Report of the article Continuous Regularized Wasserstein Barycenters

Fatima BALDE fatima.balde@telecom-paris.fr

October 19, 2024

Abstract

The article Continuous Regularized Wasserstein Barycenters studies the problem of computation of the Wasserstein Barycenter in the continuous case with regularization. This barycenter is useful for several applications, particularly in machine learning (clustering, Bayesian inference, etc.). However, in the continuous case, its computation is very difficult. For this reason, most existing methods discretise the support of the distributions before computing the barycenter. This discretisation of the supports leads to a single sample of the barycenter which is not continuous, not always desirable and may limit applications. This paper proposes a solution to this problem by computing a continuous version of the barycenter able to generate a stream of samples.

1 Introduction

1.1 Presentation of the problem

Given to distributions μ and $\nu \in \mathbf{M_1^+}(X)$,the wassertein distance between those two distributions is defined as :

$$W(\mu,\nu) = \inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathbb{X}} c(x,y)d\pi(x,y) \tag{1}$$

where $\Pi(\mu,\nu) = \{\pi \in \mathbf{M_1^+}(X^2)|(P_x)_\#\pi = \mu \text{ and } (P_y)_\#\pi = \nu\}$, P_x and P_y are the projections onto the first and second dimension respectively and $P_\#$ is the pushforward operator.

Given that problem is not easy to solve in practice, its regularized version was introduced :

$$W(\mu,\nu)_R^\xi = \inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathbb{X}} c(x,y)d\pi(x,y) + \int_{\mathbb{X}} R(\frac{d\pi}{d\xi})d\xi(x,y) \tag{2}$$

where R is a convex regulator defined as $R(t) = \epsilon(t\ln(t) - t)$(entropic regularisation) or $R(t) = \frac{\epsilon}{2}t^2$ (quadratic regularisation). Most of the time ξ is taken as $\mu \odot \nu$.

Given the definition of the Wasserstein distance between two distributions, the problem which is solved in the article is the computation of the Wasserstein barycenter which is defined as the distribution that minimises the weighted sum of the distance to a given list of distributions.
Formally given $\mu_1, \mu_2, \ldots, \mu_n \in \mathbf{M_1^+}(X)$ and weights $\lambda_1, \lambda_2 \ldots \lambda_n$ the barycenter satisfies

$$\inf_{\nu\in\mathbf{M_1^+}(X)} \sum_i W_R^{\mu_i\odot\eta}(\mu_i,\nu) \tag{3}$$

Given that we don't know ν, we take η as the distribution for the regularisation.
If we have some information on the barycenter (for example if the distributions are gaussians,

therefore the barycenter is also gaussian) , we can initialize η according to what we know.
In the case where we don't have any information on the barycenter, we will take η as an uniform distribution in the set X (in $Unif(X)$)

1.2 Previous Work

The study of this problem is in the context of solving the OT problem between two distributions and by extension the barycenter of several distributions.

The computation of the barycenter can be solved by several methods such as Sinkhorn and Bregman. However, these methods solve the problem using discrete versions of the distributions and return a discrete version of the barycenter.

Other papers have also studied the computation of the Wasserstein distance between two continuous distributions without discretizing the support. It is the case of [3], which solves the dual of (1) by parametrizing the solutions of the dual as neural networks.
This article is inspired by the latter by solving the dual problem instead of the primal and by using neural networks to approximate the solutions of the dual.

1.3 Contributions

The methods which discretize the support of the distributions have several drawbacks:

- Fixing the support a priori means solving a discrete problem, which reduces the field of applications.
- When the support is fixed a priori , the number of points required to obtain a good sample of the distribution increases exponentially with the size of the distribution.
- In addition, the computation of the barycenter leads to a single sample of the barycenter distribution reducing the possibilities of sampling.

The method developped in the article make it possible to go beyond these constraints:

- The estimated barycenter is continuous, allowing sampling to be carried out as desired.
- The method solves the dual problem and also computes the optimal transport plan between each distribution and the barycenter.
- The problem is solved using gradient descent, so only samples of the starting distributions are required.

