Stochastic Wasserstein Barycenters (Computational Optimal Transport Project)

Inès VATI ines.vati@eleves.enpc.fr
MVA, ENS Paris Saclay

January 15, 2024

Abstract

Abstract (½ page): What problem(s) is studied? Why is it relevant? What solution(s) is proposed? Which contributions (theory, numerics, etc) ?

1 Introduction

Addressing the computation of barycenters for probability distributions stands as a fundamental problem in statistics and machine learning. It becomes compelling to summarize a collection of probability distributions with due consideration to the inherent geometric structure.

The theory of optimal transport (OT) provides a promising and theoretically-grounded avenue for averaging distributions over a geometric domain. The Wasserstein barycenter is a generalization of the notion of mean to probability measures. It is defined as the minimizer of the sum of the Wasserstein distances to the input measures. The Wasserstein distance itself serves as a metric on the space of probability measures that takes into account the underlying geometry of the space. It is defined as the minimum cost of transporting one measure onto the other.

The computation of barycenters finds diverse applications across various fields, including statistics, computer vision [1, 2], signal analysis [3, 4], and medical imaging [5]. For instance, Bruckstein et al. [1] propose a new approach to compute the average of discrete probability distributions as a barycenter over the Wasserstein space and apply their method to texture synthesis and texture mixing. Texture mixing problem consists in synthesizing a new texture from a collection of atoms, i.e. examplars. In [3], the authors leverage these tools in time series modeling. Each segment in time series data can represent a state, such as running or walking in a human activity application. They propose an innovative Dynamical Wasserstein Barycenter model to estimate the distribution of pure states while improving state estimation for transition periods.

Related works. Several works have addressed the computation of Wasserstein barycenter. Cuturi et al. [6] presented a concise and efficient Sinkhorn algorithm, subsequently extended to barycenter problems through, for instance, iterative Bregman projection algorithms [7]. These algorithms introduce entropic regularization to the initial linear problem, demonstrating that the set of linear constraints can be split in an intersection of simpler constraints, allowing for closed-form projections.

Staib et al. [8] introduced a stochastic barycenter algorithm from samples, presenting a scalable and parallelized approach suitable for streaming data—continuously generated data from diverse sources. Their method is also robust to nonstationary input distributions. However, their method requires a finite, predetermined set of support points.

Several works have aimed at enhancing algorithmic speed. Dvurechenskii et al. [5] proposed a distributed algorithm for computing a discrete approximation of the regularized Wasserstein barycenter for a set of continuous probability distributions stored across a network, i.e. each agent constituing the network holds a private continuous probability measure.

Grounded in an accelerated primal-dual stochastic gradient method for convex optimization with linear equality constraints, their approach seeks to expedite computation in a decentralized fashion.

Current state-of-the-art techniques typically define the barycenter on a fixed, refined grid and optimize the weights associated with each grid point. In this studied article [9], the optimization is extended to include the grid itself, and uniform weights are assigned to each grid point.

Problem statement. We consider of collection of J distributions $\{\mu_j\}_{j=1}^{J}$, either discrete or continuous, defined on a common domain $\mathcal{X} \subset \mathbb{R}^d$. The barycenter ν of these distributions is defined as the solution of the following optimization problem

$$\min_{\nu} \frac{1}{J} \sum_{j=1}^{J} W_2^2(\nu, \mu_j)$$

The squared 2-Wasserstein distance between two probability measures ν and μ is defined as

$$W_2^2(\mu, \nu) = \min_{\pi \in \mathbf{U}(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{X}} d(x,y)^2 d\pi(x,y) \tag{1}$$

where $\mathbf{U}(\mu, \nu)$ is the set of all coupling π between μ and ν

$$\mathbf{U}(\mu, \nu) = \{\pi \in \mathcal{M}_+^1(\mathcal{X} \times \mathcal{Y}), (P_{\mathcal{X}})_{\#}\pi = \mu \text{ and } (P_{\mathcal{Y}})_{\#}\pi = \nu\}$$

$(P_{\mathcal{X}})_{\#}$ and $(P_{\mathcal{Y}})_{\#}$ are the push-forward by the projectors $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ on the first and second marginals respectively.

