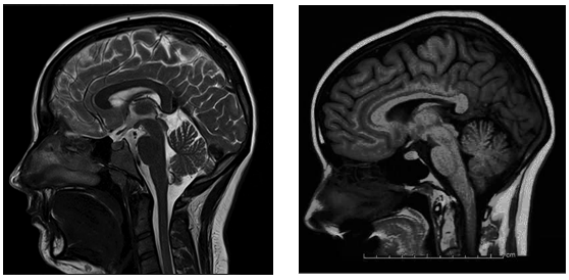

Figure 13: I_b (left) and $\overline{T}(I_m)$ (right).

Our last experiment concerns the following problem: given M measures $(\mu_k)_{k=1,\dots,M}$, we want to find the corresponding *Wasserstein barycenter* $\overline{b}$ (according to some

cost c), which we *define* as

$$\overline{b} \in \arg\min_b \sum_{k=1}^{M} \lambda_k W_c(\mu_k, b), \tag{2.3}$$

where λ_k are positive weights such that $\sum_{k=1}^{M} \lambda_k = 1$.

Remark 2.4. *Problems like* (2.3) *are very common in the literature of statistical geometry. In fact, one can see* (2.3) *as a particular case of* Fréchet mean. *For instance, in the context of* diffusion MRI *in medical imaging, one wants to estimate the diffusion tensor map $M \ni p \mapsto \Sigma_p$ (*i.e. *M is a Riemannian manifold which serves as a mathematical ansatz of the brain) from a set of acquisitions $\{\Sigma_{p_1} \ldots, \ldots, \Sigma_{p_n}\}$ on specific voxels. A way to go about this problem is to approximate the tensor outside the discretization grid as a barycenter w.r.t. the underlying Riemannian metrics.*

We're interested in the entropic approximation of (2.3). We suppose that μ_k is a discrete measure with weights $b_k \in K^{n_k}$ for all k, and that $\overline{b}$ is of the form $\frac{1}{n}\sum_{i=1}^{n} a_i \delta_{x_i}$. Then, we look at the problem

$$\min_{(P_k)_k} \sum_k \lambda_k \mathrm{KL}(P_k \| K_k), \tag{2.4}$$

where the minimization runs over the set of all couplings such that $P_k^\intercal \mathbb{1}_n = b_k$ and $P_k \mathbb{1}_{n_k} = a$, and where $K_k := e^{-C_k/\varepsilon}$ for all $k = 1, \ldots, M$ (C_k is the cost matrix between $\overline{b}$ and μ_k). For we know that the optimal couplings P_k^ε's rewrite as $P_k^\varepsilon = \Delta(u_k) K_k \Delta(v_k)$ for some scaling vectors (u_k, v_k), and idea to solve (2.4) is to iterate Sinkhorn's procedure as follows :

$$v_k^{(t+1)} = \frac{b_k}{K_k^\intercal u_k^{(t)}},$$
$$u_k^{(t+1)} = \frac{a^{(t+1)}}{K_k v_k^{(t+1)}},$$

for all $k = 1, \ldots, M$ and all time t, where $a^{(t+1)} := \prod_{k=1}^{M} (K_k v_k^{(t+1)})^{\lambda_k}$.

Reminiscent of the above remark, we take four random tensors (i.e. symmetric positive definite matrices), and we interpolate them w.r.t. $c(x, y) = \|x - y\|_2$. We run the algorithm mentioned above with $\varepsilon = 0.001$ across 1000 iterations (fig. 14). Of course, the resulting barycenters are no longer tensors, because the geodesic are unconstrained, *i.e.* they live in the entire space of *shapes*. It would therefore be interesting to see if one can generalize the above algorithm in the case of constrained shapes.

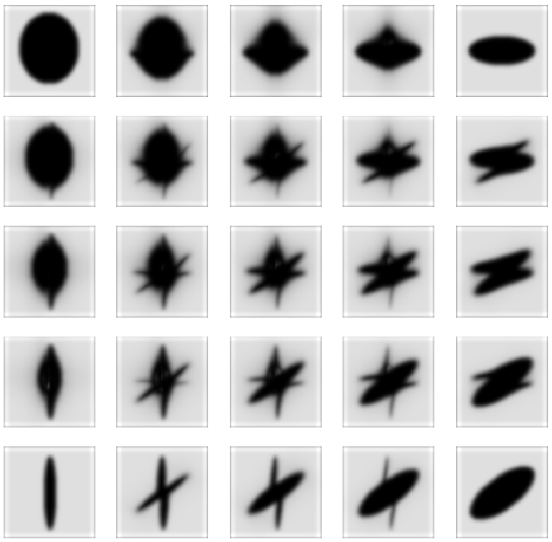

Figure 14: Interpolation using Wasserstein barycenters.

3. Semi-discrete optimal transport

This numerical tour studies *semi-discrete* optimal transport, *i.e.* when one of the two measure is discrete.