2 Continuous Regularized Wasserstein Barycenters

2.1 Presentation of the method

2.1.1 Computation of the continuous duals

As we said earlier, the aim of the article is to solve (3) by using the dual problem.
The dual problem of (2) is defined as

$$W(\mu,\nu)_R^{\xi} = \sup_{f,g\in C(X)} \int_{\mathbb{X}} f(x)d\mu_x + \int_{\mathbb{X}} g(y)d\nu_y - \int_{\mathbb{X}} R^*(f(x)+g(y)-c(x,y))d\xi(x,y) \tag{4}$$

Where $R^*(t) = \epsilon \exp(\frac{t}{\epsilon})$ (entropic regularisation) , $R^*(t) = \frac{1}{2\epsilon}(t_+)$ (quadratic regularisation).
Once we compute the dual potentials, we can compute the transport plan using the relation :

$$d\pi(x,y) = H(x,y)d\xi(x,y) \tag{5}$$

with $H(x,y) = \frac{\exp(f(x)+g(y)-c(x,y))}{\epsilon}$ (for entropic regularisation) $H(x,y) = \frac{(f(x)+g(y)-c(x,y))}{\epsilon}$ (quadratic regularisation)

Using the definition of the dual for the primal problem between two distributions, we obtain that the dual problem for the barycenter (3) is defined as:

$$\sup_{\{(f_i,g_i)\}_{i=1}^n \subset C(X^2) \text{ and } ,\sum_{i=1}^n \lambda_i g i=0} \sum_{i=1}^{n} \lambda_i \left(\int f_i d\mu_i - \int \int R^*(f_i(x) + g_i(y) - c(x,y)) d\mu_i(x) d\eta(y) \right) \quad (6)$$

By replacing g_i with $g_i - \sum_{i=0}^n \lambda_i g i$, (6) becomes:

$$\sup_{\{(f_i,g_i)\}_{i=1}^n \subset C(X^2)} \mathbb{E}_{X_i \sim \mu_i, Y \sim \eta} \left[\sum \lambda_i \left(f_i(X_i) - R^* \left(f_i(X_i) + g_i(Y) - \sum_j \lambda_j g_j(Y) - c(X_i, Y) \right) \right) \right] \quad (7)$$

That version of the problem is a maximisation of an expectation which is easier in practice and it is the version that we'll solve.

Given the distributions $\mu_1, \ldots, \mu_n$, the weights $\lambda_1, \ldots, \lambda_n$, the distribution η which is initialized according to the knowledge on the barycenter, the regularization function R^* the dual problem is solved as follows :

- Initialisation of the duals, which are parameterised as neural networks
- At each epoch ,samples of the the distributions are used to compute the empirical mean of equation (7) .
- A gradient descent is applied to f_i and g_i (As we want to maximise (7) we consider -E as our loss function and apply gradient descent to the neural networks f_i and g_i

Algorithm 1: Stochastic gradient descent to solve the regularized barycenter problem (11)

Input : distributions $\mu_1, \ldots, \mu_n$ with sample access, weights $(\lambda_1, \ldots, \lambda_n)$, dual regularizer R^*, regularizing measure η, cost function c, gradient update function `ApplyGradient`.

Initialize parameterizations $\{(f_{\theta_i}, g_{\phi_i})\}_{i=1}^n$;
for $l \leftarrow 1$ **to** n_{epochs} **do**
  $\forall i \in \{1, \ldots, n\}$: sample $x^{(i)} \sim \mu_i$; sample $y \sim \eta$;
  $\bar{g} \leftarrow \sum_{i=1}^n \lambda_i g_{\phi_i}(y)$;
  $F \leftarrow \sum_{i=1}^n \lambda_i \left(f_{\theta_i}(x^{(i)}) - R^* \left(f_{\theta_i}(x^{(i)}) + g_{\phi_i}(y) - \bar{g} - c(x^{(i)}, y) \right) \right)$;
  for $i = 1, \ldots, n$: `ApplyGradient`(F, θ_i); `ApplyGradient`(F, ϕ_i);
return dual potentials $\{(f_{\theta_i}, g_{\phi_i})\}_{i=1}^n$.

