Computational Optimal Transport

Stochastic Wasserstein Barycenters

Inès VATI*

MATHÉMATIQUES
VISION
APPRENTISSAGE

JAN 19, 2024

*École Normale Supérieure Paris-Saclay, Master MVA

Table of contents

Overview

Problem statement

Article : Claici, Chien, and Solomon 2018, *Stochastic Wasserstein Barycenters*

- Compute the Wasserstein barycenter ν of a set of any probability measures $\{\mu_j\}_{j=1}^{J}$ defined of a metric space $(\mathcal{X}, d)$ with $\mathcal{X} \subset \mathcal{R}^D$
- Primal problem

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

Problem statement

Article : Claici, Chien, and Solomon 2018, *Stochastic Wasserstein Barycenters*

- Compute the Wasserstein barycenter ν of a set of any probability measures $\{\mu_j\}_{j=1}^{J}$ defined of a metric space $(\mathcal{X}, d)$ with $\mathcal{X} \subset \mathcal{R}^D$
- Primal problem

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

- Approximation of the barycenter

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

  with optimized support $\Sigma = \{x_i\}_{i=1}^{M}$
- **Problem.** Find a minimizer of (1) with $\Sigma \subset \mathcal{X}$ and $|\Sigma| = M$

Overview

Proposed method

- Dual problem

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

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

- **Stochastic Gradient Ascent** to optimize the potentials $\{\phi_j\}_{j=1}^{J}$ (*ascent step*)
- **Fixed point iteration** to optimize the positions $\{x_i\}_{i=1}^{M}$ (*snap step*)

Ascent step

Gradient $\frac{\partial F}{\partial \phi_j}$ depend on the terms

$$a_j^i = \int_{V_{\phi_j}^i} d\mu_j(y) \qquad\qquad b_j^i = \int_{V_{\phi_j}^i} y\, d\mu_j(y)$$

$$= \mathbb{E}_{y\sim\mu_j}\left[\mathbf{1}_{y\in V_{\phi_j}^i}\right] \qquad\qquad = \mathbb{E}_{y\sim\mu_j}\left[y.\mathbf{1}_{y\in V_{\phi_j^i}}\right]$$

where $\mathbf{1}$ is the indicator function and the *power cell* $V_{\phi_j}^i$ of point x_i is

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

where $d(x, y)^2 = \|x - y\|_2^2$.

Ascent step

Algorithm Ascent step

for $j = 1, \dots, J$ **do**
  $z^{(0)} \leftarrow 0$
  while $\left\| \widehat{\frac{\partial F}{\partial \phi_j}} \right\|_2^2 > \epsilon_{ascent}$ **do**
    Compute $\hat{a}_j^l$ by Monte Carlo approximation
    $z^{l+1} \leftarrow \beta z^{(l)} + \widehat{\partial_{\phi_j} F}(\phi_j)$ {Nesterov acceleration}
    $\phi_j^{(l+1)} \leftarrow \phi_j^{(l)} + \alpha z^{(l+1)}$
  end while
end for

We will use $\epsilon_{ascent} = 10^{-4}$ and set a maximum number of iterations for the while loop to $T_{max} = 1500$
And fix $\alpha = 0.05$ and $\beta = 0.99$ like in (Claici, Chien, and Solomon 2018)

Snap step

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

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

Intuition :

$$\frac{\partial F_{dual}}{\partial \phi_j^i} = \frac{1}{M} - \int_{V_{\phi_j}^i} d\mu_j(y) = 0$$

- $\rightarrow$ Each cell V_ϕ^i contains as much mass as its associated source point x_i
- $\rightarrow$ the fixed point iteration moves each point to the center of its power cell.