They opt to a semidiscrete approximation of the barycenter. Let $\Sigma = \{x_i\}_{i=1}^{M}$ be a set of M points in $\mathcal{X}$. The barycenter ν is approximated by a discrete measure $\hat{\nu}$ with uniform weights on the points of $\Sigma \subset \mathcal{X}$, i.e.

$$\hat{\nu} = \frac{1}{M} \sum_{i=1}^{M} \delta_{x_i}$$

In this setting, the optimization problem reads

$$\min_{\forall\ i \leq M,\ x_i \in \mathcal{X}} \frac{1}{J} \sum_{j=1}^{J} W_2^2(\mu_j, \hat{\nu}) \tag{2}$$

The dual problem of each sub problem of (2) reads

$$\max_{\phi_j \in L^1(\mathcal{X})} F_{dual}\left(\phi_j, \{x_i\}_{i=1}^M\right) = \int_{\mathcal{X}} \phi_j(x) d\nu(x) + \int_{\mathcal{X}} \overline{\phi}_j(y) d\mu_j(y)$$
$$= \frac{1}{M} \sum_{i=1}^{M} \phi_j(x_i) + \int_{\mathcal{X}} \overline{\phi}_j(y) d\mu_j(y)$$

where $\phi_j \in \mathbb{R}^M$ is the discrete Kantorovith potential associated with the OT problem (1) and $\overline{\phi}_j$ is the c-transform of ϕ_j defined by $\overline{\phi}(y) = \inf_{x \in \mathcal{X}} d(x,y) - \phi(x)$ [10].

Therefore, the objective function optimize in this work [9] is

$$F\left(\{\phi_j\}_{j=1}^J, \{x_i\}_{i=1}^M\right) = \frac{1}{J} \sum_{j=1}^{J} F_{dual}\left(\phi_j, \{x_i\}_{i=1}^M\right) \tag{3}$$

Contributions. The main contribution of the research conducted in [9] lies in the introduction of a stochastic approach for computing Wasserstein barycenters, thereby eliminating the need for a fixed grid implementation. Notably, the support of the estimated barycenter is optimized and empirically demonstrated to be contained within the support of the true barycenter. In a departure from conventional approaches, their problem formulation eschews regularization. The authors also provide a theoretical analysis elucidating the algorithm's convergence dynamics.

In this report, we delve into the method proposed by [9], applying it to both simple 1D and 2D cases. Additionally, we conduct a comparative analysis, evaluating the performance of this approach against existing methods in terms of both quality and computational speed[1].

2 Presentation of the method

The proposed algorithm [9] boils down to two key steps :

1. **Stochastic Gradient Ascent for Potentials:** With $\{x_i\}$ fixed, a stochastic gradient ascent is employed to optimize the potentials $\{\phi_j^i\}$, leveraging the concavity of F (3) in ϕ_j.

2. **Fixed-Point Iteration for Positions:** With $\{\phi_j^i\}$ fixed, a single fixed point iteration is performed to update the barycenter grid $\{x_i\}$. This

[1] Code is available at https://github.com/InesVATI/projectOT_stochastic_wasserstein_barycenters

update is facilitated by a closed-form expression for the zeroed values of the gradient, i.e.,

$$\frac{\partial F}{\partial x_i} = 0$$

2.1 Stochastic Gradient Ascent for Potentials

For step 1, we note that F is concave in the potentials ϕ_j. Indeed, we have shown in the course [10] that the c-transform is concave for the euclidean cost $d(x,y)^2 = \|x-y\|_2^2$. To get the gradient, we need to compute the quantities

$$\begin{aligned} a_j^i &= \int_{V_{\phi_j}^i} d\mu_j(y) & \qquad b_j^i &= \int_{V_{\phi_j}^i} y\, d\mu_j(y) \\ &= \mathbf{E}_{y\sim\mu_j}\left[\mathbb{1}_{y\in V_{\phi_j}^i}\right] & &= \mathbf{E}_{y\sim\mu_j}\left[y.\mathbb{1}_{y\in V_{\phi_j^i}}\right] \end{aligned}$$

where $\mathbb{1}$ indicates the indicator function of a set. In the derivative of (3), the *power cell* $V_{\phi_j}^i$ of point x_i comes into play and is defined as follows link with voronoi cell !!!

$$V_\phi^i = \{x \in \mathcal{X}, d(x,x_i)^2 - \phi_i \leq d(x,x_{i'})^2 - \phi_{i'}, \forall i'\}$$

The computation of b_j^i is given in the algorithm 1, the computation of a_j^i is similar.