Figure 1: Algorithm to compute the duals

Thus defined, the duals represent continuous functions that can be used to compute the optimal transport plan between the distributions and the estimated barycenter using the equation (5)

2.1.2 Computation of the continuous Barycenter

Given the optimal transport plan , the barycenter is equal to $(Py)_{\#}\pi_i$ regardless of i. This computation of the barycenter can be performed using several methods:

- If the π_i distributions have densities, we can use numerical integration to compute $(Py)_{\#}\pi_i(x) = \int \pi_i(x,y)dy$

- We can also use Markov chain Monte Carlo (MCMC) methods to sample according to the π_i (again if we know the densities of π_i).

As these two methods are limited by the prior knowledge required of the density, we will instead compute an approximation of the monge map π_i using one of the following methods and taking as our cost function $c(x, y) = ||x - y||_2^2$

a) We can use the gradients of the dual potentials and compute the map monges which are equal to $T_i(x) = x - 1/2\nabla f_i(x)$ in the case where the cost function is defined as above. Given that f_i is a neural network, $f_i(x)$ is calculated by passing a batch sample over f_i.
b) We can directly use the formula (5) and parameterise the monge maps as neural networks

$$Ti = \underset{T:X \to X}{\arg\min}\, \mathbb{E}_{X\sim\mu_i, Y\sim\eta}\left[c(T(X), Y)H(X, Y)\right]$$

Here again, the T_i are parameterised as neural networks and updated by considering the expectation defined above as the loss function and by performing backpropagation on the networks.

The calculation using a is simpler than the calculation using b but when the size of the distributions becomes large b can be more stable.
With these two methods, each distribution i will correspond to a monge map T_i and to compute a sample of the barycenter, we compute a sample of the distribution i of our choice which we pass to the corresponding neural network T_i (method b)) or in the algorithm of the method a). We can also compute T_i for all i and take the average sample, but all the T_i should agree in practice.

2.2 Numerics

I coded the algorithm for the method in python and carried out some simple tests to evaluate its performance.
As explained previously the first part of the method consists of computing dual potentials which are parameterised as neural networks and are trained to maximise the expectation of the formula (7).
This algorithm takes several variable parameters such as the value of the regularisation, the size of the batches for sampling the distributions and the learning rate.
The second part of the method consists of calculating the continuous barycentre. Of the methods given in the article, I have chosen to test two of the 3 methods that use the Monge Map.
Similarly, one of these two methods requires the Monge Map T to be parameterised as a neural network. As a result, there are also variables to take into account, in particular the value of the regularisation, the size of the batches and the learning rate.

Since these variables are not predefined, I first chose to study the impact of these variables on the results of the algorithm before carrying out tests on several scenarios.

2.2.1 Impact of epsilon

Computing the regularised wassertein requires an epsilon regularisation value which varies according to the distributions.
Therefore before carrying out all the tests, It may be useful to measure the impact of epsilon by computing the barycenter. 6 centred Gaussians with different variances were used as distributions. Since we know that the barycenter of a gaussian is also a gaussian, the regularisation distribution chosen is a centred and reduced gaussian

The gaussians distributions were chosen because the closed form of their barycenter is known and can be obtained using the fixed point approach algorithm [2].

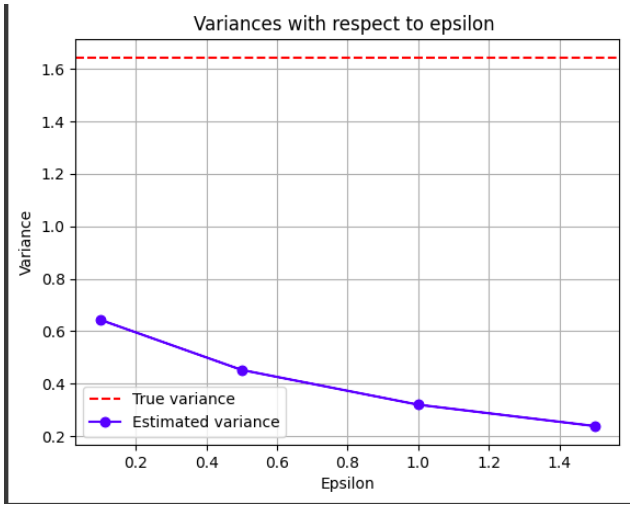

Figure 2: Evolution of the estimated variance with respect to the regularisation

The results obtained are not optimised, but we can clearly see that the higher the regularisation, the further the estimated variance is from the true variance of the barycenter.
Also, when the value of the regularisation is very small, the gradients can explode and so the f_i can produce nan values.