Algorithm Ascent and Snap algorithm for computing Stochastic Wasserstein Barycenters

for $t = 1, \ldots, T$ **do**
    Update $\{\phi_j\}$ with algorithm $\{$Ascent step$\}$
    Compute $\hat{b}_j^i$ by Monte Carlo approximation
    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 $\Sigma^* = \{x_i\}_{i=1}^{M}$

Theoretical and Empirical Convergence

Assumptions

- At least one of the input distribution μ_j is absolutely continuous with respect to the Lebesgue measure.
- The domain $\mathcal{X}$ is a compact subset of the Euclidean space $\mathbb{R}^d$.

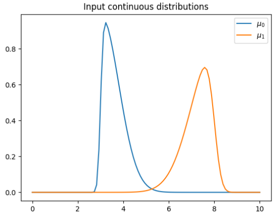
Garantees

- The global minimizer of (1) efficiently approximate the true barycenter as $M \to \infty$
- No garantee of convergence toward ν_M^*
- The ascent snap algorithm converges to a local minimum
- Algorithm monotonically $\searrow F_{primal}$ (1)
- The support of the estimated barycenter is empirically demonstrated to be contained within the support of the true barycenter
- Empirical evidences $\implies$ convergence even in cases where these assumptions are not met

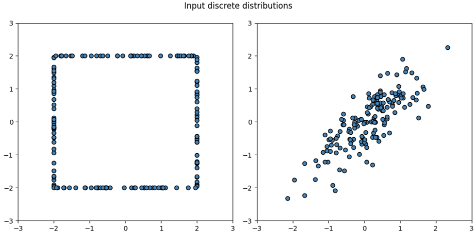
Overview

Experiments and Numerical Analysis

Applications to both 1D and 2D cases



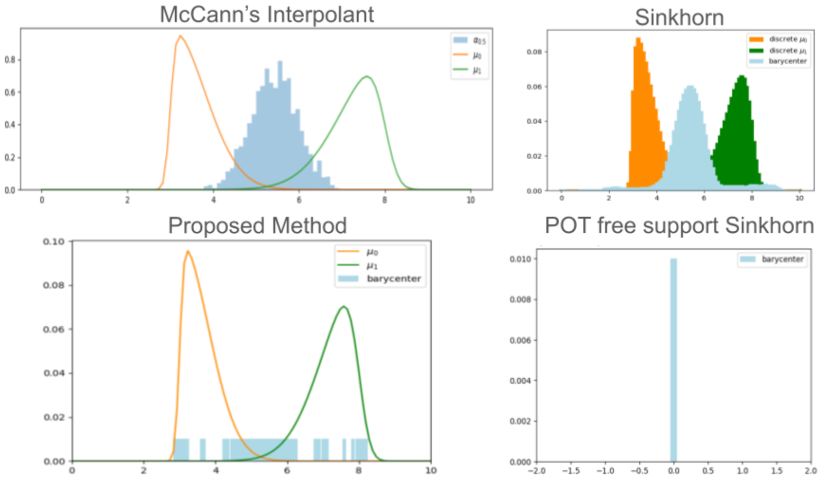
Continuous skewed normal distributions



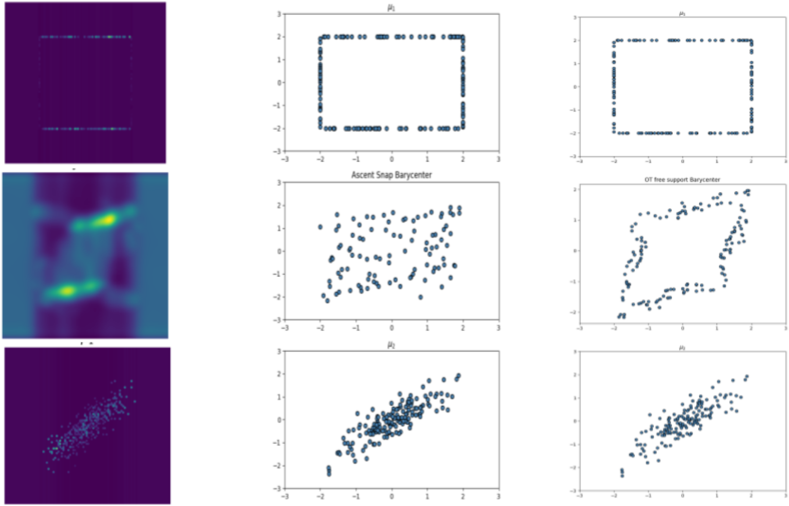
Discrete uniform distributions

Comparison to existing methods

- McCann's interpolation
- Iterative Sinkhorn algorithm
- Free support Sinkhorn algorithm (Cuturi and Doucet 2014) from POT toolbox (Flamary et al. 2021)


McCann's Interpolant
$\alpha_{0.5}$
μ_0
μ_1
Sinkhorn
discrete μ_0
discrete μ_1
barycenter
Proposed Method
μ_0
μ_1
barycenter
POT free support Sinkhorn
barycenter

Experiment 2



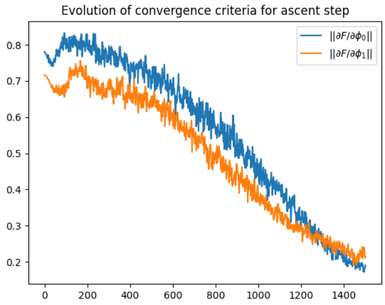
Sinkhorn Algorithm (left), Proposed method (middle), Free support Sinkhorn using POT (right)