3.1. Theoretical background

In the very general setting where $(\mathcal{X}, \alpha)$ and $(\mathcal{Y}, \beta)$ are two (Polish) probability spaces and $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} \cup \{+\infty\}$ is some abstract (lower semicontinuous) cost, then *Kantorovich duality* holds [4, Theorem 5.9], that is the primal solution to Kantorovitch formulation of optimal transport, *i.e.*

$$W_c(\alpha, \beta) := \min_{\pi \in \Pi(\alpha, \beta)} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) \pi(\mathrm{d}x, \mathrm{d}y),$$

is equal to

$$\sup_{(\phi, \psi) \in \mathcal{R}_c} \int_{\mathcal{X}} \phi(x) \alpha(\mathrm{d}x) + \int_{\mathcal{Y}} \psi(y) \beta(\mathrm{d}y), \tag{3.1}$$

where $\mathcal{R}_c$ is the set of bounded and continuous *potentials*, *i.e.* functions that verify $\phi(x) + \psi(y) \leq c(x, y)$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Furthermore, by fixing ϕ, we see that the supremum at (3.1) is achieved at $\psi = \phi^c$, where ϕ^c is the *c-transform* of ϕ, which we recall to be defined as

$$\phi^c(y) := \inf_{x \in \mathcal{X}} c(x, y) - \phi(x).$$

In particular, the problem (3.1) rewrites as

$$\sup_{\phi} \mathcal{E}(\phi) \triangleq \int_{\mathcal{X}} \phi(x)\alpha(\mathrm{d}x) + \int_{\mathcal{Y}} \phi^c(y)\beta(\mathrm{d}y) \qquad (3.2)$$

and when the measure α is discrete, *i.e.* $\alpha = \sum_{i=1}^n a_i \delta_{x_i}$, the $c-$transform reads $\phi^c(y) = \min_i c(x_i, y) - \phi_i$ where $\phi_i := \phi(x_i)$, and the functional $\mathcal{E}$ rewrites as

$$\mathcal{E}(\phi) = \sum_{i=1}^n a_i\phi_i + \sum_{i=1}^n \int_{\mathbb{L}_i(\phi)} (c(x_i, y) - \phi_i)\,\beta(\mathrm{d}y),$$

for all $\phi \in \mathbb{R}^n$ where the $\mathbb{L}_i(\phi)$'s are the *Laguerre cells*, defined as $\mathbb{L}_i(\phi) := \{y \in \mathcal{Y} : i = \arg\min_j c(x_j, y) - \phi_j\}$, which correspond to *Voronoi cells* when ϕ is constant. In particular, the gradient of $\mathcal{E}$ reads

$$\nabla\mathcal{E}(\phi)_i = a_i - \beta(\mathbb{L}_i(\phi)), \quad i = 1, \dots, n.$$

3.2. *Numerical experiments*

To solve (3.2) numerically, a natural way to go is to perform a *gradient ascent* as we have an explicit formula for the gradient of $\mathcal{E}$. Given a learning rate schedule $\tau(t)$, we iteratively update

$$\phi^{(t+1)} \leftarrow \phi^{(t)} + \tau(t)\nabla\mathcal{E}(\phi^{(t)}).$$

To compute the gradient $\nabla\mathcal{E}(\phi)$, we only need to determine the Laguerre cells, whose geometries only depends on c (and the current dual variable $\phi^{(t)}$). For fun, we display the Voronoi cells (*i.e.* $\phi \equiv 0$) given four arbitrarily chosen points in the unit square $[0, 1]^2$, for the L^p-metric with $p = 1, 2, \infty$ and the Poincaré metric on the upper-half complex plane, *i.e.* $\rho(z, w) = 2\mathrm{arctanh}\left(\frac{|z-w|}{|z-\overline{w}|}\right)$ (fig. 15).

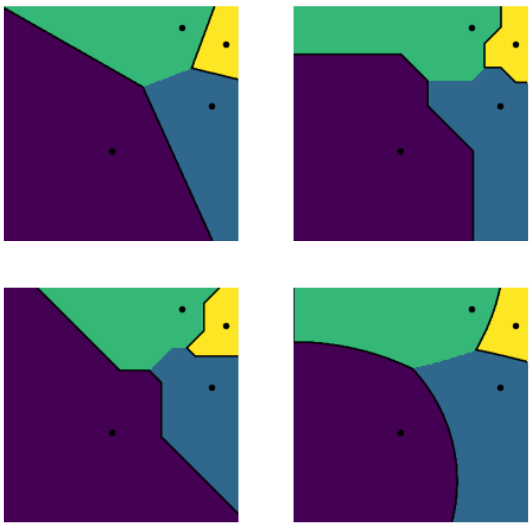

Figure 15: Voronoi cells for L^2 (top left), L^1 (top right), L^∞ (bottom left) and the Poincaré metric (bottom right).

