No-collision Transportation Maps

Levon Nurbekyan, Alexander Iannantuono & Adam M. Oberman
Department of Mathematics & Statistics, UCLA & Department of Mathematics & Statistics, McGill University
lnurbek@math.ucla.edu, alexander.iannantuono@mail.mcgill.ca, adam.oberman@mcgill.ca

Half-space preserving transportation maps

Given a probability measure $\mu \in \mathcal{P}(\mathbb{R}^d)$ and a Borel measurable map $T : \mathbb{R}^d \to \mathbb{R}^d$, the **push-forward** of μ by T is a probability measure $T\#\mu$ that is given by

$$(T\#\mu)(B) = \mu(T^{-1}(B)) \quad (B \text{ Borel}).$$

Furthermore, for probability measures $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$ and a Borel measurable map $T : \mathbb{R}^d \to \mathbb{R}^d$, we say that T **transports** μ to ν if $\nu = T\#\mu$.

A natural problem is to search for T with useful properties. This leads to the theory of *optimal transport*: given a cost $c(x, y)$ for transporting unit mass from x to y, one searches for a transportation (correspondence) that minimizes the cost of transporting μ to ν:

$$\min_{\nu = T\#\mu} \int_{\mathbb{R}^d} c(x, T(x))\, d\mu(x).$$

In this work, our goal is to find a transportation map that satisfies the **no-collision** property: for all $x_1 \neq x_2 \in \text{supp}(\mu)$, we have that

$$(1-\lambda)x_1 + \lambda T(x_1) \neq (1-\lambda)x_2 + \lambda T(x_2) \quad (\lambda \in (0,1)).$$

In words, this says that the point-masses will not collide if travelling in a straight line with constant speeds. When solving the optimal transport problem, T has the no-collision property if $c(x, y) = h(x - y)$ with h strictly convex and even; though this is typically quite expensive to calculate exactly.

A map $T : \Omega \subset \mathbb{R}^d \to \mathbb{R}^d$ is said to be **half-space preserving** if for every $x_1 \neq x_2 \in \Omega$, there exists a direction $v \in \mathbb{S}^{d-1}$ such that

$$\langle x_2, v\rangle - \langle x_1, v\rangle \geq 0,\ \langle T(x_2), v\rangle - \langle T(x_1), v\rangle \geq 0,$$

and at least one of the inequalities is strict.

(Theorem 2.1) A map $T : \Omega \subset \mathbb{R}^d \to \mathbb{R}^d$ is half-space preserving if and only if it has the no-collision property.

Relation to k-d trees

The data structure we are relying on for this construction are k-d trees. Each node in the tree represents a collection of points (or a measure) and the two children of each node (except the leaves) is a bipartition of the parent into sets (or measure) of equal size, i.e. half the size. An example of the bipartition can be seen in the figures below.

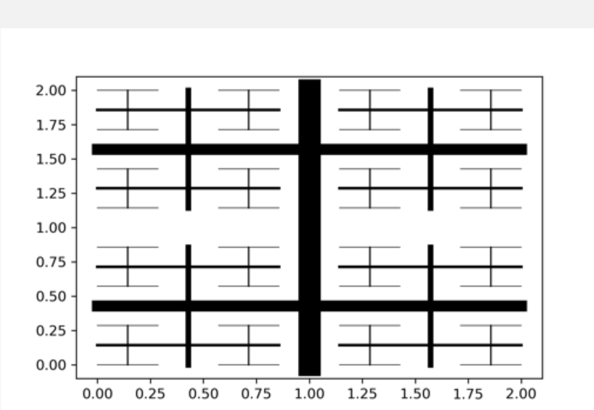

Figure: Bisection on grid

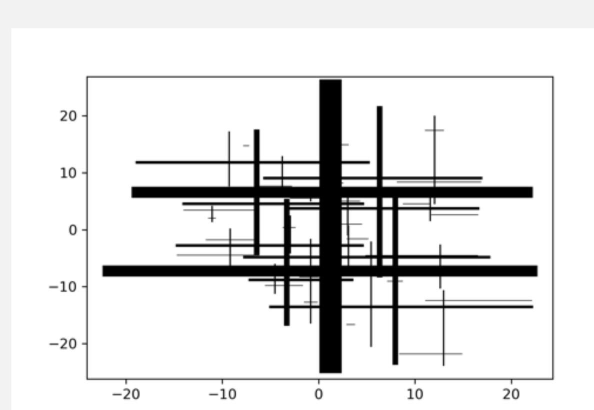

Figure: Bisection on Gaussian

Map construction via Theorem 2.1

Assume that $\mu \in \mathcal{P}(\mathbb{R}^d)$ is absolutely continuous and $\{v_k\}$ is an arbitrary sequence of directions. Denote $\Omega_0 := \{\mathbb{R}^d\}$. Since μ is absolutely continuous, there exists $h_1 \in \mathbb{R}^d$ such that

