Optimal Transport Project: Approximating the Earth Mover's Distance

Ryan Boustany
ENSAE Paris
ryan.boustany@ensae.fr

Emma Sarfati
ENSAE Paris
emma.sarfati@ensae.fr

Abstract

In this paper, we propose a detailed study, interpretation and implementation explanation of the article of Wuchen Li, Stanley Osher and Ernest K.Ryu and Wotao Yin that was published in the *Journal of Scientific Computing* in 2018 [4] : A parallel method for Earth Mover's Distance. This paper is based on Optimal Transport and L_1 type regularization. The idea is to propose a smart implementation of optimization algorithms based on a reformulation of the Earth Mover's Distance that is efficient for computations. We also used some information on the preprint version of the article [3]. Our implementation is available at `https://github.com/esarf/emd_approximation`.

Introduction

Optimal Transport has become a trendy domain in the field of statistical research. It has enjoyed a growing litterature since a few years ([2],[6]), and seen two recent Fields medal prizes attached to its interest (Villani 2008, Figalli 2018). First, the domain itself presents a very interesting theoretical framework as it is at the outcome of statistics, algebra, computations and even physics. Second, the possible practical applications of this theory are multiple, and the discipline sometimes allows to achieve remarkable results in popular fields as image processing or machine learning ([7]). Third, the motivations of Optimal Transport are very intuitive and are based on historical problems of soldiers movements or stock flux. The paper aims at resolving one of those problems by carrying out a discretization of the transport plan, and using a "classical" optimization procedure thanks to a dynamical reformulation of the Optimal Transport Kantorovich problem, also called the Wasserstein Distance.
In what follows, we propose a compact summary of the authors' motivations and the published algorithm. We finally study our simulations and provide details about our implementation.

1 Review on optimal transport problem

The objective of Optimal Transport is to find the optimal movement of a set of elements, towards another target set of elements. The framework, meaning the plan on which we make this transfer can be varied: an image can be considered a plan on which the pixels correspond to the elements, a context in which we want to transport an amount of merch towards some shops is another example. We first recall the definition of the Optimal Transport Problem with the Earth Mover's Distance.

1.1 Earth Mover's Distance

Let μ and ν in $\mathcal{P}(\Omega)$, Ω is a measurable space and $\mathcal{P}(\Omega)$ the space of probability measures on the geometric space Ω, that is $\mathcal{P}(\Omega) = \{\rho(x) \in L^1(\Omega) : \int_\Omega \rho(x)\mathrm{d}x = 1, \rho(x) \geq 0\}$. The Monge

Problem was the first optimal transport to be formulated, in the XVIIIth century and it aimed at first to find an economic way to move soil between to places to make embankments.

$$\underline{\textbf{Monge Problem}} : \inf_{T\#\mu=\nu} \int_{\Omega} c(x, T(x))\mu(\mathrm{d}x)$$

where $T\#\mu$ corresponds to the image measure : $T\#\mu := \mu(T^{-1}(B))\ \forall B \in \Omega$, $c : \Omega \times \Omega \to \mathbb{R}$ is the cost function. Let now $\Pi(\mu, \nu)$ be the space of couplings:

$$\Pi(\mu, \nu) = \{P \in \mathcal{P}(\Omega, \Omega) : \forall A, B \subset \Omega, P(A \times \Omega) = \mu(A) \text{ and } P(\Omega \times B) = \nu(B)\}$$

Then, the Kantorovich problem is defined below.

$$\underline{\textbf{Kantorovich Problem}} : \inf_{P\in\Pi(\mu,\nu)} \int \int_{\Omega\times\Omega} c(x, y)P(x, y)\mathrm{d}x\mathrm{d}y$$

The Kantorovich problem is a computation-friendly reformulation of the Monge Problem. From this, we can define the Earth Mover's Distance, which exactly corresponds to the Kantorovich problem. For two probability measures ρ^0 and ρ^1 in $\mathcal{P}(\Omega)$, we write:

$$\mathrm{EMD}\left(\rho^0, \rho^1\right) := \min_{\pi\in\Pi(\rho^0,\rho^1)} \int_{\Omega} \int_{\Omega} c(x, y)\pi(x, y)\mathrm{d}x\mathrm{d}y \tag{1}$$

