Estimating Quadratically Regularized Wasserstein Distance on k-Connected Graphs

Austin Du

Abstract

Given a collection of points in $\mathbb{R}^d$, with some mass distributions across those points, the practical question arises asking for the most efficient method of transferring one mass distribution to another. One answer to this question results from a computationally intense quadratically-regularized (QR) optimal transport to calculate QR-Wasserstein distance. We intend on bypassing this calculation with two efficient algorithms to estimate this QR-distance, one using new methods of random connectivity, and another using a novel geometric approach.

1 Introduction

The study of optimal transport attempts to find the most efficient method of transforming an initial mass distribution to a target mass distribution, given various constraints. This efficiency is measured via a "cost function" that places weights on the individual aspects of each transport, resulting in the "Wasserstein distance" or "earth-mover's distance". In doing so, one quantifies the difference between distributions, allowing classification or clustering of such distributions. The classical formulation of the problem places these distributions in metric spaces with respective probability measures [1]. However, modern developments in computer science and data representation motivates a similar optimal transport problem over graphs. This is the problem presently discussed.

In this construction, we are given a graph and a distribution over the vertices and aim to find transports that move the entire mass along the edges. We represent such a transport as a mapping from each edge to the weight along that edge, using the vector $\mathbf{w} \in \mathbb{R}^{|E|}$. As with traditional Optimal Transport, different choices of a cost function $c(\mathbf{w})$ will produce different optimal weights and transports, but the most common choice is a cost linear to the length of each edge, $c(\mathbf{w}) = \mathbf{l} \cdot \mathbf{w}$ where $\mathbf{l}$ is the vector of edge-lengths. We can set this as our objective function, but minimizing solutions for linear functions are rarely unique. Issues arise when computing transports algorithmically, as small variance in input could lead to unconstrained minima.

This issue has been historically addressed by adding a non-linear regularization term to the objective function. Many fruitful results come from considering entropic regularization (addition of a positive entropy term to each edge weight), most notably a rapid solution via Sinkhorn's Algorithm [1]. Another solution uses quadratic regularization, where a scaled quadratic term is added to each edge-weight, making the objective function

$$c(\mathbf{w}) = \sum_{e \in E} l_e w_e + \frac{\alpha}{2} w_e^2.$$

Recently, this approach has also produced compelling results, featuring an efficient Newtonian algorithm by Essid and Solomon [1]. This paper focuses on the quadratic approach.

Such an optimization problem can be solved naively with quadratic programming over an incidence matrix, but the computation can quickly become cumbersome. This paper attempts to efficiently approximate a transport distance using the same algorithm on a more sparsely connected graph. Given a point cloud of vertices, the traditional method of creating a connected graph connects all vertices to their k nearest neighbors. In [2] however, Linderman, et al. proposed the creation of Near-Neighbor graphs whose vertices are uniformly randomly connected to k of the K nearest neighbors, as well as an efficient "partition algorithm" to do so. This paper studies the abilities of these new graphs to estimate transport distance on the dense original graph.

2 Background Information

2.1 Definitions

Let X be a random collection of n points in $[0,1]^d$. According to the empirical results of the [2], we have $K \propto \log n$ and $k \propto \log\log n$ such that connecting each node uniformly randomly to k of its nearest K neighbors produces a connected graph with high probability.

Let $G = (E, V)$ be a connected graph, where vertices are points in $[0,1]^k$. Define $l \in \mathbb{R}^{|E|}$ to be the Euclidean edge lengths of G, where if $e = (u, v)$, $l_e = ||u - v||$. Define $\rho_0, \rho_1 \in \mathbb{R}^{|V|}$ to be the source and sink distributions on the nodes of G with $\mathbf{1} \cdot \rho_0 = \mathbf{1} \cdot \rho_1 = 1$ and let $\mathbf{f} = \rho_1 - \rho_0$. Then construct the incidence matrix $D \in \{-1, 0, 1\}^{|V| \times |E|}$ such that

$$D_{ev} = \begin{cases} -1 \text{ if } (v, w) \in E \text{ for some } w \in V \\ \ \ 1 \text{ if } (w, v) \in E \text{ for some } w \in V \\ \ \ 0 \text{ otherwise} \end{cases}$$

Note that each column has only two non-zero entries: -1 and 1, and thus $D^T\mathbf{1} = \mathbf{0}$. We use this to represent all valid source-to-sink transports as vectors $\mathbf{w}$ that satisfy $D\mathbf{w} = f$.

We now define the quadratically-regularized Wasserstein distance on G as:

$$\mathcal{W}(G, \mathbf{f}, \alpha) = \begin{cases} \min_{\mathbf{w} \in \mathbb{R}^{|E|}} & \sum_{e \in E} |w_e| l_e + \frac{\alpha}{2} \sum_{e \in E} w_e^2 \\ \text{s.t.} & D\mathbf{w} = f \end{cases}$$

Normally for directed graphs, there is an extra constraint that forces $\mathbf{w} > 0$, but this is relaxed in our case since we assume bi-directional flow is always possible.