2.2.2 Impact of the batch size

Batch size has a major impact on the speed of the algorithm and is not predefined.
A study on the impact of this variable was also carried out using the same Gaussian distributions as before.
As can be seen from the previous figure, the variance increases slowly with batch size.

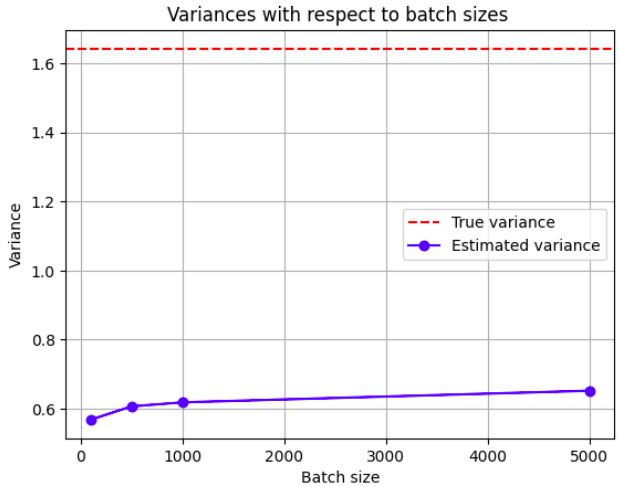

Figure 3: Evolution of the estimated variance with respect to the batch size

This may suggest that any batch size can be used. But in practice, I noticed that the regularisation was close to the true value when the batch size was large enough to describe the distribution well.

2.2.3 Test on 1D gaussians

The study of the value of the regularisation and the size of the batches made it possible to reduce the time needed to find the optimum parameters for calculating the barycenters.
Therefore, tests was first carried out on six centred barycenters of different variances (0.5, 0.8,1,1.5, 1.8, 2).
We chose centred barycenters because, when computing the barycenter, the distributions are first centred before the proper computation, and the monge map are transposed according to the average mean. Therefore, we can directly use centred distributions.
The two methods a) and b) of the algorithm were used to compute the monge maps.
To compare with a discretisation method, Sinkhorn's algorithm was used to compute an estimation of the barycenter.
With the parameters batchsize=8000 , nepochs=1000 and epsilon=0.1, I obtain the results in the next figure.