1.2 Dynamical approach

[4] provides a dynamical reformulation of the Earth Mover's Distance, where the proof is established in section 2. in the authors' paper. We won't go into further details of this reformulation here, but it all starts with a variational form of the EMD. Writing $\mathbf{x} = (x, y) \in \Omega \subset \mathbb{R}^2$, this reformulation leads to:

$$\mathrm{EMD}\left(\rho^0, \rho^1\right) = \left(\begin{array}{ll} \text{minimize} & \int_\Omega L(\mathbf{m}(\mathbf{x}))d\mathbf{x} \\ \text{subject to} & \nabla \cdot \mathbf{m}(\mathbf{x}) + \rho^1(\mathbf{x}) - \rho^0(\mathbf{x}) = 0 \\ & \mathbf{m}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = 0, \text{ for all } \left\{ \begin{array}{l} \mathbf{x} \in \partial\Omega \\ \mathbf{n}(\mathbf{x}) \text{ normal to } \partial\Omega \end{array} \right. \end{array} \right) \tag{2}$$

where the optimization variable $\mathbf{m} : \Omega \to \mathbb{R}^d$ is a flux vector satisfying the zero flux boundary condition (meaning it is padded with 0 at the boundary right and bottom), and L corresponds to the ground metric : $L(\mathbf{v}) = \|\mathbf{v}\|_2$ or $L(\mathbf{v}) = \|\mathbf{v}\|_1$. ∇ corresponds to the gradient operator.
An explanation of this reformulation is pretty intuitive. The flux vector $\mathbf{m}$ takes as input an element of Ω, $i.e$ a point $\mathbf{x} = (x, y)$, and outputs its *flow* along the coordinate x, and along the coordinate y (hence, $\mathbf{m}(\mathbf{x}) \in \mathbb{R}^2$). A flow means the mass that is transported when we move according to one coordinate (here, x or y). Concretely, the updated values inside the matrices of $\mathbf{m}$ are *direction* values, and they are optimized such as they point towards the potential flux, $\rho^0 - \rho^1$. In image example, that we will carry for our simulations study, this potential $\rho^0 - \rho^1$ just corresponds to the part of the grid where both images intersect themselves. This intersection is actually the path that allows to go from ρ^0 to ρ^1. It is then natural to enforce the optimal direction to go in this way. The minimization part just means that we want to force the values to cross this path with *minimum cost*.

1.3 Discretization

In [4], the space Ω is approximated with its discrete version. Particularly, $\Omega \subset \mathbb{R}^2$. We work in the following transport plan:

$$G = \{(x_i, y_j)\ \ i, j \in \{1...n\}\} \approx \Omega \subset \mathbb{R}^2$$

Note that G can be simply seen as a $n \times n$ squared grid. We then approximate $\Omega \subset \mathbb{R}^2$ by $\{x_1, ..., x_n\} \times \{y_1, ..., y_n\}$, where $(x_i, y_i)_{i=1...n}$ are the **support points**. The idea is that G can be seen as a squared grid, or a lattice graph equivalently.

The authors consider the discretizations of ρ^0 and ρ^1, with $\rho^0, \rho^1 \in \mathbb{R}^{n\times n}$, defined for $(i,j) \in \mathbb{R}^{n\times n}$ as:

$$\rho^0_{ij} \approx \int_{C(x_i,y_j)} \rho^0(x,y)dxdy$$
$$\rho^1_{ij} \approx \int_{C(x_i,y_j)} \rho^1(x,y)dxdy$$

where ρ^0 and ρ^1 correspond respectively to the input probability measures of the EMD, with $C(x,y)$ be the $\Delta x \times \Delta x$ cube (or square, as we are in $\mathbb{R}^2$ in this case) centered at (x,y), i.e.

$$C(x,y) = \left\{(x',y') \in \mathbb{R}^2 : |x'-x| \leq \Delta x/2, |y'-y| \leq \Delta x/2\right\}$$

The intuition behind this discretization is that ρ^k_{ij} ($k = \{0,1\}$) corresponds to the density that is around the reference point (x_i, y_j), where the term *around* means inside the cube centered on i,j, of radius $\Delta x/2$ (or length Δ). For instance, in a normalized image of size $n \times n$, where normalized means that we divide each pixel by n^2, the value of a pixel at place i,j represents ρ^k_{ij}. This value is representative of the "mass" around this place i,j (thanks to the normalization). Note that **the authors did not provide any illustration of their notations**, so it is difficult to understand the problem description. Below, we propose an illustration of the transport problem on a 3×3 lattice. This illustration corresponds to our own interpretation, and we do not fully certify its exactitude w.r.t the authors' interpretation.