Algorithm 1 Estimate a_j^i and b_j^i by Monte Carlo approximation

```
Require: iid samples $\{Y\}_{k=1}^{K} \sim \mu_j$
for $y$ in $Y$ do
    if $y \in V_{\phi_j}^i$ then
        $b_j^i \leftarrow b_j^i + y$
    end if
end for
$b_j \leftarrow \frac{1}{K} b_j$
return $b_j^i$
```

The well known gradient ascent step is given by

$$\phi_j^{(l+1)} = \phi_j^{(l)} + \alpha \frac{\partial F}{\partial \phi_j}(\phi_j^{(l)})$$

at iteration l, where λ is the step size. To improve performance, the authors apply the Nesterov acceleration. The *ascent step* is given in algorithm 2.

Algorithm 2 Ascent step

```
for $j = 1, \dots, J$ do
    $z^{(0)} \leftarrow 0$
    while $\left\| \widehat{\frac{\partial F}{\partial \phi_j}} \right\| > \epsilon_{ascent}$ do
        Compute $\hat{a}_j^i$ thanks to algorithm 1
        $z^{l+1} \leftarrow \beta z^{(l)} + \widehat{\partial_{\phi_j} F}(\phi_j)$ {Nesterov acceleration}
        $\phi_j^{(l+1)} \leftarrow \phi_j^{(l)} + \alpha z^{(l+1)}$
    end while
end for
```

In algorithm 2, $\widehat{\partial_{\phi_j} F}$ denote the stochastic approximation of the gradient of F with respect to ϕ_j.

2.2 Fixed Point Iteration for Positions

The update of the point $\{x_i\}$ is given by

$$\frac{\partial F}{\partial x_i} = 0 \implies x_i = \frac{\sum_{j=1}^{J} b_j^i}{\sum_{j=1}^{J} a_j^i}$$

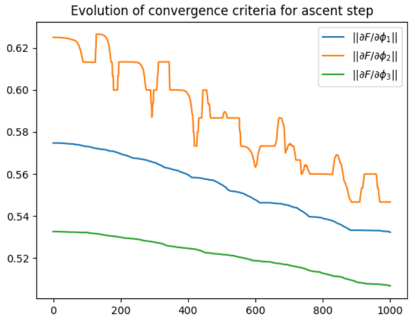
This step boils down to a simple fixed point iteration and is denoted as the *snap step*. The algorithm 3 summarize the whole procedure.

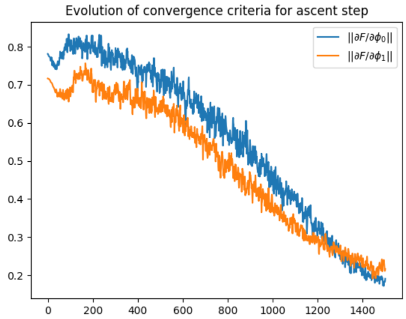
Algorithm 3 Ascent and Snap algorithm for computing Stochastic Wasserstein Barycenters

```
for $t = 1, \dots, T$ do
    Update $\{\phi_j\}$ with algorithm 2 {Ascent step}
    Compute $\hat{b}_j^i$ with algorithm 1
    for $x_i \in \Sigma$ do
        $x_i \leftarrow \frac{\sum_{j=1}^{J} \hat{b}_j^i}{\sum_{j=1}^{J} \hat{a}_j^i}$ {Snap step}
    end for
end for
return Opitimized barycenter support $\Sigma^* = \{x_i\}_{i=1}^{M}$
```

2.3 Theoretical guarantees

Well behavior of the algorithm is ensured for the following assumptions.





- At least one of the input distribution μ_j is absolutely continuous with respect to the Lebesgue measure.

- ...

3 Experiences and Numerical Analysis

3.1 Analysis of the ascent step convergence

During the iterative procedure, we plot the evolution of the converge criteria, namely $\left\|\frac{\partial F}{\partial \phi_j}\right\|$ for $j \in [\![1, J]\!]$.

