Computing Kantorovich-Wasserstein Distances on d-dimensional histograms using $(d+1)-$partite graphs

Gennaro Auricchio[a], Federico Bassetti[b], Stefano Gualandi[a], Marco Veneroni[a]

[(a)] Università degli Studi di Pavia, Dipartimento di Matematica "F.Casorati", [(b)] Politecnico di Milano, Dipartimento di Ingegneria Matematica

gennaro.auricchio01@universitadipavia.it, stefano.gualandi@unipv.it, marco.veneroni@unipv.it, federico.bassetti@polimi.it

Abstract

TASK: *To compute the distance between two d-dimensional histograms having n bins. For instance, images are 2-dimensional histograms with n bins (pixels)*

PROBLEM: *The mathematical tool used to compute this distance requires the solution of an optimization problem with up to n^2 variables*

IDEA: *To exploit the structure of the cost function in order to reduce the number of variables of the optimization problem*

1. Kantorovich-Wasserstein distance

The Kantorovich-Wasserstein distance is a metric between probability distributions or d-dimensional histograms.

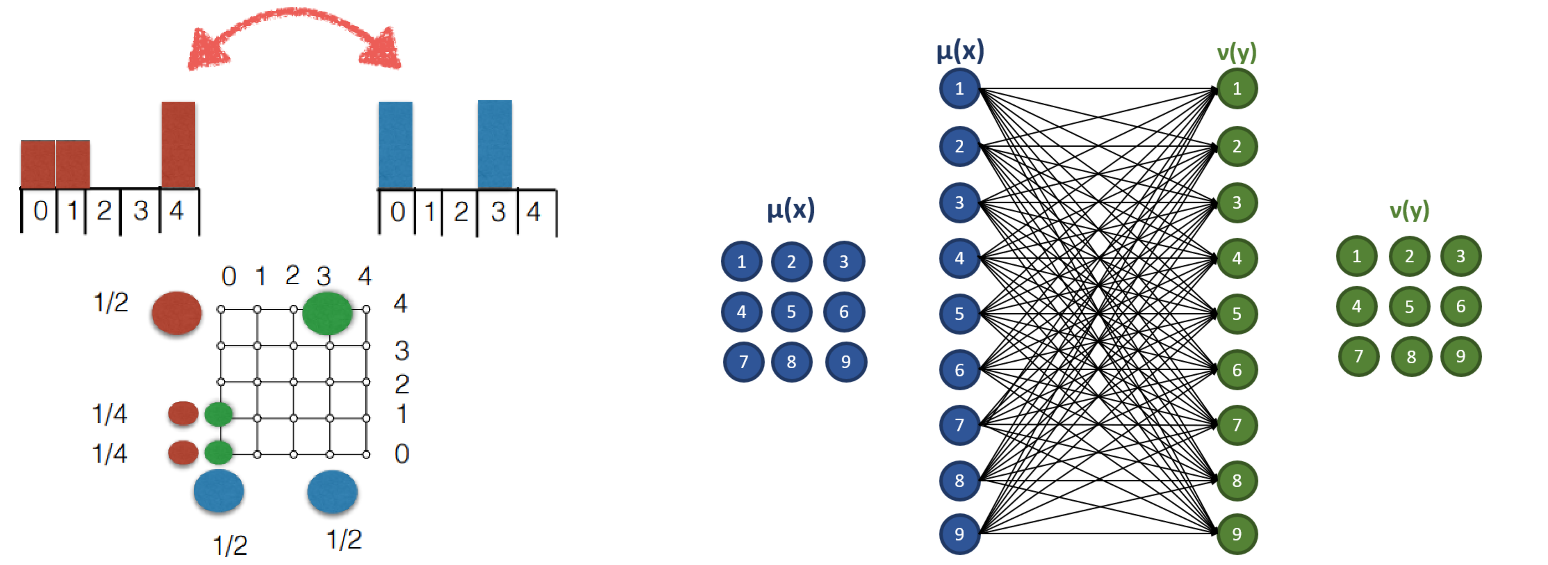

Figure 1: *Left: transport map π (green dots) between $1D$-histograms (red and blue dots) with $n = 5$ bins. Right: bipartite graph associated to constrained optimization problem used to compute the Kantorovich-Wasserstein distance for $2D$-histograms with $n = 9$ bins*

The idea of the Kantorovich-Wasserstein distance is to find the optimal way to map a probability μ to a probability ν where moving a unit mass from x to y costs $c_{x,y}$. Figure 1 (left).

Whenever the cost is $c_{x,y} = \|x-y\|_2^2$, one gets the Wasserstein distance of order 2

$$W_2(\mu,\nu) := \min \sum_x \sum_y c_{x,y}\pi_{x,y}$$

where the minimum is over all the probability measures π with marginals ν and μ, i.e.

$$\sum_x \pi_{x,y} = \nu_y, \quad \text{and} \quad \sum_y \pi_{x,y} = \mu_x.$$

The computation of W_2 distances between histograms with n bins requires the solution of a constrained optimization problem.