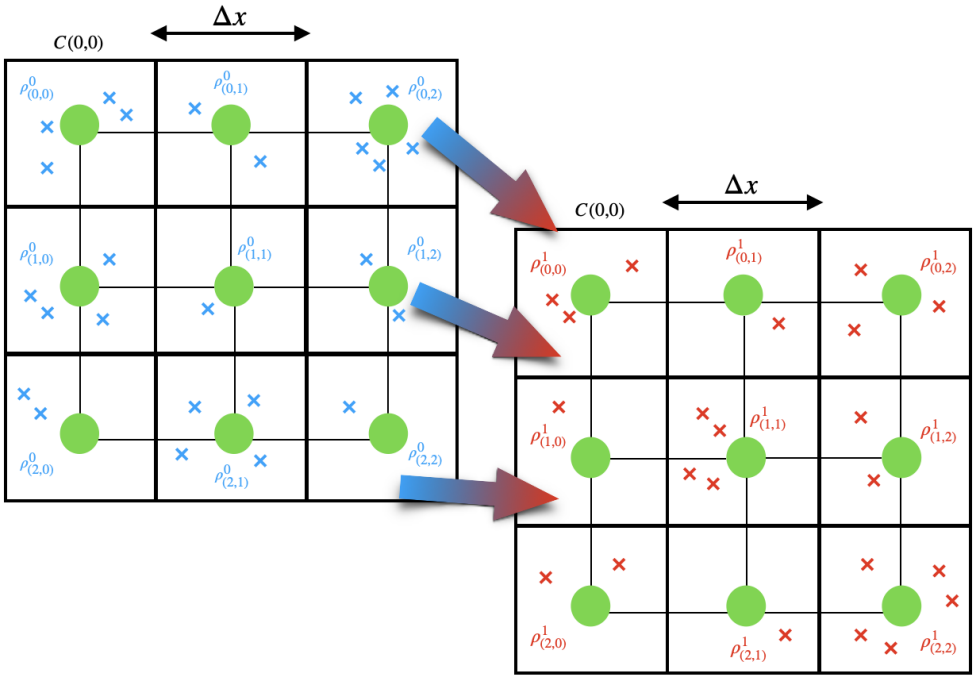

Figure 1: The discretized Optimal Transport, on a 3×3 grid. The objective is to transport the blue mass towards the red one (on the same grid), with minimum cost. Large green dots are the support points.

The flux vector $\mathbf{m}$ must also be discretized. A natural discrete representation of the flux going on the graphe for both coordinates is a 3D tensor, where the first matrix of the tensor is of shape $(n-1) \times n$ and contains the flux around $C(x_i + \Delta x, y_j)$, for $i \in \{1...n-1\}$ and $j \in \{1...n\}$ (*i.e* the flux along each x coordinate), and the second matrix of the tensor is of shape $n \times (n-1)$ and contains the flux around $C(x_i, y_j + \Delta x)$, for $i \in \{1...n\}$ and $j \in \{1...n-1\}$ (*i.e* the flux along each y coordinate). Hence, with padding the last row of the first matrix and the last column of the second matrix with 0, the flux vector $\mathbf{m}$ can be finally written as

$$\mathbf{m} = (m_x, m_y) \in \mathbb{R}^{n\times n\times 2}$$

Our interpretation is explained in the following figure. Once again, this interpretation is very personal.