The majority of computation time is consumed by the ascent step (1 hours).

The ascent step take most of the computation time. In [9], the authors use a step size of $\alpha = 10^{-3}$ $\epsilon_{ascent} = 10^{-6}$ in Algorithm 2. However, the while loop took an excessive amount of time to converge, exceeding 4 hours. To enhance computational efficiency, we opt for $\alpha = 0.05$ and $\epsilon_{ascent} = 10^{-4}$. Nesterov acceleration is set to $\beta = 0.99$ like in [9]. Additionally, we introduce a maximum number of iterations, limiting the while loop to $T_{max} = 1500$ for efficiency.

3.2 Quantitative Analysis and Comparative Evaluation

3.2.1 Optimal Monge Map Approach to Barycenter Computation

In the case $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ and $d(x, y)^2 = \|x - y\|^2$ and if at least one of the two input measures has a density with respect to the Lebesgue measure, the Theorem 2.1 of [10] states that the optimal coupling π of (1) is unique and is given by $\pi = (Id, T)_{\#}\mu$ where $T : \mathcal{X} \to \mathcal{Y}$ denote the "optimal Monge map" with $T_{\#}\mu = \nu$.

In the 1D case [see 10, Remark 2.30], $d = 1$, the optimal Monge map between two distributions μ_1 and μ_2 writes

$$T = \mathcal{C}_{\mu_2}^{-1} \; o \; \mathcal{C}_{\mu_1} \tag{4}$$

where $\mathcal{C}_{\mu} : \mathbb{R} \to [0,1]$ and $\mathcal{C}_{\mu}^{-1} : [0,1] \to \mathbb{R}$ are respectively the cumulative distribution function and its pseudoinverse, also called the generalized quantile function of μ .

Using the Remark 7.1 [10], the McCann's interpolation [11] between two measures reads, for $t \in [0,1]$

$$\begin{aligned} \mu_t &= (tT + (1-t)Id)_{\#}\mu_1 \\ \mu_t &= (t\mathcal{C}_{\mu_2}^{-1} \; o \; \mathcal{C}_{\mu_1} + (1-t)Id)_{\#}\mu_1 \end{aligned} \tag{5}$$

Figure 2 shows the displacement interpolation between 1-D measures, using the cumulative distribution function as detailed in (5).