Let α be the discrete uniform measure on a sample $X = \{x_i\}_{i=1,\dots,n}$ of size $n = 30$ drawn from a uniform distribution on $B((\frac{1}{2}, \frac{1}{2}), \frac{2}{5})$ and β be the continuous distribution constructed from four Gaussian *bumps* located near the coins of the unit square with random weights and deviations (numerically, β is discretized on a mesh-grid of size 300×300) (fig. 16). The cost used is $c(x, y) = \|x - y\|_2$.

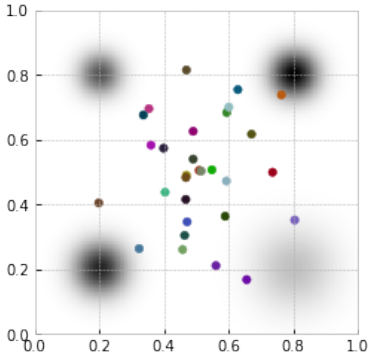

Figure 16: α and β.

Choosing $\tau(t)$ is always both a subtle and coarse task. If this learning rate is too high, we expect oscillations around local minima for the learning is unable to weave its way through to the narrow wells. If $\tau(t)$ is too small, it might take too many iterations to converge.

We perform the gradient ascent across 300 iterations with starting point $\phi^{(0)} \equiv 0$, for the learning rate schedules $\tau_1(t) = \frac{1}{(1+t)^{1/4}}$, $\tau_2(t) = \frac{1}{(1+t)}$ and $\tau_3(t) = \frac{1}{(1+t)^2}$. As expected, τ_1 does not decay quickly enough, and the learning displays oscillations; τ_3 decay too quickly and is unable to achieve a proper optimization with so few iterations; eventually, τ_2 seems to be the best trade-off (fig. 17). We display the Laguerre cells for different steps across the ascent (with τ_2) (fig. 18).

Remark 3.1. *A class of learning schedules that was proved to be rather efficient is given by the learning rates that satisfy the* Robbins-Monroe conditions, i.e.

$$\sum_{t\geq 0} \tau(t) = +\infty, \quad \sum_{t\geq 0} \tau(t)^2 < \infty.$$

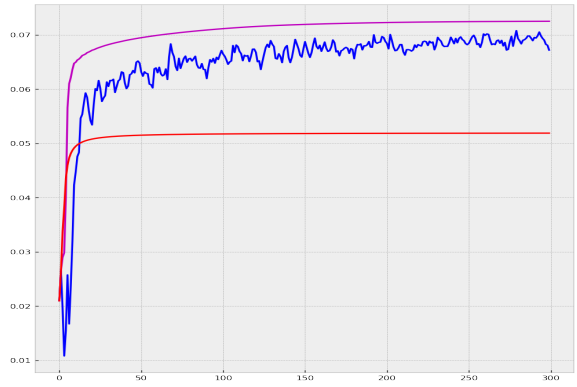

Figure 17: τ_1 (blue), τ_2 (fuchsia), and τ_3 (red).

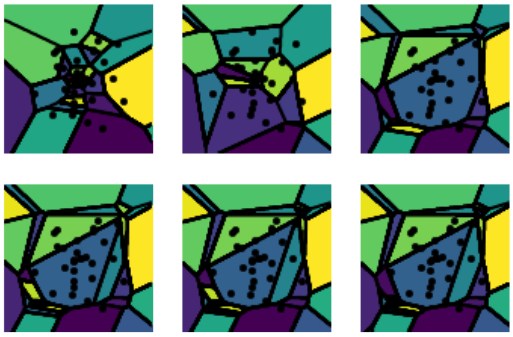

Figure 18: Laguerre cells at different steps (with τ_2).

Instead of performing a classical gradient ascent, we can implement a stochastic version of this procedure. Indeed, the functional $\mathcal{E}$ can be written as

$$\mathcal{E}(\phi) = \sum_{i=1}^{n} a_i \phi_i + \mathbb{E}_{Y\sim\beta}\left[\phi^c(Y)\right],$$

One can thus perform a *stochastic gradient ascent* (SGD) by drawing at each iteration a unique sample $y_t \sim \beta$ and updating the dual variable ϕ as

$$\phi_i^{(t+1)} \leftarrow \phi_i^{(t)} + \tau(t)(a_i - \mathbb{1}_{\mathbb{L}_i(\phi)}(y_t)), \quad i = 1, \dots, n.$$

A typical schedule is given by $\tau(t) = \frac{\tau_0}{1+t/t_0}$, where t_0 serves as a warm-up phase.

Remark 3.2. *For a convex objective, one can prove that the convergence rate of the stochastic gradient descent is* $\mathcal{O}(1/\sqrt{t})$.

