Numerical Tours: Linear Programming and Entropy-Regularized OT

Yvann Le Fay

January 2024

Let $C \in \mathbb{R}_+^{n \times m}$ be a cost matrix, let $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ be two histograms. The Kantorovitch problem in the discrete case consists in solving the following linear program in P:

$$\min_{P\mathbb{1}_n = a, P^\top \mathbb{1}_m = b} \sum_{i,j} P_{i,j} C_{i,j}. \tag{1}$$

To solve it, we use the Python package `CVXPY`. The dataset is taken from the Python package `POT` available at `https://github.com/PythonOT/POT`. It consists in the positions of bakeries and Cafés (resp. the $x_i's$ and the y_j's) and their production and consumption of croissants (resp. the a_i's and the b_j's after normalisation). The cost between the bakery i and the café j is given by the Euclidean distance between x_i and y_j, $C_{i,j} = \|x_i - y_j\|^2$. The `npz` file available at `https://github.com/PythonOT/POT/blob/master/data/manhattan.npz`. It can be shown that in the case where $n = m$ and $a_i = b_j = 1/n$, the solution of (1) is unique and is a matrix permutation. We illustrate this results by taking $n = m = 5$, keeping only the 5 bakeries with the most production. Our experiment shows the necessity to set $a_i = b_j = 1/n$ for the optimal P to be a permutation matrix. Under the sole hypothesis that $n = m$, the Kantorovitch-Monge equivalence states that there exists a optimal solution to (1) which is a permutation matrix. However, the solver of CVXPY does not lead to a permutation solution, see Figure 2.

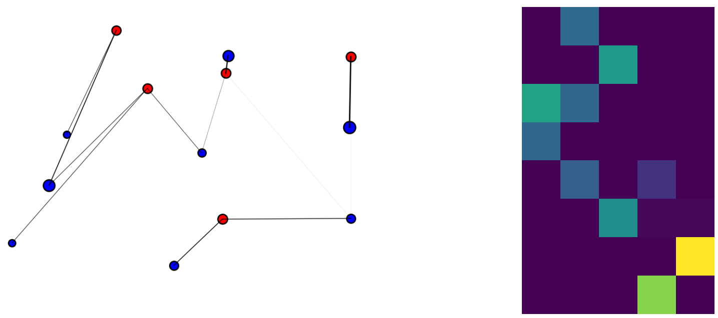

Figure 1: The Bakeries (in blue) and the Cafés (in red), the linewidth thickness from i to j is proportional to $P_{i,j}$. The corresponding transportation plan matrix P.

The entropy-regularization OT formulation makes the Kantorovitch objective strictly convex and guarantee the uniqueness of the solution. Let $C \in \mathbb{R}_+^{n\times m}$ be a cost matrix, let $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ be two histograms. The problem consists in solving the following problem in P:

$$\min_{P\mathbb{1}_n = a, P^\top \mathbb{1}_m = b} \sum_{i,j} P_{i,j} C_{i,j} + \varepsilon \sum_{i,j} P_{i,j}(\log(P_{i,j}) - 1), \tag{2}$$

this is done using the so-called Sinkhorn Algorithm.

We consider two sets of points, the x_i's are drawn uniformly inside a square of side 1 while the y_i's are drawn to lie (non-uniformly though) in an anulus around the circle defined by the l_p-norm for $p = 0.5$, according to 3, see Figure 3. The cost function is defined using the Euclidean norm, the regularization parameter ε is set to 0.001.