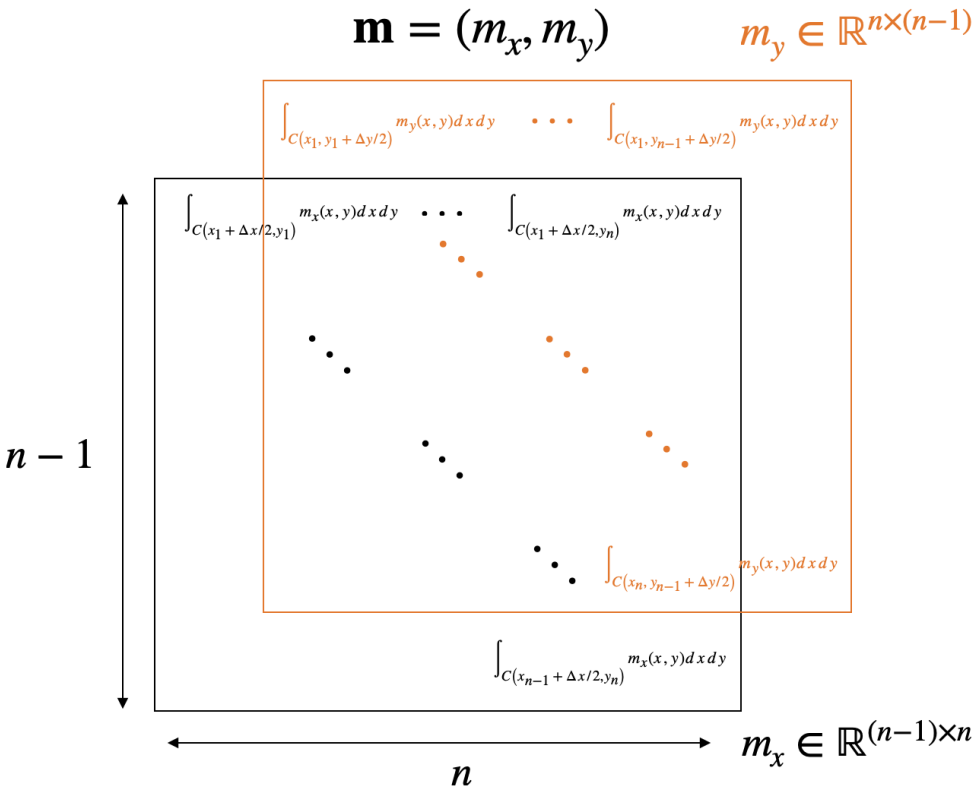

Figure 2: The discretized flux vector. The matrices contain an approximation of the continuous flux along each coordinate of the grid.

2 The optimization setting

The objective is to solve 2. At this aim, the authors propose to consider this problem but in the discrete case, with hope that it will approximate the *true* continuous solution. It turns out that the discretization leads to the ability of using a classical primal-dual method for solving the problem. First, the idea is to discretized all the continuous operators in 2. Recalling the paper definitions, the discrete gradient operator is defined for $\Phi \in \mathbb{R}^{n\times n}$ at point i, j as

$$\begin{aligned}(\nabla\Phi)_{x,ij} &= (1/\Delta x)\,(\Phi_{i+1,j} - \Phi_{i,j}) \quad \text{for } i = 1,\dots,n-1, j = 1,\dots,n\\ (\nabla\Phi)_{y,ij} &= (1/\Delta x)\,(\Phi_{i,j+1} - \Phi_{i,j}) \quad \text{for } i = 1,\dots,n, j = 1,\dots,n-1.\end{aligned}$$

A more interesting approach of the gradient in the case of zero-flux at the boundary is rather the divergence operator, which is given by $\mathrm{div}(\mathbf{m}) \in \mathbb{R}^{n\times n}$

$$\mathrm{div}(\mathbf{m})_{ij} = \frac{1}{\Delta x}\left(m_{x,ij} - m_{x,(i-1)j} + m_{y,ij} - m_{y,i(j-1)}\right)$$

An intepretation of this discrete divergence operator is that it measures the "variation" between m_x at point i, j and m_x at point $i-1, j$, the same with m_y. Moreover, the objective function $\int_\Omega L(\mathbf{m}(\mathbf{x}))$ considered on the discrete version of Ω, namely $G = \{\{x_1, ..., x_n\} \times \{y_1, ..., y_n\}\}$, is no longer an integral but a (double) sum, that is

$$\int_G L(\mathbf{m}(\mathbf{x})) = \sum_{i=1}^{n}\sum_{j=1}^{n} \|\mathbf{m}_{ij}\|_2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \sqrt{m_{x,ij}^2 + m_{y,ij}^2} = \|\mathbf{m}\|_{1,2}$$

where we have set $L := \|\cdot\|_2$ for example. These reformulations of both the gradient and the linear constraints allow to rewrite the problem 2 in the discrete case as:

$$\begin{aligned}\min_{\mathbf{m}} \quad & \textstyle\sum_{i=1}^{n}\sum_{j=1}^{n} L(\mathbf{m}_{ij})\\ \text{subject to} \quad & \mathrm{div}(\mathbf{m}) + \rho^1 - \rho^0 = 0\end{aligned} \tag{3}$$

or in the specific case of $L := \|\cdot\|_2$

$$\begin{array}{ll} \underset{\mathbf{m}}{\text{minimize}} & \|\mathbf{m}\|_{1,2} \\ \text{subject to} & \operatorname{div}(\mathbf{m}) + \rho^1 - \rho^0 = 0 \end{array} \tag{4}$$