We perform the stochastic gradient ascent across 300 iterations with starting point $\phi^{(0)} \equiv 0$, for the learning rate schedules $\widehat{\tau}_1(t) = \frac{\tau_0}{(1+t/t_0)^{1/4}}$, $\widehat{\tau}_2(t) = \frac{\tau_0}{(1+t/t_0)}$ and $\widehat{\tau}_3(t) = \frac{\tau_0}{(1+t/t_0)^2}$ with $\tau_0 = 0.1$ and $t_0 = 10$ (fig. 19). The schedule $\widehat{\tau}_1$ is even more chaotic than in the case of the classical gradient ascent, which is explained by the conjonction of low decay and stochastic updates; the schedule $\widehat{\tau}_3$ displays a slighty better performance compared to τ_3, which might be explained by the added stochasticity (and warm-up phase); the schedule $\widehat{\tau}_2$ is once again the best trade-off, though slightly less performant than the gradient ascent with τ_2.

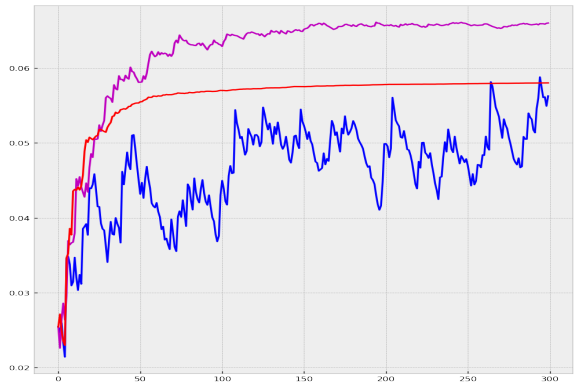

Figure 19: $\widehat{\tau}_1$ (blue), $\widehat{\tau}_2$ (fuchsia), and $\widehat{\tau}_3$ (red).

Remark 3.3. *Drawing from an idea issued from the field of reinforcement learning (*i.e. temporal difference TD(λ)*), we implement the scheme*

$$\widetilde{\phi}^{(t+1)} \leftarrow \phi^{(t)} + \tau(t)\nabla\mathcal{E}(\phi^{(t)}, y_t)$$
$$\phi^{(t+1)} \leftarrow (1-\sigma(t))\phi^{(t)} + \sigma(t)\widetilde{\phi}^{(t+1)},$$

where σ is another learning rate schedule. In the case where $\sigma(t) = 1/t$, this corresponds to the well-know average stochastic gradient descent *(ASGD). This allows to bootstrap on previous iterations of the ascent, which eventually makes the procedure more stable and quicker to converge. Because we are less likely to trust iterations from the beginning, we implement this algorithm with $\sigma(t) = 1/\sqrt{t}$ and $\tau(t) = \frac{\tau_0}{\sqrt{t/t_0+1}}$ with $\tau_0 = 0.1$ and $t_0 = 10$. We iterate across 300 iterations, and starting from a better initial point $\phi^{(0)}$ (fig. 20).*

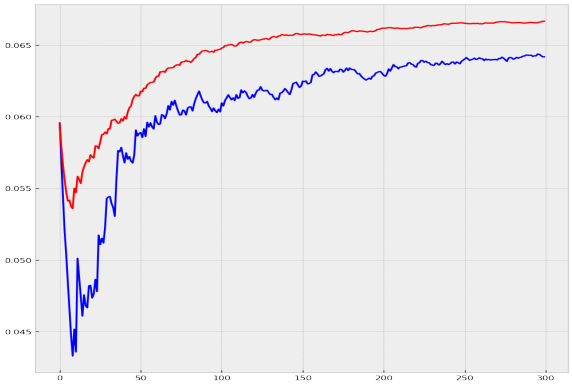

Figure 20: SGD with $\widehat{\tau}_2$ (blue) vs. ASGD with $\sigma(t) = 1/\sqrt{t}$ (red).

Another important application of semi-discrete optimal transport is the problem of *optimal quantization*, that is, given a continuous distribution β, finding a discrete measure $\alpha = \sum_{i=1}^n a_i x_i$ which minimizes $W_c(\alpha, \beta)$, i.e.

$$\min_{(a_i)_i,(x_i)_i} W_c\left(\sum_{i=1}^n a_i \phi_i, \beta\right).$$

This problem is convex in the a_i's, but sadly non-convex to the x_i's, which makes it hard to solve. Solving explicitly the minimization over the a_i's, we see that necessarily the dual potentials at optimality verify $\phi = 0$ (which means that the associated optimal Laguerre cells should be the Voronoi cells), and the problem therefore amounts to solving

$$\min_{x:=(x_i)_i} \mathcal{F}(x) \triangleq \sum_{i=1}^n \int_{\mathbb{L}_i(x)} c(x_i, y)\beta(\mathrm{d}y).$$

Suppose that β has a density w.r.t. to the Lebesgue measure, so that $\mathcal{F}$ is differentiable. By definition of the Voronoi cells, the gradients w.r.t. to the $\mathbb{L}_i(x)$'s vanish, so that for all $i = 1, \ldots, n$, if $c(x, y) = \|x - y\|_2$, one has

$$\nabla\mathcal{F}(x)_i = x_i\beta(\mathbb{L}_i(x)) - \int_{\mathbb{L}_i(x)} y\beta(\mathrm{d}y).$$

Therefore, any local minimizer should satisfy the fixed point equation

$$x_i = \frac{\int_{\mathbb{L}_i(x)} y\beta(\mathrm{d}y)}{\beta(\mathbb{L}_i(x))}, \quad i = 1, \ldots, n.$$

The *Lloyd's algorithm* consists in iteratively applying this fixed point equation. Though there are no theoretical guarantees of convergence, in practice it always converge to a local minimum. Note that this algorithm corresponds to the famous *k-means* algorithm, where at each step one replaces the centroids by the barycenter of the current cells.