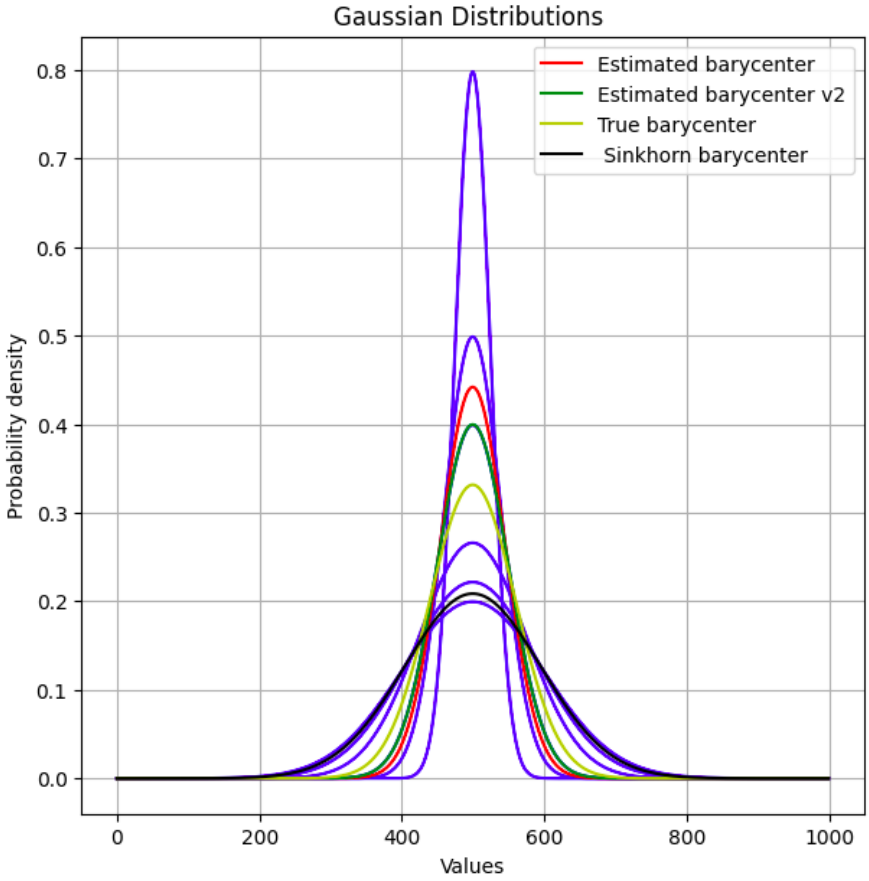

Figure 4: Gaussian distributions and corresponding barycenters

As can be seen from the figure 4 above, the distribution of the barycenters computed with the two versions of the article method are much closer to the true distribution barycenter than that computed with sinkhorn.

The difference between the variances of the true distribution and that of the estimate are 0.20 and 0.3 for the two version of the article method and 0.69 for sinkhorn.

2.2.4 Test on 2D gaussians

Tests were also carried out with six 2D Gaussians distributions of different means $((0 \quad 0.)\ (0.5 \quad 0.5)\ (1 \quad 1)\ (2 \quad 2)\ (1 \quad 0)\ (0 \quad 1))$

and with respective variances

$$\begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0.5 \\ 0.5 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 2 & 1.5 \\ 1.5 & 2 \end{pmatrix} \begin{pmatrix} 2.5 & 2 \\ 2 & 2.5 \end{pmatrix}$$

The following results were obtained with the following parameters batchsize=8000 , nepochs=1000 and reg=0.1 .

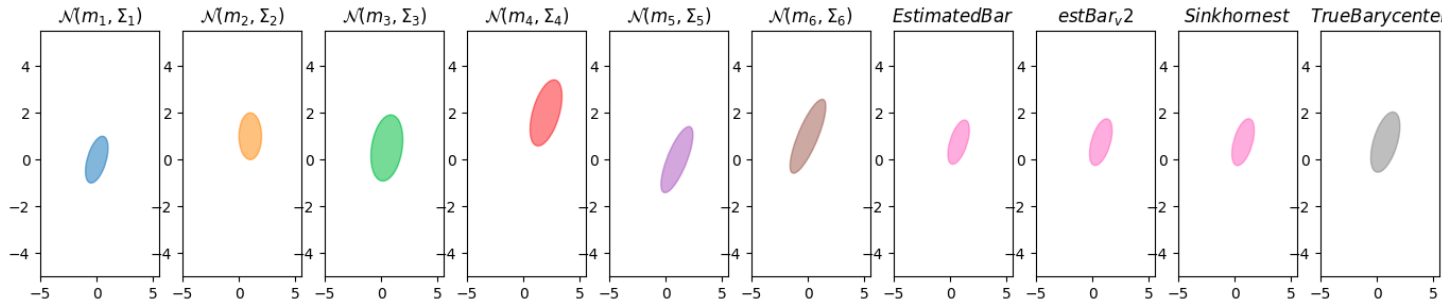

Figure 5: Six 2D gaussians distributions, the estimated barycenters v1 and v2 , Sinkhorn barycenter, True barycenter

Visually the results between the two versions of the algorithm and sinkhorn's seem to be the same.
Therefore we computed $||\Sigma_{est} - \Sigma*||_F$ to evaluate the differences , Σ_{est} is the estimated covariance and $\Sigma*$ the true covariance (computed with the fixed point algorithm).
Since six gaussians were used, six monge map are estimated by the algorithm. Therefore, all of them were used to point out the difference between them and also to compare all of them with their estimation with the true covariance.