2.2 Algorithms

We discuss the viability of multiple graph-generating algorithms henceforth described as k-Nearest-Neighbors (kNN), k-of-K-Near-Neighbors (kNEAR), and the Partition Algorithm. All algorithms take as input an arbitrary point cloud and connect neighborhoods of points to produce a connected graph (or graph with single, large component). kNN is seen as the traditional method of connecting, but kNEAR and Partition were proposed by [2] to reduce edge count of the resulting graphs.

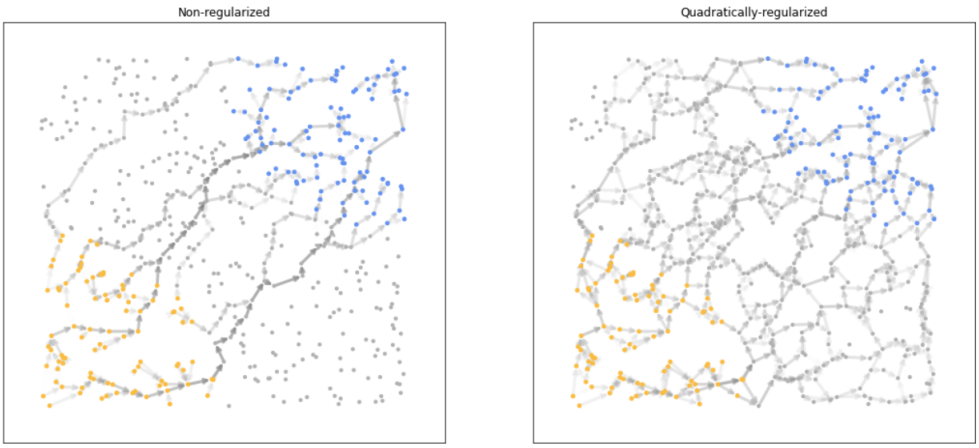

Figure 1: Comparing non-regularized transport to QR-regularized transport

kNN operates by simply connecting each node to its k nearest neighbors. Computationally, this consists of finding neighbors for each node and creating an edge, which can be done in $O(|V||E|)$. kNEAR operates similarly, but by considering a size-K neighborhood and uniformly randomly creating edges with probability $p \propto \frac{\log\log|V|}{\log|V|}$. kNEAR does not produce a connected graph, but it does produce a "giant component" of size $|V| - o(|V|)$ with high probability [2].

The Partition Algorithm emulates kNEAR in its random selection of neighbors, but does so more efficiently. Empirical results also show that its resulting graphs have high probability of being connected, so we consider PART as an alternative to kNN. If we aim to connect points using k neighbors chosen from a K-sized neighborhood, PART splits points into $\lfloor \frac{K}{k} \rfloor$ partitions and performs kNN within a randomly chosen partition. See Figure 1 for resulting graphs.

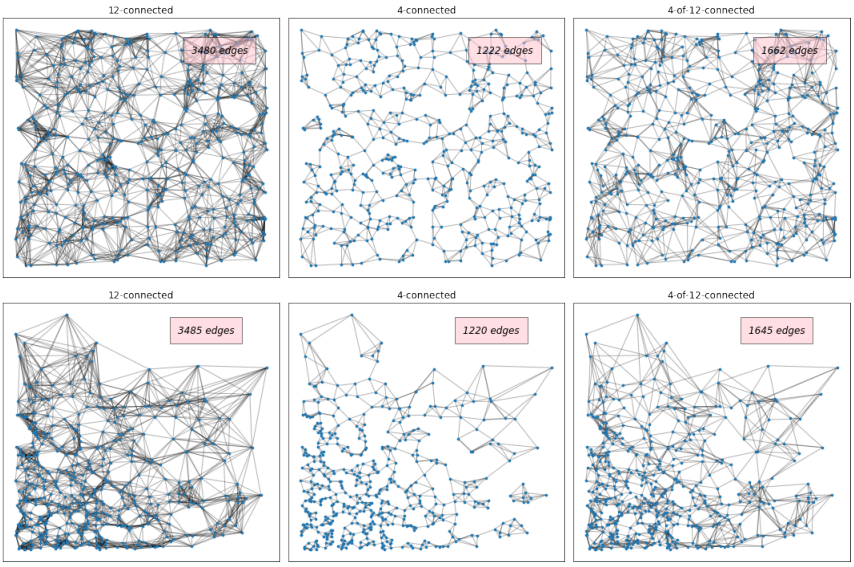

Figure 2: Comparing kNN, kNear, and Partition methods of connecting for various point clouds

We can see through Figure 2 that connecting a uniform point cloud of 500 points using 12-Nearest-

Neighbors required 3480 edges, where the 4-of-12 Partition Algorithm required only 1662 edges while maintaining the connectedness. This can be contrasted with simply using 4-Nearest-Neighbors which has a high probability of generating a disconnected graph. A similar analysis was conducted on a non-uniform point cloud.

3 The Partition Algorithm

3.1 Existence of the Optimal Transport

We are looking for an optimal $\mathbf{w}$ such that $D\mathbf{w} = \mathbf{f}$. However, since the incidence matrix D could be rank deficient, it still remains to be shown that our problem is well-defined, in particular that $D\mathbf{w} = \mathbf{f}$ has a solution provided that $\mathbf{1} \cdot \mathbf{f} = 0$. We handle this with the following theorem.