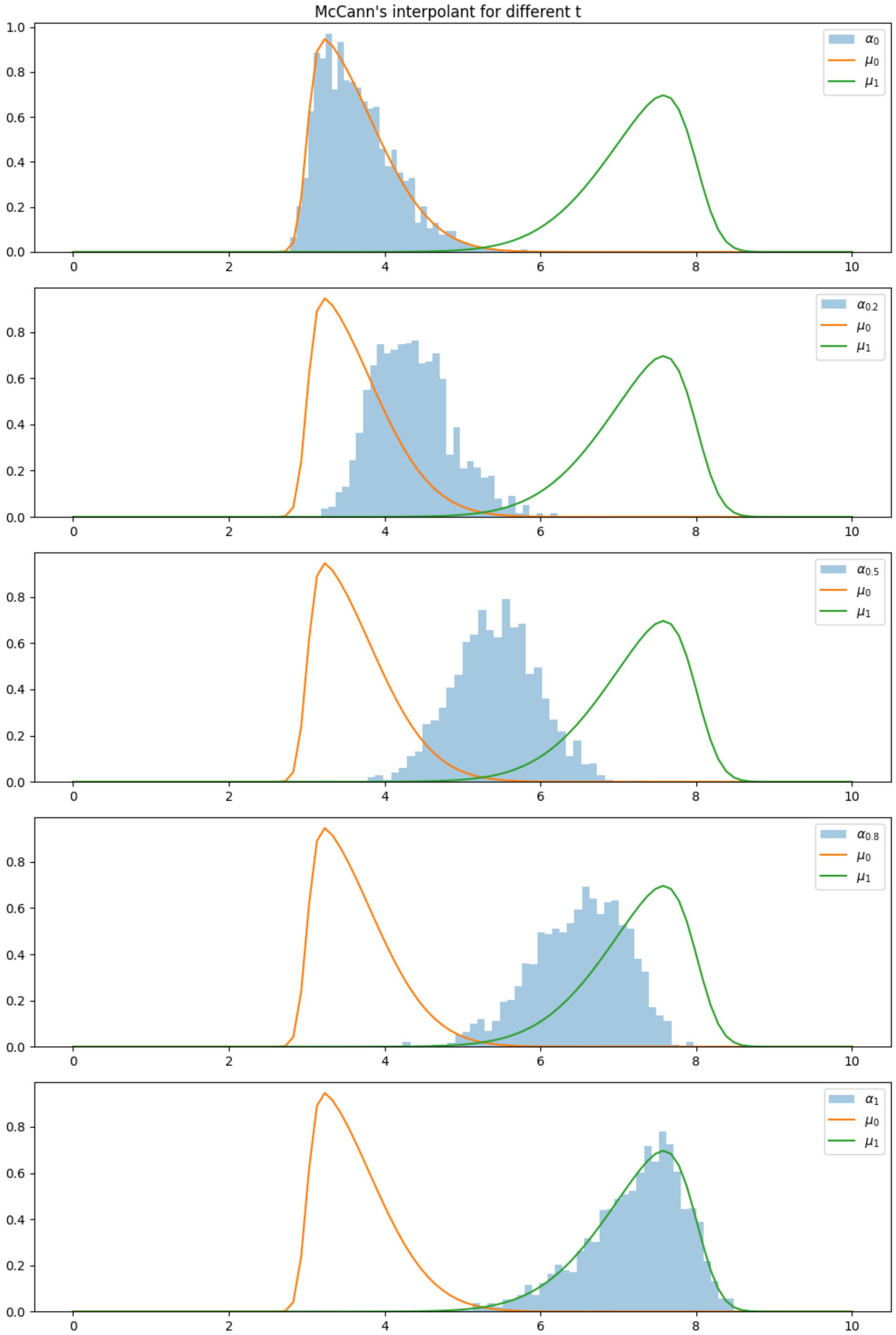


Figure 2: McCann's interpolant between two skew-normal distributions. From top to bottom, t=0, 0.2, 0.5, 0.8, 1.

To obtain the interpolant, we generate K samples (X_k) independent and identically distributed (iid) according to μ_1. Then, we compute the samples (Y_k) as follows

$$Y_k = tTX_k + (1-t)X_k, \qquad \forall k \in [\![1, K]\!]$$

where T is given in (4). We have that (Y_k) is iid according to μ_t.

To compare with the algorithm 3, we use $t = 0.5$. The only hyperparameter to fix is the number of samples K to generate. We choose $K = 2000$. The inconvenient is that we can apply this computation only in the 1-D case and for two input measures.

3.2.2 Computation of the barycenter with Iterative Bregman Projections

We will also compare our algorithm 3 to the computation of the barycenter using the Sinkhorn algorithm [10]. This algorithm solved the discretized regurlarized optimal transport problem using the optimality condition that shows that the optimal coupling P_ϵ necessarily has the form

$$P_\epsilon = diag\,(u)\,K\,diag\,(v)$$

where the Gibbs kernel is defined as

$$K := e^{-\frac{C}{\epsilon}}.$$

where $C_{ij} = d(x_i, y_j)^2$ is the cost matrix. The vectors u and v are non negative vectors. ϵ is the regularization parameter.

Figure 3 depicts the barycenter obtained for various ϵ. It's worth noting that the algorithm encounters instability issues for ϵ values below 0.005. As illustrated, the lower ϵ, the sharper the density of the barycenter.

Henceforth, we will use $\epsilon = 0.01$.

3.2.3 Computation on free grid using POT toolbox

The POT toolbox (Python Optimal Transport) [12] contains implementations of a number of founding works of OT for machine learning such as Sinkhorn algorithm and Wasserstein barycenters. In particular, it provides an algorithm based on [13] (Algorithm 2) that solves the free support regularized Wasserstein barycenter problem. Like the method §2, we optimize over the locations of the barycenter but not over its weights.

The figures 4, 5 and 6 in the Appendices shows the influence of the regularization ϵ on the free support barycenters for the 2D dataset.