Indeed, the standard approach to compute W_2 distances between $2D$ histograms with n bins is to solve the corresponding uncapacitated min cost flow problem [2] on a bipartite graph, with $2n$ nodes (2 times the number of bins) and n^2 arcs (one for all the possible costs $c_{x,y}$). See Figure 1 (right).

The solution of this problem requires $O(n^3 \log(n))$ time, which for large n is computationally too heavy.

2. Our contribution

In [1] we propose a novel approach for computing the W_2 distance that exploits the structure of the cost function to reduce the number of arcs.

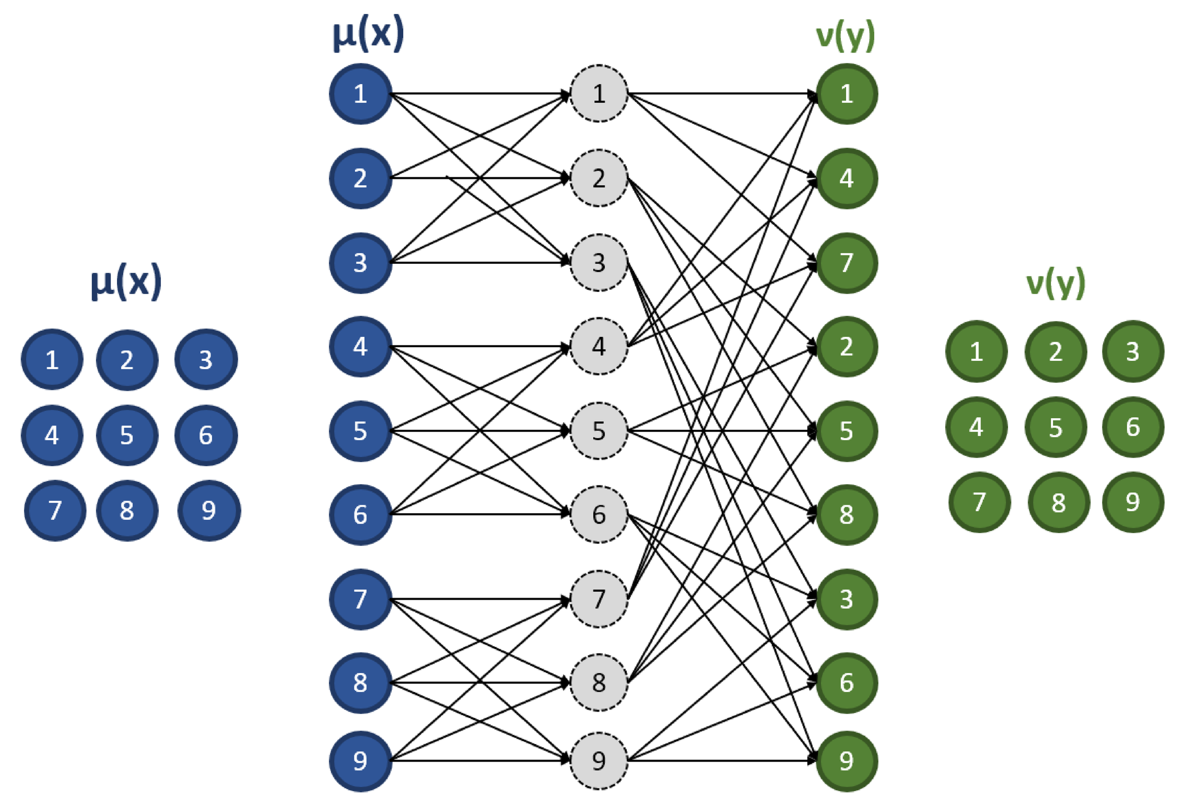

Figure 2: *$3-$partite graph reformulation for the computation of W_2*

Since the cost function is separable, $c_{x,y} = (x_1-y_1)^2 + (x_2-y_2)^2$, it can be computed as the concatenation of the costs along the two main directions: $(x_1,x_2) \rightarrow (y_1,x_2)$ and $(y_1,x_2) \rightarrow (y_1,y_2)$. See Figure 2.

CONTRIBUTION: *We prove that the W_2 distance between d-dimensional histograms can be computed as a flow problem on $(d+1)$-partite graph, with $(d+1)n$ nodes and $dn^{1+\frac{1}{d}}$ arcs.*

- Our method requires $dn^{1+\frac{1}{d}}$ arcs while the standard bipartite graph method requires n^2.
- The method can be adapted to any cost function that is separable, *i.e.* can be written as a sum of independent contributions.
- The method provides an exact solution.