The authors solve this optimization problem using a standard primal-dual approach. The objective is to solve the minimax problem over the Lagrangian

$$\min_{\mathbf{m}} \max_{\Phi} \quad \mathcal{L}(\mathbf{m}, \Phi)$$

An analytical resolution of the problem leads to closed-form solutions for $\mathbf{m}$ and Φ. In this short report, we do not aim at redo all the computations. Proofs sketches are however available in the notebook. Note that the following optimization conditions are inspired by the algorithm proposed by Chambolle and Pock [1]. The primal-dual first order conditions of the problem are

$$\begin{aligned} \mathbf{m}^{k+1} &= \underset{\mathbf{m}}{\operatorname{argmin}} \left\{ \|\mathbf{m}\|_{1,2} + \left\langle \Phi^k, \operatorname{div}(\mathbf{m}) \right\rangle + \frac{1}{2\mu} \left\| \mathbf{m} - \mathbf{m}^k \right\|_2^2 \right\} \\ \Phi^{k+1} &= \underset{\Phi}{\operatorname{argmax}} \left\{ \left\langle \Phi, \operatorname{div}\left(2\mathbf{m}^{k+1} - \mathbf{m}^k\right) + \rho^1 - \rho^0 \right\rangle - \frac{1}{2\tau} \left\| \Phi - \Phi^k \right\|_2^2 \right\} \end{aligned}$$

where μ, τ are the "Chambolle-Pock Lagrangian" parameters that will need to be tuned in the simulations. With analytical computations, this leads to the following updates for $\mathbf{m}$ and $\mathbf{\Phi}$

$$\begin{aligned} \tilde{\mathbf{m}}_{ij}^{k+1} &= \operatorname{shrink}_2\left(\tilde{\mathbf{m}}_{ij}^k + \mu \left(\nabla \Phi^k\right)_{ij}, \mu\right) \text{ for } i,j = 1,\dots,n \\ \Phi_{ij}^{k+1} &= \Phi_{ij}^k + \tau \left(\left(\operatorname{div}\left(2\mathbf{m}^{k+1} - \mathbf{m}^k\right)\right)_{ij} + \rho_{ij}^1 - \rho_{ij}^0 \right) \text{ for } i,j = 1,\dots,n \end{aligned} \tag{5}$$

where $\tilde{\mathbf{m}}$ stands for the *padded* version of $\mathbf{m}$, meaning 0 in the last row of m_x and 0 in the last column of m_y, as we evoked in part 1.3. These updates are the only theoretical background needed to solve the optimization problem and so the EMD approximation. In the following section, we discuss our implementation and our results.

3 Results

In this part, we study the simulations of our implementation of algorithm 5. Note that the authors proposed a Matlab implementation of their algorithms that was available on GitHub. Our experiments were however lead on Python, so our work was helped by the authors' code but some incompatibility/unabled functionalities between Matlab and Python forced us **to have a full, global understanding of the article in order to code it properly**. We ran our experiments on a MacBook Pro with 8 Go of RAM and, 1.4 GHz Intel Core i5 4-cores processor. The data on which we tested the algorithms are the same as in the paper, *i.e* two cat pictures of grid $n = 256$. Note that in our simulations, we also tested the algorithms for grids $n = 128, 512, 1024$, but we decided to keep this version to be consistent with the paper illustration.

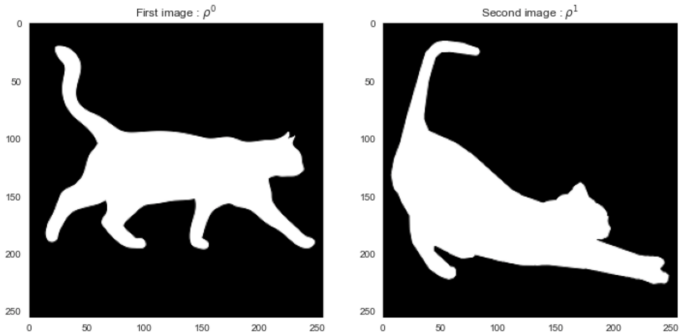

Figure 3: The two input matrices of the EMD algorithms for $n = 256$. The goal is to move optimally from the standing cat to the crouching one.