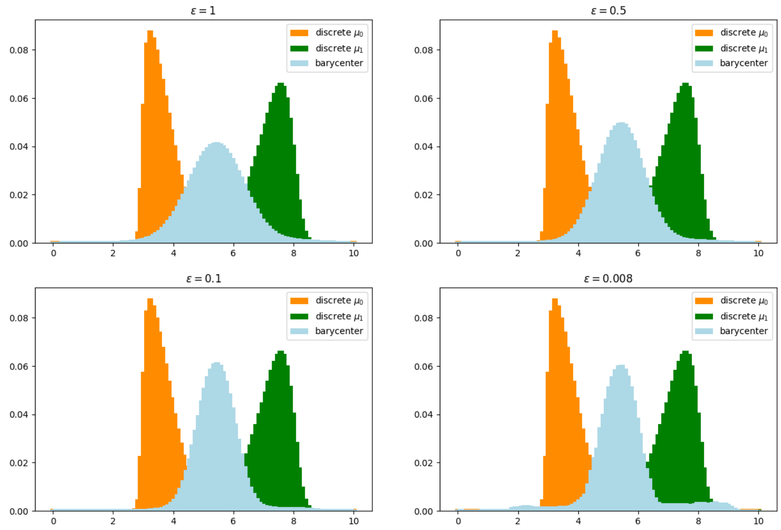


Figure 3: Histogram showing the input distributions (skew-normal distributions) and the barycenter computed using the Sinkhorn algorithm for different values of the regurlarized parameter ϵ.

3.2.4 Experience 1 : 1D skewed normal distributions

3.2.5 Experience 2 : 2D uniform distributions

4 Conclusion and Perspectives

Conclusion and perspective (1 page) Summary of the result obtained: pros and cons (limitation, problems, error in the articles, etc) Possible improvement/extension

Log-domain Sinkhorn

5 Connection with the course

What are the notions/results/algorithms presented in the course that are used or related to the one presented in this paper?

References

[1] J. Rabin, G. Peyré, J. Delon, M. Bernot in *Scale Space and Variational Methods in Computer Vision*, *Vol. 6667*, (Eds.: A. M. Bruckstein, B. M.

Ter Haar Romeny, A. M. Bronstein, M. M. Bronstein), Series Title: Lecture Notes in Computer Science, Springer Berlin Heidelberg, Berlin, Heidelberg, **2012**, pp. 435–446.

[2] A. Korotin, V. Egiazarian, L. Li, E. Burnaev.

[3] K. Cheng, S. Aeron, M. C. Hughes, E. L. Miller in Advances in Neural Information Processing Systems, *Vol. 34*, Curran Associates, Inc., **2021**, pp. 27991–28003.

[4] K. C. Cheng, S. Aeron, M. C. Hughes, E. L. Miller, Nonparametric and Regularized Dynamical Wasserstein Barycenters for Sequential Observations, **2023**.

[5] P. Dvurechenskii, D. Dvinskikh, A. Gasnikov, C. Uribe, A. Nedich in Advances in Neural Information Processing Systems, *Vol. 31*, Curran Associates, Inc., **2018**.

[6] M. Cuturi, Sinkhorn Distances: Lightspeed Computation of Optimal Transportation Distances, **2013**.

[7] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, G. Peyré, Iterative Bregman Projections for Regularized Transportation Problems, **2014**.

[8] M. Staib, S. Claici, J. Solomon, S. Jegelka, Parallel Streaming Wasserstein Barycenters, **2017**.

[9] S. Claici, E. Chien, J. Solomon, Stochastic Wasserstein Barycenters, **2018**.

[10] G. Peyré, M. Cuturi, Computational Optimal Transport, **2020**.