We apply this algorithm to our datasets across 100 iterations. As expected, the $n = 30$ centroids accumulate toward the four Gaussian bumps of the distribution β(fig. 21). We can monitor $\mathcal{F}$ along the way to check that it is indeed steeply decreasing (fig. 22).

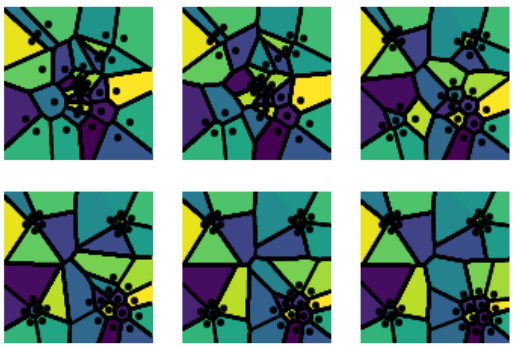

Figure 21: Laguerre cells at different steps.

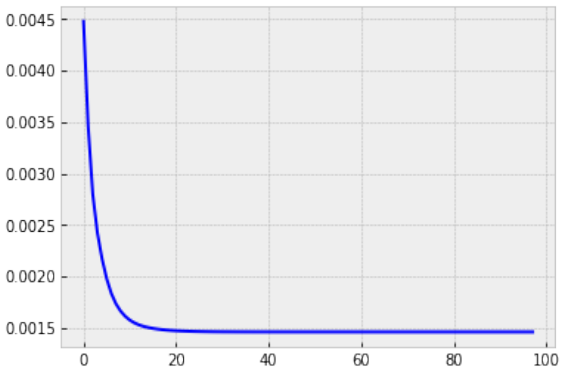

Figure 22: $\mathcal{F}$ across the iterations.

4. Advanced topics on Sinkhorn's algorithm

This numerical tour explore several extensions of the basic Sinkhorn's method.

4.1. *Theoretical background*

One can prove [3, Proposition 4.4] that strong duality hold for the primal regularized Kantorovitch problem (2.1),

and that the dual reads

$$\max_{\phi\in\mathbb{R}^n,\psi\in\mathbb{R}^m} \langle\phi,a\rangle + \langle\psi,b\rangle - \varepsilon\langle e^{\psi/\varepsilon}, K^\varepsilon e^{\phi/\varepsilon}\rangle, \quad (4.1)$$

where the dual variables are linked to the scaling vectors (u,v) through the relation $(u,v) = (e^{\phi/\varepsilon}, e^{\psi/\varepsilon})$. By denoting $Q(\phi,\psi)$ the objective in (4.1), we have that

$$\nabla_\phi Q(\phi,\psi) = a - e^{\phi/\varepsilon} \odot (K^\varepsilon e^{\psi/\varepsilon}) \quad (4.2)$$
$$\nabla_\psi Q(\phi,\psi) = b - e^{\psi/\varepsilon} \odot (K^{\varepsilon,\intercal} e^{\phi/\varepsilon}). \quad (4.3)$$

One strategy to solve (4.1) is therefore to iteratively vanish these gradients (*i.e.* which essentially amounts to a *block coordinate ascent with exact line searching*), which leads to the numerical scheme

$$\phi^{(t+1)} \leftarrow \varepsilon\log(a) - \varepsilon\log(K^\varepsilon e^{\psi^{(t)}/\varepsilon}) \quad (4.4)$$
$$\psi^{(t+1)} \leftarrow \varepsilon\log(b) - \varepsilon\log(K^{\varepsilon,\intercal} e^{\phi^{(t+1)}/\varepsilon}), \quad (4.5)$$

where we start from an initial condition $\psi^{(0)}$. This scheme is seen to be completely equivalent to Sinkhorn's algorithm by expressing the iterations back into the primal variables. The advantage of such a formulation is that it allows to naturally carry the computation of Sinkhorn's iterations into the log-domain.