We can see in the following figure 6 that the estimations using the method a) are more stable than those of the method b). They are also always better than sinkhhorn's estimation.
The results obtained with the method a) are also better in general than sinkhhorn's estimation.
Some of the monge maps estimated produce slightly worse results but this is due to the instability of the computations(convergence problem, the number of iterations may not be sufficient etc...).

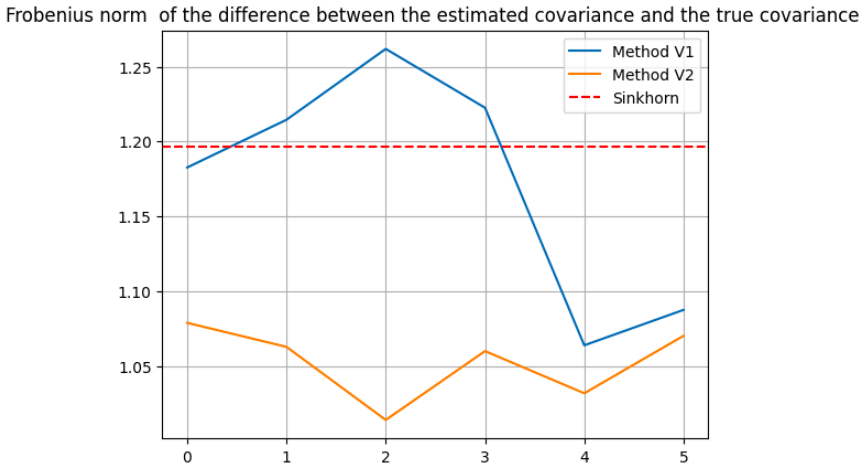

Figure 6: Frobenius norm of the difference between the estimated covariance and the true covariance

2.2.5 Test on Rectangle distributions

After carrying out tests on Gaussian distributions with a known closed shape, tests were carried out on more complex distributions: rectangular shapes. To take into account the training time for the neural networks and the complexity of the distributions, only 3 rectangles of different lengths 1,1,2 and widths 1,2,1 were chosen.

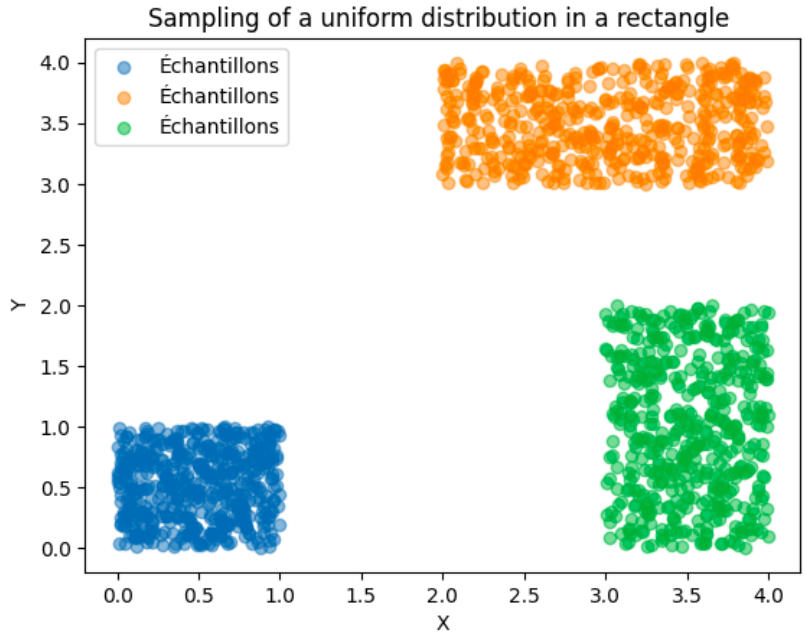

Figure 7: Sample of Rectangle distributions

Unlike the Gaussian case, the distribution for regularisation is chosen as an uniform distribution over X=[0,4]*[0,4].