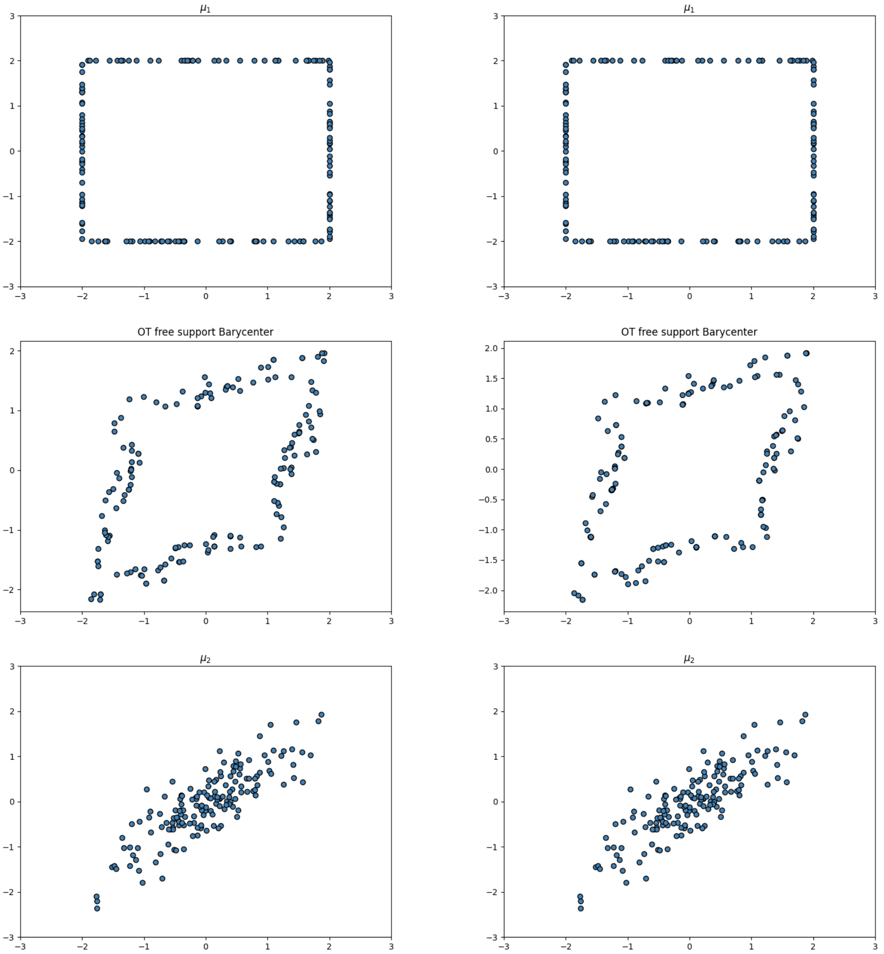
[11] R. J. McCann, *Advances in Mathematics* **1997**, *128*, 153–179.

[12] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. H. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, T. Vayer, *Journal of Machine Learning Research* **2021**, *22*, 1–8.

[13] M. Cuturi, A. Doucet, Fast Computation of Wasserstein Barycenters, **2014**.