For a vector $\xi\in\mathbb{R}^n$, if we define the ε-soft-minimum as $\min_\varepsilon(\xi) := -\varepsilon\log\left(\sum_{i=1}^n e^{-\xi_i/\varepsilon}\right)$, the iterations rewrite as

$$\phi_i^{(t+1)} \leftarrow \min_\varepsilon(C_{i,\cdot} - \psi^{(t)}) + \varepsilon\log(a_i)$$
$$\psi_j^{(t+1)} \leftarrow \min_\varepsilon(C_{\cdot,j} - \phi^{(t+1)}) + \varepsilon\log(b_j),$$

for all $i = 1,\dots,n$ and $j = 1,\dots,m$. To avoid underflow, the idea is to use here the celebrated *log-sum-exp trick*, which relies on the fact that for any vector $\xi\in\mathbb{R}$, if we write $\xi_\star = \min_i \xi_i$, we have $\min_\varepsilon(\xi) = \xi_\star + \min_\varepsilon(\xi - \xi_\star)$. This allows the quantities inside the soft-minimum to remain bounded, which in turn guarantees higher stability.

4.2. *Numerical experiments*

We experiment on the same datasets as in section 2, when dealing with the classical Sinkhorn's algorithm (fig. 7). Back then, we were unable to perform this algorithm underneath the regularization parameter $\varepsilon \approx 0.005$. We are now able to go far below this threshold.

We perform Sinkhorn's algorithm with log-domain across 1000 iterations for different regularization parameters ε and we display the errors $\|P^{(k)}\mathbb{1}_m - b\|_1$ on the mass conservation constraints (fig. 23). We observe that the smaller ε gets, the poorer the rate of decay of this error is, meaning that one must perform many more iterations to obtain satisfactory approximations. Compared to the classical Sinkhorn's algorithm, the computations performed in the log-domain are no longer simple vector and matrix multiplication, which greatly deteriorates the speed of the algorithm.

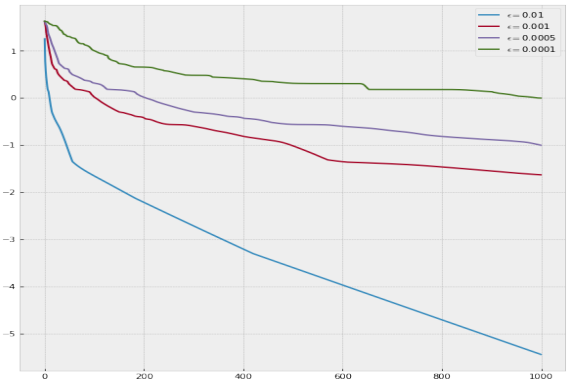

Figure 23: Errors on the constraint for $\varepsilon = 0.01$ (blue), 0.001 (red), 0.0005 (fuchsia) and 0.0001 (green).

Remark 4.1. *Even carrying out computations into the log-domain, when ε is extremely small, entries will ultimately overtake machine precision. In particular, if one wants to asymptotically solve the unregularized Kantorovitch problem* (1.1) *by gradually decreasing ε (which actually turns out to be impractical), one needs to deploy some fancier techniques, for instance by interpolating previous approximations with higher ε. (see [1, Sections 4.4.2, 4.4.4]).*

Let us now consider a very important application where entropic regularization serves a key ingredient. Recall that many problems in statistics boils down to fitting densities, *i.e.* given some observed data sampled from a distribution β, and some parametrized family of distributions $\{\alpha_\theta : \theta\in\Theta\}$ where $\Theta\subset\mathbb{R}^d$, one wants to find $\theta^\star$ such that $\alpha_{\theta^\star}$ best « fits » β, in some sense that has to be defined. While most widely-used strategies usually rely on *information-based approaches* (*e.g. Maximum Likelihood* (ML)), these methods may fail in some cases, for example when the distributions at stake are not absolutely continuous w.r.t. the Lebesgue measure (*e.g.* supported

on low-dimensional manifolds). To overcome this, another route to follow is to look at the geometries involved rather than the statistical information carried by the data.

An idea is to resort to optimal transport, *i.e.* solving

$$\theta^\star \in \arg\min_{\theta\in\Theta} W_c(\alpha_\theta, \beta) \tag{4.6}$$

for some cost c. The problem (4.6) raises many challenges, among which the computational cost needed to compute the Wasserstein losses and their general lack of smoothness, which makes gradient-based optimization methods instable.

We look at an entropic regularized version of (4.6). We define the *Sinkhorn loss* as

$$\underbrace{\min_{\pi\in\Pi(\alpha,\beta)} \int c(x,y)\pi(\mathrm{d}x,\mathrm{d}y) - \varepsilon \mathrm{KL}(\pi\|\alpha\otimes\beta)}_{:=W_c^\varepsilon(\alpha,\beta)}, \tag{4.7}$$

and we look at the regularized problem

$$\theta_\varepsilon^\star \in \arg\min_{\theta\in\Theta} W_c^\varepsilon(\alpha_\theta, \beta). \tag{4.8}$$

The Sinkhorn loss has the advantage of being differentiable, more robust to quantify, and easily computed through Sinkhorn's algorithm.