3. Numerical Results

As problem instances, we use the gray scale images proposed by the DOTMark benchmark, and a set of d-dimensional histograms obtained by biomedical data measured by flow cytometry.

Table 1: *Comparison on Flow Cytometry data with increasing value of d.*

			Bipartite Graph			$(d+1)$-partite Graph		
N	d	n	$\|V\|$	$\|A\|$	Runtime	$\|V\|$	$\|A\|$	Runtime
16	2	256	512	65 536	0.024 (0.01)	768	8 192	**0.003 (0.00)**
	3	4 096	8 192	16 777 216	38.2 (14.0)	16 384	196 608	**0.12 (0.02)**
	4	65 536	*out-of-memory*			327 680	4 194 304	**4.8 (0.84)**
32	2	1 024	2 048	1 048 756	0.71 (0.14)	3072	65 536	**0.04 (0.01)**
	3	32 768	*out-of-memory*			131 072	3 145 728	**5.23 (0.69)**

Figure 3: *DOTmark benchmark: Classic, Microscopy, and Shapes images.*

Table 2: *Comparison on 32×32 images. The runtime (in secs) is given as "Mean (StdDev)". The gap to the optimum opt is computed as $\frac{UB-opt}{opt}\cdot 100$, where UB is the upper bound computed by Sinkhorn's algorithm. Each row reports the averages over 45 instances.*

	EMD[4]	Sinkhorn [3]				Bipartite	3-partite
		$\lambda=1$		$\lambda=1.5$			
Image Class	Runtime	Runtime	Gap	Runtime	Gap	Runtime	Runtime
Classic	24.0 (3.3)	6.0 (0.5)	17.3%	8.9 (0.7)	9.1%	0.54 (0.05)	**0.07 (0.01)**
Microscopy	35.0 (3.3)	3.5 (1.0)	2.4%	5.3 (1.4)	1.2%	0.55 (0.03)	**0.08 (0.01)**
Shapes	25.2 (5.3)	1.6 (1.1)	5.6%	2.5 (1.6)	3.0%	0.50 (0.07)	**0.05 (0.01)**

	Improved Sinkhorn [5]						3-partite
	$\lambda=1$		$\lambda=1.25$		$\lambda=1.5$		
Image Class	Runtime	Gap	Runtime	Gap	Runtime	Gap	Runtime
CauchyDensity	0.22 (0.15)	2.8%	0.33 (0.23)	2.0%	0.41 (0.28)	1.5%	**0.07 (0.01)**
Classic	0.20 (0.01)	17.3%	0.31 (0.02)	12.4%	0.39 (0.03)	9.1%	**0.07 (0.01)**
GRFmoderate	0.19 (0.01)	12.6%	0.29 (0.02)	9.0%	0.37 (0.03)	6.6%	**0.07 (0.01)**
GRFrough	0.19 (0.01)	58.7%	0.29 (0.01)	42.1%	0.38 (0.02)	31.0%	**0.05 (0.01)**
GRFsmooth	0.20 (0.02)	4.3%	0.30 (0.04)	3.1%	0.38 (0.04)	2.2%	**0.08 (0.01)**
LogGRF	0.22 (0.05)	1.3%	0.32 (0.08)	0.9%	0.40 (0.13)	0.7%	**0.08 (0.01)**
LogitGRF	0.22 (0.02)	4.7%	0.33 (0.03)	3.3%	0.42 (0.04)	2.5%	**0.07 (0.02)**
Microscopy	0.18 (0.03)	2.4%	0.27 (0.04)	1.7%	0.34 (0.05)	1.2%	**0.08 (0.02)**
Shapes	0.11 (0.04)	5.6%	0.16 (0.06)	4.0%	0.20 (0.07)	3.0%	**0.05 (0.01)**
WhiteNoise	0.18 (0.01)	76.3%	0.28 (0.01)	53.8%	0.37 (0.02)	39.2%	**0.04 (0.00)**

References

[1] G. Auricchio, F. Bassetti, S. Gualandi, and M. Veneroni. Computing Kantorovich-Wasserstein distances on d-dimensional histograms using (d+1)-partite graphs. *arXiv preprint arXiv:1805.07416*, 2018.

[2] F. Bassetti, S. Gualandi, and M. Veneroni. On the computation of Kantorovich-Wasserstein distances between 2D-histograms by uncapacitated minimum cost flows. *arXiv preprint arXiv:1804.00445*, 2018.

[3] M. Cuturi. Sinkhorn distances: Lightspeed computation of Optimal Transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.

[4] Y. Rubner, C. Tomasi, and L.J. Guibas. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision*, pages 59–66. IEEE, 1998.

[5] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)*, 34(4):66, 2015.