$$y_0 \sim \mathcal{U}([-1,1]), \quad r \sim 0.8 + 0.2\mathcal{U}([0,1]), \quad y_1 \sim \epsilon(r^p - |y_0|^p)^{1/p}, \tag{3}$$

where ϵ follows the Rademacher's law, $\epsilon = 1$ with probability 1/2. The two histograms are chosen to be uniform. As the regularization parameter ε goes to 0, the computed optimal transport plan becomes more and more sparse, see Figure 4. We implement a log-sum-exp version of the Sinkhorn algorithm by replacing the update steps on the potentials by update steps on the log-potentials and use the log-sum-exp trick to avoid arithmetic underflow. This gives rise to a numerically more stable algorithm. For the second experiment, we apply the Sinkhorn algorithm to the two following distributions $a \sim \mathcal{N}(0.8, 0.06)$ and $b \sim \text{Laplace}(0.25, 0.12)$. The logarithm of the optimal transport plans for different regularization parameter ε is plotted in Figure 5. Again, we see greather ε is, the more smooth (or less sparse) the transport plan is. The last experience consists in computing the Wasserstein barycenter a between four histograms (here images) a_1, a_2, a_3 and a_4, which is defined as the solution of

$$\min \lambda_1 W_2^2(a_1, a) + \lambda_2 W_2^2(a_2, a) + \lambda_3 W_2^2(a_3, a) + \lambda_4 W_2^2(a_4, a), \tag{4}$$

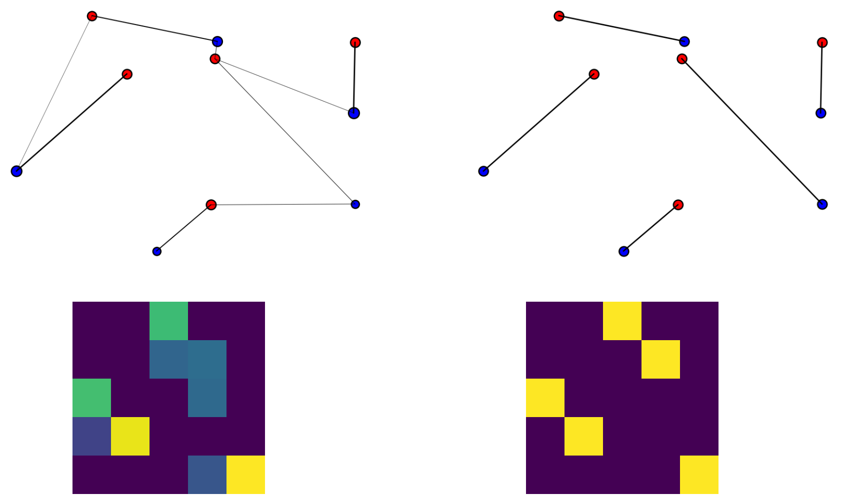

Figure 2: Same as previous but with 5 bakeries. Without and with uniform histograms.

with $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$. In our case, we set the λ's to $\lambda_1 = (1 - x_i)(1 - y_j)$, $\lambda_2 = (1 - x_i)y_j$, $\lambda_3 = x_i(1 - y_j)$ and $\lambda_4 = x_i y_j$ for $x_i = i/4$ and $y_j = j/4$ for $i, j \in [0, 4]$. See Figure 6.

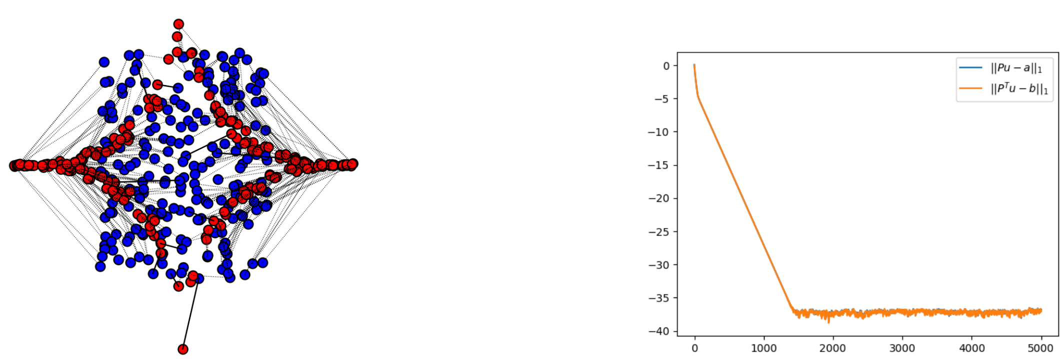

Figure 3: Left: Points lying inside a square are distributed to points inside the anulus. Right: Constraint violations.