In what follows, we consider the *Lagrangian discretization* of (4.8), that is we suppose that $\beta = \frac{1}{m}\sum_{j=1}^m \delta_{y_i}$ (*e.g.* discretization of a continuous distribution) where $(y_j)_{j=1,\dots,n}$ are fixed points in $\mathbb{R}^d$ and $\alpha_\theta := \alpha_x = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$, *i.e.* $\Theta = (\mathbb{R}^d)^n$. In this context, the Sinkhorn loss is smooth w.r.t. to $\theta = x$ [3, Proposition 9.2] and its gradient reads

$$\nabla_x W_c^\varepsilon(\alpha_x,\beta) = \left(\sum_{j=1}^m P_{i,j}^\varepsilon \nabla_{x_i} c(x_i,y_j)\right)_{i=1}^n \in (\mathbb{R}^d)^n,$$

where P^ε is the unique optimal coupling. When $c(x,y) = \|x-y\|_2^2$, one therefore has

$$\nabla_{xi} W_c^\varepsilon(\alpha_x,\beta) = 2\left(a_i x_i - \sum_{j=1}^m P_{i,j}^\varepsilon y_j\right), \quad i=1,\dots,n.$$

We can thus implement a gradient descent. We use the same datasets as previously, except that $n = m = 100$. Note that, in this context of balanced optimal transport, the solution to (4.6) trivially reads $x_i = y_i$ for all $i = 1,\dots,n$. We perform 70 iterations of the gradient descent with 300 inner iterations for the Sinkhorn's algorithm with log-domain, and set $\varepsilon = 0.001$. With this small ε, we expect to solution to (4.8) to be closed to the solution of (4.6). Indeed, the gradient flow shows (fig. 24) that the five round clusters are slowly merging into the annulus (with x_i getting close to $y_{i^\star}$ where $i^\star = \sigma^\star(i)$, where $\sigma^\star$ is the optimal permutation).

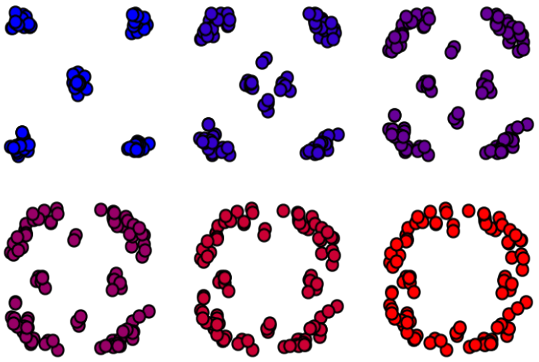

Figure 24: Gradient flow with $\varepsilon = 0.001$

What happens when ε increases ? Well, we know that $P^\varepsilon \to ab^\intercal$ as $\varepsilon\to\infty$. Therefore, at the limit $\varepsilon = \infty$, the gradients reads

$$\nabla_{xi} W_c^\infty(\alpha_x,\beta) = 2a_i\left(x_i - \sum_{j=1}^m b_j y_j\right), \quad i=1,\dots,n,$$

which means that the optimal solution is $\alpha_{x^\star}$ with $x_i^\star = \sum_{j=1}^m b_j y_j$ for all i, *i.e.* the barycenter of β. This follows our intuition: as $\varepsilon\to\infty$, we expect α_x to converge to the distribution with minimal entropy that is the « closest » to β, which we intuitively believe to be the barycenter of β. To numerically verify this fact, we run the gradient descent once more on our dataset, with 200 iterations and still 300 inner iterations for the Sinkhorn's algorithm with log-domain, and we set $\varepsilon = 10$. We observe (fig. 25) that the five round clusters merge roughly into the origin, as expected.

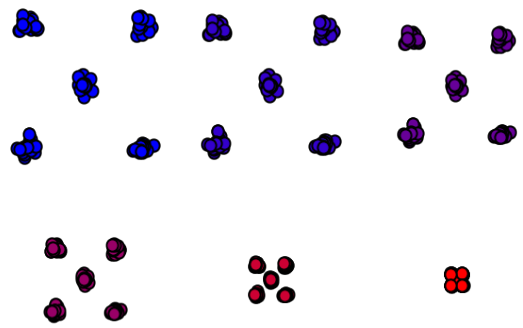

Figure 25: Gradient flow with $\varepsilon = 10$

Let us now define the *Sinkhorn divergence* as

$$\overline{W}_c^\varepsilon(\alpha,\beta) := W_c^\varepsilon(\alpha,\beta) - \frac{W_c^\varepsilon(\alpha,\alpha)}{2} - \frac{W_c^\varepsilon(\beta,\beta)}{2}.$$