$$\mu(\langle x, v_1\rangle \leq h_1) = \mu(\langle x, v_1\rangle > h_1).$$

We then denote $A_0 := \{\langle x, v_1\rangle \leq h_1\}$ and $A_1 := \{\langle x, v_1\rangle > h_1\}$ to be the two half-planes, with $\Omega_1 := \{A_0, A_1\}$.

To every $x \in A$, we assign a binary sequence $s(x)$, with

$$s(x) = \begin{cases} 0, & x \in A_0 \\ 1, & x \in A_1. \end{cases}$$

Pick a set from Ω_1, say A_1, and find h_2 such that

$$\mu(A_1 \cap \{\langle x, v_2\rangle \leq h_2\}) = \mu(A_1 \cap \{\langle x, v_2\rangle > h_2\}),$$

and denote the new sets $A_{10} := A_1 \cap \{\langle x, v_2\rangle \leq h_2\}$ and $A_{11} := A_1 \cap \{\langle x, v_2\rangle > h_2\}$, $\Omega_2 := \{A_0, A_{11}, A_{10}\}$. We update the binary sequence for $x \in A_1$ as follows

$$s(x) = \begin{cases} 10, & x \in A_{10} \\ 11, & x \in A_{11}, \end{cases}$$

and repeat. If the directions $\{v_k\}$ are chosen appropriately, the sequence $s(x)$ is a bijection between $\text{supp}(\mu)$ and $\{0,1\}^\infty$.

If ν is another absolutely continuous measure, then the same directions and subsets will give us the binary sequence $r(x)$. Then, the transportation map from μ to ν is simply $t = r^{-1} \circ s$.

(Theorem 2.2) With some reasonable assumptions on μ and ν, and the way the directions $\{v_k\}$ are chosen, it can be shown that t is a.e. Borel measurable from int(supp(μ)) to int(supp(ν)), and also a.e. continuous.

Runtime comparisons

- Construction of a no-collision map via k-d trees costs $O(n\log(n))$, where n is the number of points.
- When the cost is separable under addition, [1] propose a clever back-and-forth approach that achieves $O(n\log(n))$ on the dual problem using c-transforms. However, the algorithm requires a regular grid discretization and is not suitable for discrete measures with irregular support.
- In the quadratic cost case, [2] achieves near-linear complexity, with $\tilde{O}(n\varepsilon^{-3}(C\log(n)\varepsilon^{-1})^d)$, though it might not have the no-collision property.

Experimental results: *Transporting points in* $\mathbb{R}^2$

The algorithm for constructing the map T was coded in both the Python and C programming languages. It was tested for $n = 64, 256, 1024, 4096$ points in Python, and up to 4^{12} points in C to measure speed. The repository for the Python coded can be found by scanning the QR code on the poster.

Further work for implementation:

- To be run on NVIDIA GPUs with the CUDA architecture
- For higher dimensions using the GPU version
- For fractional weights on points