3.1 Visual results

We tested our algorithms for L_1 and L_2 norm, with the same set of parameters. For the two images below, we fixed $\mu = 10^{-7}$, $\tau = 2$ and a maximal number of 15,000 iterations.

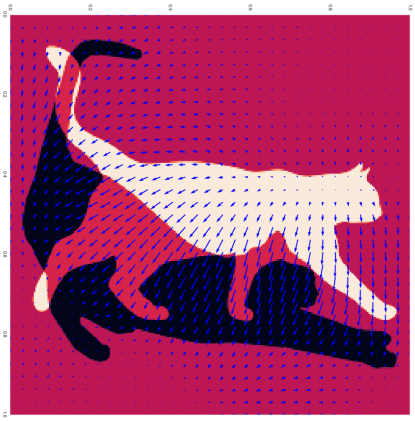

Figure 4: L_1 solution after 15,000 iterations

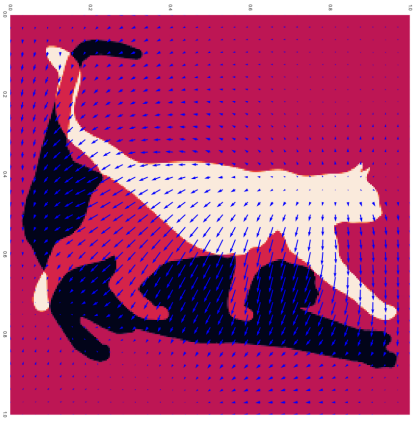

Figure 5: L_2 solution after 15,000 iterations

We see that the results are almost perfectly the same. This is due to the fact that below a (high) number of iterations, both methods hardly differ in terms of visual results. With 100,000 iterations, the results provided by the authors in the paper show distinct results for EMD L_1 and L_2. As we exclusively worked on a CPU, we did not test the algorithms above 50,000 iterations, which was already huge for our machine. In the above example, 15,000 iterations were used, and we see that it already gives us powerful results. The number of iterations is crucial: below, an example with just 5,000 iterations less than the first figure.

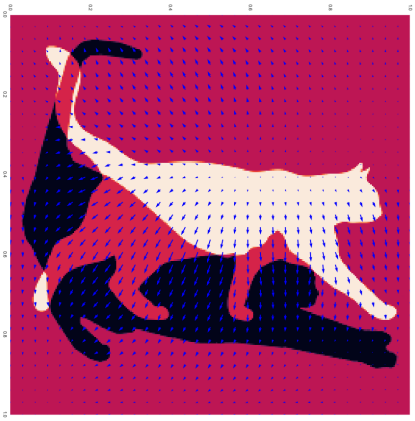

Figure 6: L_1 solution after 10,000 iterations

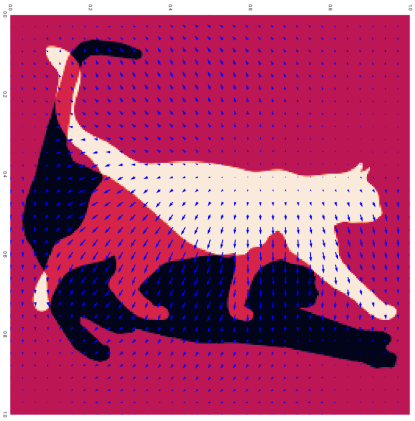

Figure 7: L_2 solution after 10,000 iterations

We see that the arrows at the top-left part of the images are not pointing towards the optimal flow, as it is going in the inverse direction of the crouching cat. The arrows are generally way less coherent than before. However with 15,000 iterations the default is corrected; this proves the importance of the number of iterations.