We used the following parameters to calculate the duals: nepochs=600 reg=0.1 batch-size=4000. For the a) method we found that the sampling of the barycenter have the following form:

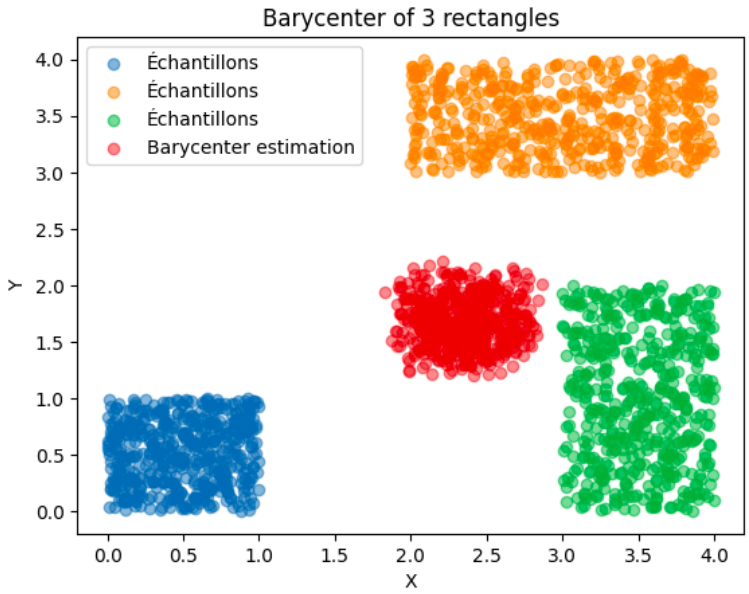

Figure 8: Barycenter obtained with a)

For the method b)we found that the sampling of the barycenter have the following form:

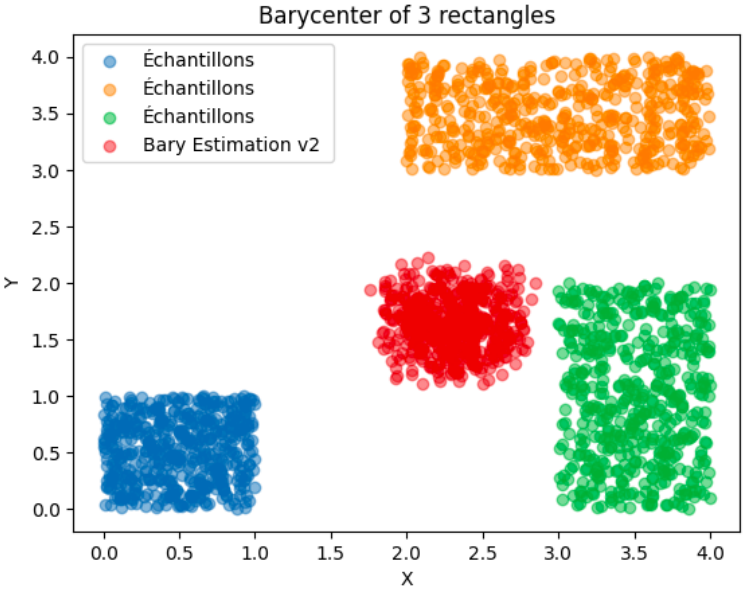

Figure 9: Barycenter obtained with b)

To make a comparison, we also computed the barycenter with sinkhorn As can be seen