Appendices

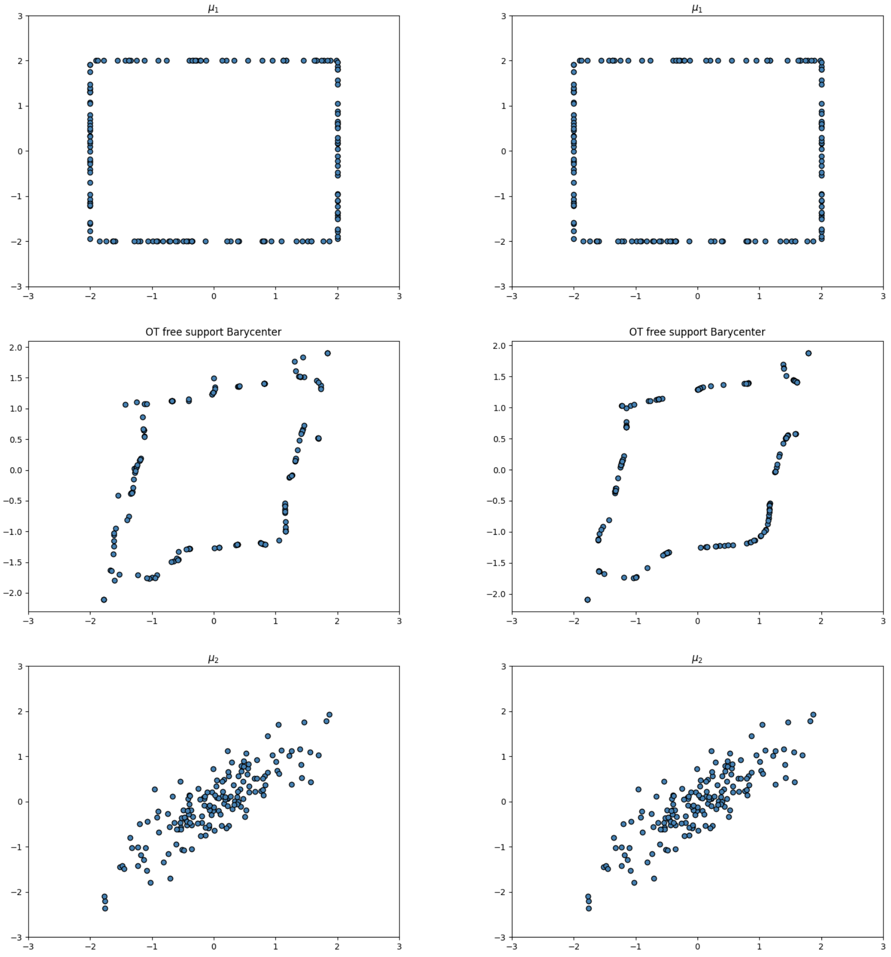
Barycenters computed with POT toolbox



(a) $\epsilon = 0.005$

(b) $\epsilon = 0.01$

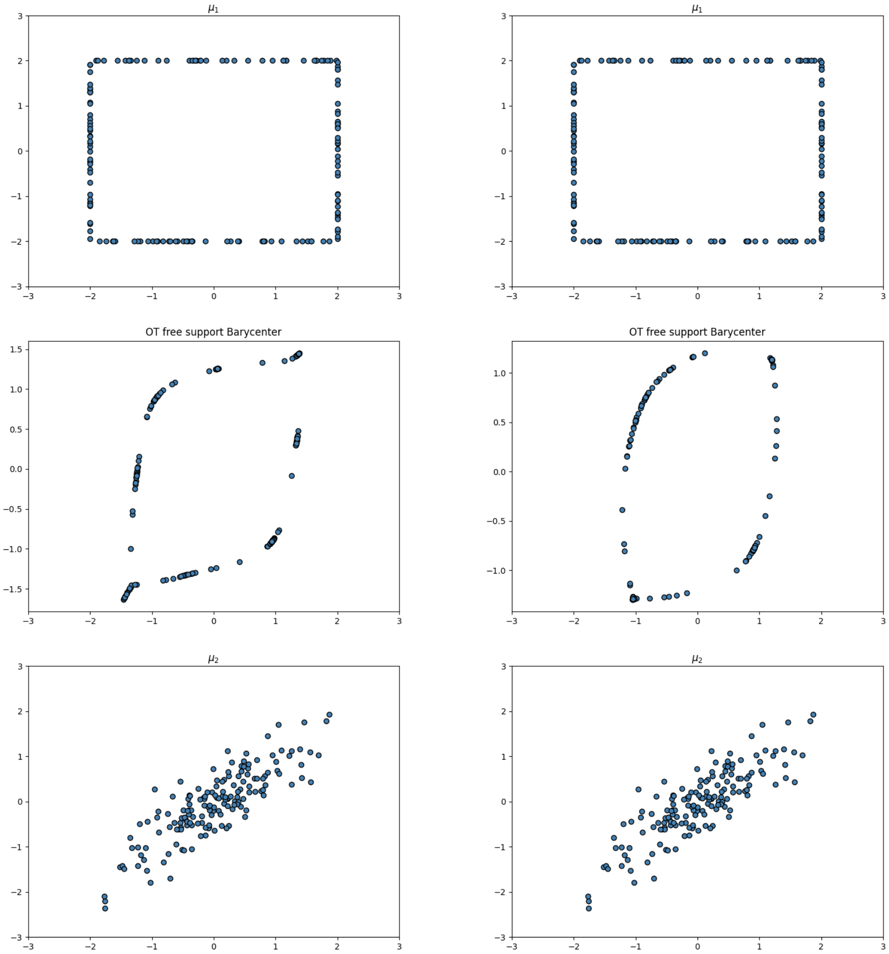
Figure 4: Free support barycenters computed with POT toolbox



(a) $\epsilon = 0.05$

(b) $\epsilon = 0.1$

Figure 5: Free support barycenters computed with POT toolbox



(a) $\epsilon = 0.5$

(b) $\epsilon = 1$

Figure 6: Free support barycenters computed with POT toolbox