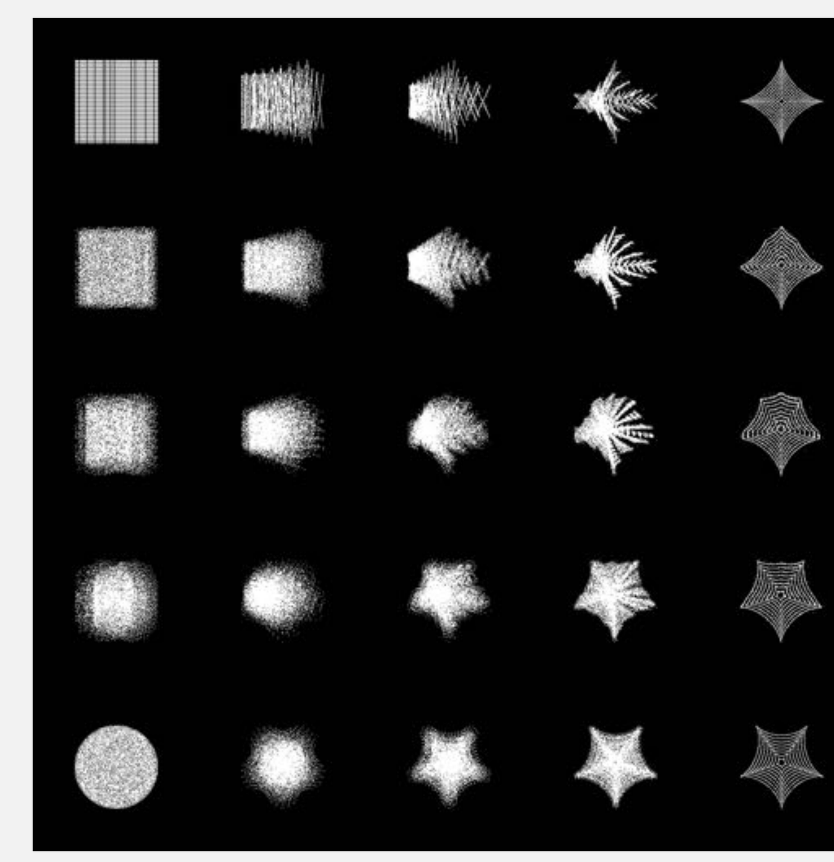

Figure: Barycentres of four starting shapes (shown in corners) at various weights

Note: For the tables below, $C(x, x+y) = \|y\|_p^q, p = 1, 2, \infty, q = 1, 2.$

Cost	Method	n: 64	256	1024	4096
$\|\cdot\|_2^2$	OT	0.97	1.02	1.23	1.14
	HV (ratio)	1.49 (1.54×)	1.23 (1.21×)	1.32 (1.07×)	1.16 (1.02×)
	SH (ratio)	0.98 (1.02×)	1.05 (1.03×)	1.26 (1.03×)	1.16 (1.02×)
	LEX (ratio)	6.52 (6.72×)	6.59 (6.46×)	7.00 (5.72×)	7.13 (6.27×)
$\|\cdot\|_2$	OT	1.01	0.77	0.78	0.84
	HV (ratio)	1.15 (1.13×)	0.83 (1.08×)	0.81 (1.04×)	0.85 (1.02×)
	SH (ratio)	1.04 (1.02×)	0.80 (1.03×)	0.81 (1.04×)	0.87 (1.04×)
	LEX (ratio)	2.25 (2.22×)	2.18 (2.83×)	2.27 (2.92×)	2.31 (2.75×)
$\|\cdot\|_1^2$	OT	1.27	1.47	1.22	1.09
	HV (ratio)	1.74 (1.38×)	1.75 (1.19×)	1.37 (1.13×)	1.15 (1.05×)
	SH (ratio)	1.28 (1.01×)	1.49 (1.01×)	1.25 (1.03×)	1.10 (1.01×)
	LEX (ratio)	9.42 (7.44×)	10.37 (7.07×)	11.27 (9.27×)	11.13 (10.21×)
$\|\cdot\|_\infty^2$	OT	1.39	1.23	1.06	1.00
	HV (ratio)	1.79 (1.29×)	1.49 (1.21×)	1.19 (1.12×)	1.05 (1.05×)
	SH (ratio)	1.41 (1.01×)	1.26 (1.02×)	1.08 (1.03×)	1.03 (1.02×)
	LEX (ratio)	6.58 (4.75×)	5.97 (4.85×)	5.89 (5.57×)	5.97 (5.94×)

Table: Average costs of sets of points sampled from $\mathcal{N}(0,1)$ to a new set of points sampled from $\mathcal{N}(0,1)$ (then transformed to a 3:1 aspect ratio and rotated $90°$ counter-clockwise) measured in various cost functions

References

[1] M. Jacobs and F. Léger, “A fast approach to optimal transport: the back-and-forth method,” *Preprint*, 2019, arXiv:1905.12154 [math.OC]. [Online]. Available: https://arxiv.org/abs/1905.12154

[2] J. Altschuler, F. Bach, A. Rudi, and J. Weed, “Approximating the quadratic transportation metric in near-linear time,” *arXiv preprint arXiv:1810.10046*, 2018.

[3] J. L. Bentley, “Multidimensional binary search trees used for associative searching,” *Commun. ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975. [Online]. Available: http://doi.acm.org/10.1145/361002.361007

[4] R. Flamary and N. Courty, “POT Python Optimal Transport library,” 2017. [Online]. Available: https://github.com/rflamary/POT