One can prove that $\overline{W}_c^\varepsilon(\alpha,\beta) \to W_c(\alpha,\beta)$ as $\varepsilon \to 0$ and that $\overline{W}_c^\varepsilon(\alpha,\beta) \to \int -c\mathrm{d}(\alpha-\beta)\otimes \mathrm{d}(\alpha-\beta)$ [2]. Sinkhorn loss had two important disadvantages, *(i)* $\alpha \notin \arg\min_\beta W_c^\varepsilon(\alpha,\beta)$ and *(ii)* it behaves like an inner product in the limit $\varepsilon = +\infty$, and not like a norm. The Sinkhorn divergence corrects both of these issues. Note that

$$\nabla_{x_i} W_c^\varepsilon(\alpha_x,\alpha_x) = 2\sum_{j=1}^n (P_{i,j}^\varepsilon + P_{j,i}^\varepsilon)(x_i - x_j).$$

Remark 4.2. *For some reasons, when experimenting on the same dataset as before, we do not notice any difference with the Sinkhorn loss. For that reason, we do not include the plots in this document (there are available on the Github).*

In the case of *Generative Model Fitting*, we now consider that $x = G_\theta(z)$, where $z = (z_i)_{i=1,\dots,n}$ is a given point cloud and G_θ is a deformation parametrized by $\theta \in \Theta$. In this setting, the problem (4.8) typically corresponds to a problem of *image* (and *shape*) *registration*, as encounters for instance in medical imaging. Furthermore, we make the simplifying hypothesis that G_θ acts independently on each point z_i, *i.e.* $G_\theta(z) := (g_\theta(z_i))_i$. Therefore, $\alpha_\theta = \frac{1}{n}\sum_{i=1}^n \delta_{g_\theta(z_i)}$, and by applying the chain rule, the gradient w.r.t. θ of the objective of (4.8) reads

$$\sum_{i=1}^n [\nabla_\theta g_\theta(z_i)]^* \nabla_{x_i} W_c^\varepsilon(\alpha_\theta,\beta).$$

For instance, we take $g_\theta : \mathbb{R}^2 \to \mathbb{R}^2$ the non-linear transformation

$$g_\theta(x_1,x_2) = \begin{pmatrix} \theta_{00}^1 + \theta_{10}^1 x_0 + \theta_{01}^1 x_1 + \theta_{11}^1 x_0 x_1 \\ \theta_{00}^2 + \theta_{10}^2 x_0 + \theta_{01}^2 x_1 + \theta_{11}^2 x_0 x_1 \end{pmatrix},$$

We take z to be a sample of size $n = 100$ drawn from a uniform distribution on the annulus $\{x \in \mathbb{R}^2 : 1 \le \|x\|_2 \le 2\}$, and $y = G_{\theta^\star}(z)$ with $\theta^\star$ some randomly selected parameters (the weights a and b are uniform) (fig. 26).

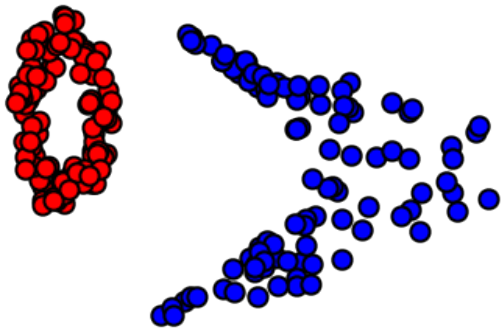

Figure 26: Datasets z (red) and $y = G_{\theta^\star}(z)$ (blue).

We perform the gradient descent with across 30 iterations with learning rate $\tau(t) = 1/(1+t)$ and starting point $\theta = 0$. The inner Sinkhorn's algorithms are performed with $\varepsilon = 0.01$ and 300 iterations fig. 27.

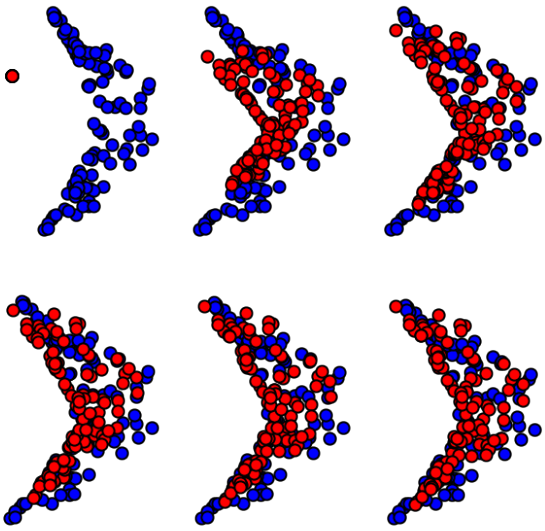

Figure 27: Flow of the gradient descent

References

[1] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. "Scaling algorithms for unbalanced optimal transport problems". In: *Mathematics of Computation* 87.314 (2018), pp. 2563–2609.

[2] Aude Genevay, Gabriel Peyré, and Marco Cuturi. "Learning generative models with sinkhorn divergences". In: *arXiv preprint arXiv:1706.00292* (2017).

[3] Gabriel Peyré, Marco Cuturi, et al. "Computational optimal transport". In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607.

[4] Cédric Villani. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media, 2008.