Computation time

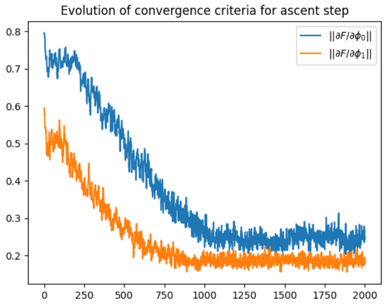
Experiment	McCann	Iterative Bregman	POT free support	Ascent Snap
1D skewnorm	0.02 ± 0.01	0.22 ± 0.02	0.74 ± 0.28	7814.27 ± 0.37
2D discrete	-	2.18 ± 0.34	0.38 ± 0.30	16351.66 ± 0.23

Table: Computation time (in seconds) for the different methods. The experiments were conducted three times for the Ascent snap algorithm and ten times for the others.

Ascent step convergence



(a) Two 1D continuous distributions



(b) Two 2D discrete distributions

Figure: Evolution of $\left\|\frac{\partial F}{\partial \phi_j}\right\|$ during the ascent step for each experiment. In the x-axis, is the number of interations of the while loop.

Ascent step time

Experiment	Ascent step	Snap step
1D skewnorm	8309.44 ± 0.33	5.28 ± 0.23
2D discrete	11938.70 ± 0.34	9.81 ± 0.34

Table: Computation time (in seconds) for each step of the proposed algorithm. The experiences were repeated 3 times.

Overview

Notes

- Huge computation time
- Robust and versatile method
- Support of $\hat{\nu}$ not necessarily included in the support of ν

Perspectives

- Test another choice for the initialization of x_i
- Compare with the Log-domain Sinkhorn Algorithm

References

Claici, Sebastian, Edward Chien, and Justin Solomon (June 6, 2018). *Stochastic Wasserstein Barycenters*. arXiv: 1802.05757[cs,math,stat]. URL: http://arxiv.org/abs/1802.05757 (visited on 12/21/2023).

Cuturi, Marco and Arnaud Doucet (June 17, 2014). *Fast Computation of Wasserstein Barycenters*. arXiv: 1310.4375[stat]. URL: http://arxiv.org/abs/1310.4375 (visited on 01/14/2024).

Flamary, Rémi et al. (2021). "POT: Python Optimal Transport". In: *Journal of Machine Learning Research* 22.78, pp. 1–8. ISSN: 1533-7928. URL: http://jmlr.org/papers/v22/20-451.html (visited on 01/13/2024).