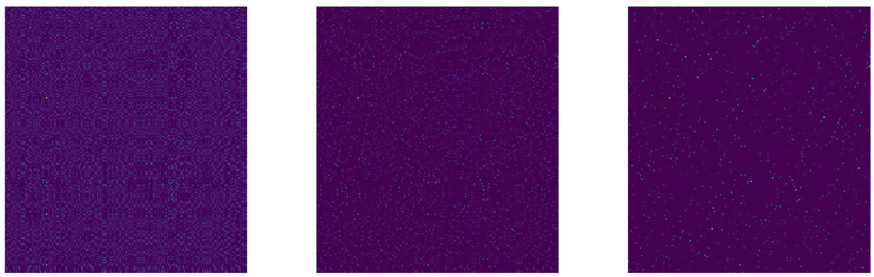

Figure 4: Impact of ε on the transport plan. From left to right, $\varepsilon = 10^{-n}$ for $n = 0, \dots, 2$.

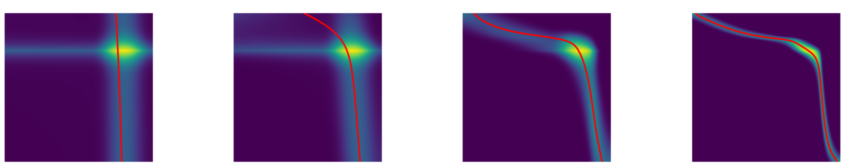

Figure 5: Log. of the transport plan. Impact of ε on the transport plan. From left to right, $\varepsilon = 10^{-n}$ for $n = 0, \dots, 3$.

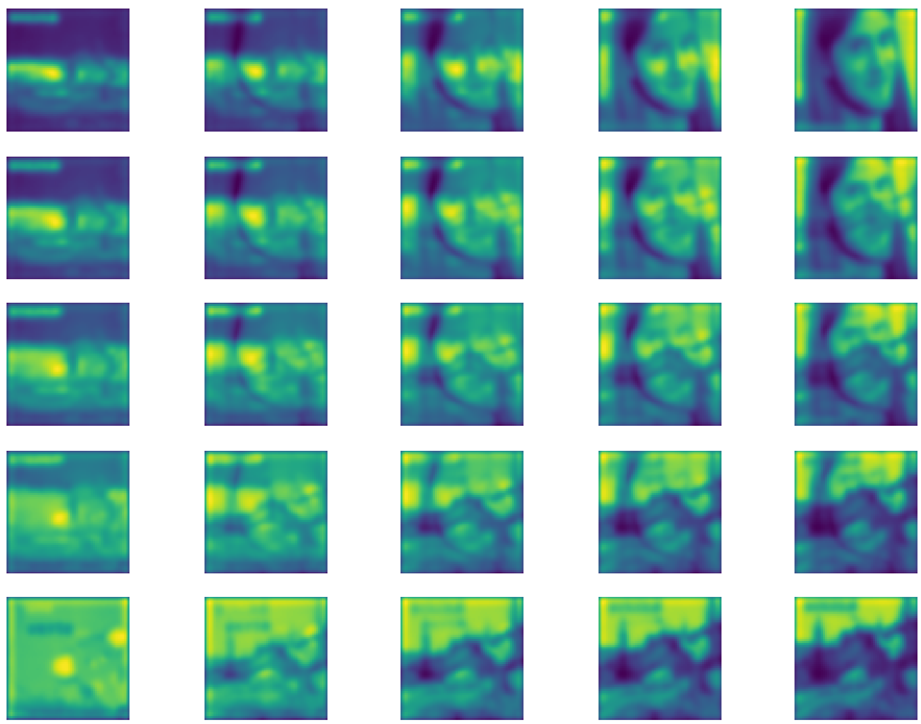

Figure 6: Bilinear interpolation using Wasserstein barycenters between 4 different images.