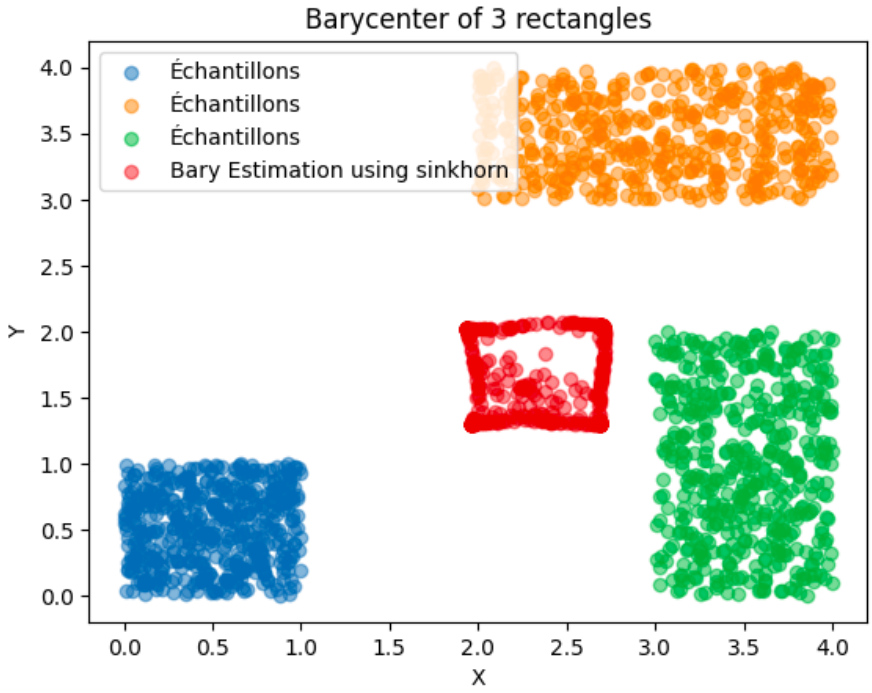

Figure 10: Barycenter with Sinkhorn

from the previous figures, the barycenter found by taking uniform weights is also a uniform distribution over a rectangle. The methods in the article and the sinkhorn method seem to agree on the general shape and position of the distribution.
It seems that the article's method calculates the barycentre better because the distribution is uniform over a rectangle, whereas Sinkhorn's method calculates a barycentre whose distribution is more concentrated on the edges of the rectangle.
Moreover the shape is more accurate with the article's method, not to mention the possibility of generating several samples of the distribution, which is not the case with Sinkhorn.

3 Conclusion

Computing the barycenter of a list of distributions using a discrete support is a good idea as long as you manage to have a number of samples that are representative of the distribution.
However, this method will fail when the distributions are complex, requiring a large number of samples, or when the size of the distributions becomes large.
The continuous barycenter method gives slightly better results in dimension 1 and 2 on Gaussian distributions compared with sinkhorn. It seems to handle slightly better more complex distributions, such as distributions with geometric shapes, etc.. It also has one very important aspect compared with discretisation methods, which is the fact that it has a continuous barycenter, allowing as many samples as required to be generated, which can be very useful depending on the application.
The main disadvantage of this method is the need to find the optimum parameters for convergence. Gradient descent may not converge if the parameters (batch size, number of epochs, regularisation value, learning rate) are not optimal for the distributions to be tested. You have to find the right parameters each time and the more complex the distribution

becomes, the more you have to increase the number of epochs etc... I also pushed the tests further by trying with digits but the loss curve was not smooth and the barycenter just looked like a cloud of points. The same test carried out in the article gave better results but the number of epochs used (20.000) was not feasible with the resources I have. Evaluating more advanced tests requires a good GPU.

4 Connexion with the course

The paper studied is closely related to the concepts studied in class.
the paper studies a subject which is a continuation of several themes (Kantorovitch Relaxation, Entropic Regularization, Wassertein distance etc.) seen in the course.
First of all, the problem starts from the wassertein regularised barycenter seen in the numerical tours.
Then there's the study of the related dual problem studied in class, which was also used in the article.
I also compared the method in the article with the sinkhorn algorithm seen in class

References

[1] Lingxiao Li, Aude Genevay, Mikhail Yurochkin, Justin Solomon *Continuous Regularized Wasserstein Barycenters*. arxiv 1511.05355

[2] Pedro C. Alvarez-Esteban , E. del Barrio1 , J.A. Cuesta-Albertos2 and C. Matran *A fixed-point approach to barycenters in Wasserstein space*. arXiv:1511.05355v3

[3] Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. *"Large-scale optimal transport and mapping estimation"*. arXiv:1711.02283 (2017)

[4] Hicham Janati, Boris Muzellec, Gabriel Peyré, Marco Cuturi *"Entropic Optimal Transport between Unbalanced Gaussian Measures has a Closed Form"*. arXiv:2006.02572v2