3.2 Model evaluation

In this part, we focus on the numerical evaluation of the authors' model. Let us focus on EMD L_1 algorithm. The ideas are the same for L_2 algorithm. A way to evaluate an optimization process with primal-dual updates is to compute the residual of the model, at each timestep (meaning, at each iteration k). Note that computing the value of the primal function at each iteration is not really relevant as the value of $\|\mathbf{m}\|_{1,1}$ won't be automatically decreasing due the constraint (4 is a minimax problem, and the max part of the optimization also changes at each iteration, hence the min is not necessarily decreasing). A chosen criterion of evaluation by the authors is the residual of the model, defined by

$$R^k = (1/\mu)\left\|\mathbf{m}^{k+1} - \mathbf{m}^k\right\|_2^2 + (1/\tau)\left\|\Phi^{k+1} - \Phi^k\right\|_2^2 - 2\left\langle \Phi^{k+1} - \Phi^k, \operatorname{div}\left(\mathbf{m}^{k+1} - \mathbf{m}^k\right)\right\rangle$$

and it corresponds to the difference between the functions inside the first-order primal-dual conditions. As we aim at minimizing a function of $\mathbf{m}$, and maximizing a function of Φ at each timestep, computing the difference between the updated numbers will tell us whether we achieved to minimize and maximize the variables $\mathbf{m}$ and Φ respectively, simultaneously, at each iteration (we put $+$ in front of the argmin function, and $-$ in front of the argmax). If the number becomes smaller, it means that the algorithms "works". We compute it for 6 different maximum numbers of iterations K.

K	Final R^K
5,000	6.7630
10,000	5.5343
15,000	4.7517
20,000	4.1777
25,000	3.7290
30,000	3.3636

Table 1: Final residuals for EMD L_1 according to maximum number of iterations K.

We the residuals decreasing as the number of maximal iteration increases, which is an expected result. In our experiments, we also plot some curves to monitor the evolution of the residuals during the algorithm (see our notebook).

3.3 Hyperparameter tuning

An important concern that is not mentioned by the authors and that we noticed during our experiments is the sensitivity of EMD L_1 and L_2 algorithms to some parameters of the model. Especially when it comes to μ, the results can be very different. We focus here on the L_1 to illustrate this phenomenon. Below we provide an example of the flux computed with $\mu = 10^{-8}$ instead of $\mu = 10^{-7}$. Other parameters are the same: 15,000 iterations, $\tau = 2$.

Figure 8: The optimal flow computed by EMD L_1 with $\mu = 10^{-8}$.

We can note that the arrows provide bad flux directions. On the top right of the image, they point towards the opposite direction of the crouching cat. Left part of the image is also not interesting as the arrows are not coherent.

Conclusion

We gave a summary of the paper's main concepts and ideas. We provided our personal illustrations and interpretation; in this sense, this report **should not** be considered infallible and this is necessary to take a look at the paper to have an own opinion. We tested the algorithms proposed by the authors on an Anaconda environment, using Python. This lead us to coherent visual results and numerical ones. More convincing optimal flow could be computed with more iterations, for both algorithms L_1 and L_2. However, we did not access any GPU so we had to cut-off the number of iterations to a maximal number of $50,000$. But the results were still visually and numerically interesting. Further refinements of this algorithm, proposed by some of the same authors, are available in [5]. We study this algorithm at the end of our notebook.

References

[1] Antonin Chambolle and Thomas Pock. "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging." In: *Journal of Mathematical Imaging and Vision* 40.1 (2011), pp. 120–145. URL: http://dblp.uni-trier.de/db/journals/jmiv/jmiv40.html#ChambolleP11.

[2] Marco Cuturi. "Sinkhorn Distances: Lightspeed Computation of Optimal Transport". In: *Advances in Neural Information Processing Systems*. Ed. by C. J. C. Burges et al. Vol. 26. Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf.

[3] Wuchen Li, Stanley Osher, and Wilfrid Gangbo. *A fast algorithm for Earth Mover's Distance based on optimal transport and L1 type Regularization*. 2016. arXiv: 1609.07092 [math.NA].

[4] Wuchen Li et al. "A Parallel Method for Earth Mover's Distance". In: *Journal of Scientific Computing* 75 (Apr. 2018). DOI: 10.1007/s10915-017-0529-1.

[5] Jialin Liu et al. *Multilevel Optimal Transport: a Fast Approximation of Wasserstein-1 distances*. 2019. arXiv: 1810.00118 [stat.CO].

[6] Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport*. 2020. arXiv: 1803.00567 [stat.ML].

[7] Mathieu Serrurier et al. *Achieving robustness in classification using optimal transport with hinge regularization*. 2020. arXiv: 2006.06520 [cs.LG].