Theorem 3.1. *Let $D \in \mathbb{R}^{m \times n}$ be the incidence matrix for a connected graph and let $\mathbf{f} \in \mathbb{R}^m$ have the property that $\mathbf{1} \cdot \mathbf{f} = 0$. Then $D\mathbf{w} = \mathbf{f}$ has at least one solution.*

Proof. We first use the fact that incidence matrices of connected graphs have rank $(V-1)$. Thus, we can perform a singular value decomposition on D to get $D = U\Sigma V^T$, where U and V are orthonormal and

$$\Sigma = \left[\begin{array}{cccc|c} \lambda_1 & & & & \\ & \ddots & & & \\ & & \lambda_{V-1} & & \mathbf{0} \\ & & & 0 & \end{array}\right]$$

Write U as $[\mathbf{u}_1, \ldots, \mathbf{u}_V]$. From SVD, we know that these are the eigenvectors of DD^T, specifically with eigenvalues $\lambda_1^2, \ldots, \lambda_{V-1}^2, 0$. Since D has rank $(V-1)$, we know that for $1 \leq i \leq V-1$, $\lambda_i \neq 0$ and thus $\mathbf{u}_V$ is the only eigenvector with eigenvalue 0.

Now create the pseudo-inverse of Σ

$$\Sigma^+ = \left[\begin{array}{cccc} \frac{1}{\lambda_1} & & & \\ & \ddots & & \\ & & \frac{1}{\lambda_{V-1}} & \\ & & & 0 \\ \hline & & \mathbf{0} & \end{array}\right]$$

We claim that $\mathbf{w} = V\Sigma^+U^T\mathbf{f}$ is a solution to $U\Sigma V^T\mathbf{w} = \mathbf{f}$ after calculating the following:

$$\begin{aligned} U\Sigma V^T\mathbf{w} &= (U\Sigma V^T)(V\Sigma^+U^T)\mathbf{f} \\ &= (U\Sigma)(\Sigma^+U^T)\mathbf{f} \\ &= U\left(I - \begin{bmatrix} 0 & \cdots & 0 \\ & & 1 \end{bmatrix}\right)U^T\mathbf{f} \\ &= \mathbf{f} - U\begin{bmatrix} 0 & \cdots & 0 \\ & & 1 \end{bmatrix}U^T\mathbf{f} \\ &= \mathbf{f} - [\mathbf{1}]\mathbf{f} \\ &= \mathbf{f} \end{aligned}$$

so indeed solutions exist. □

3.2 Apply Partition Algorithm and Measure Efficiency

Calculating the QR Wasserstein Distance on $G_k(X)$ requires running a quadratic program which is largely encumbered by the large matrix D with size $|V| \times |E|$. With a more sparse connection given by the Near-Neighbor algorithm, we can significantly reduce the size of D while still maintaining connection of the graph. Thus, given a point cloud X, we propose a more efficient method to estimate a QR Wasserstein distance $\mathcal{W}(G_k(X))$ by first creating the sparser connected graph $G_{k,K}(X)$ with the Partition Algorithm and then using traditional Quadratic Programming methods to find $\mathcal{W}(G_{k,K}(X))$.

We apply this process over 4 different types of random graphs shown below in Figure 3: uniform points with disjoint sources/sinks, uniform points with overlapping sources/sinks, skewed points with disjoint sources and sinks, and skewed points with overlapping sources/sinks.

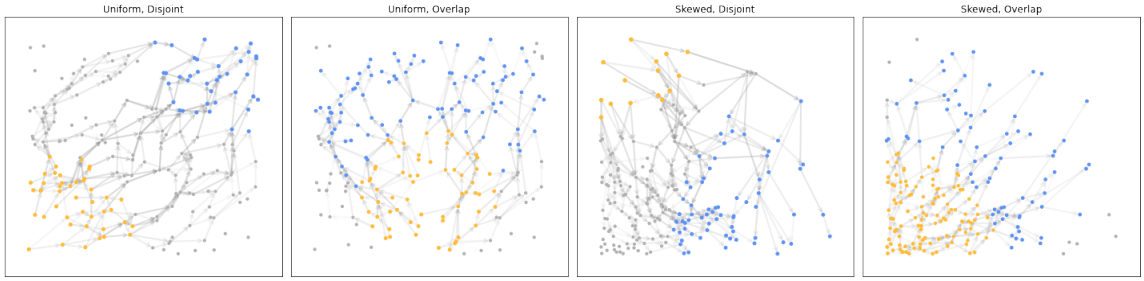

Figure 3: Demonstrating the different types of simulated QR Optimal Transport

After plotting the weights, we see from Figure 4 that the Partition algorithm consistently overestimates the kNN QR Wasserstein Distance by a factor of around 1.2 regardless of point or mass distribution. This constant scaling gives merit to the Partition Algorithm as an estimator for kNN distance, if a re-scaling step is applied thereafter.

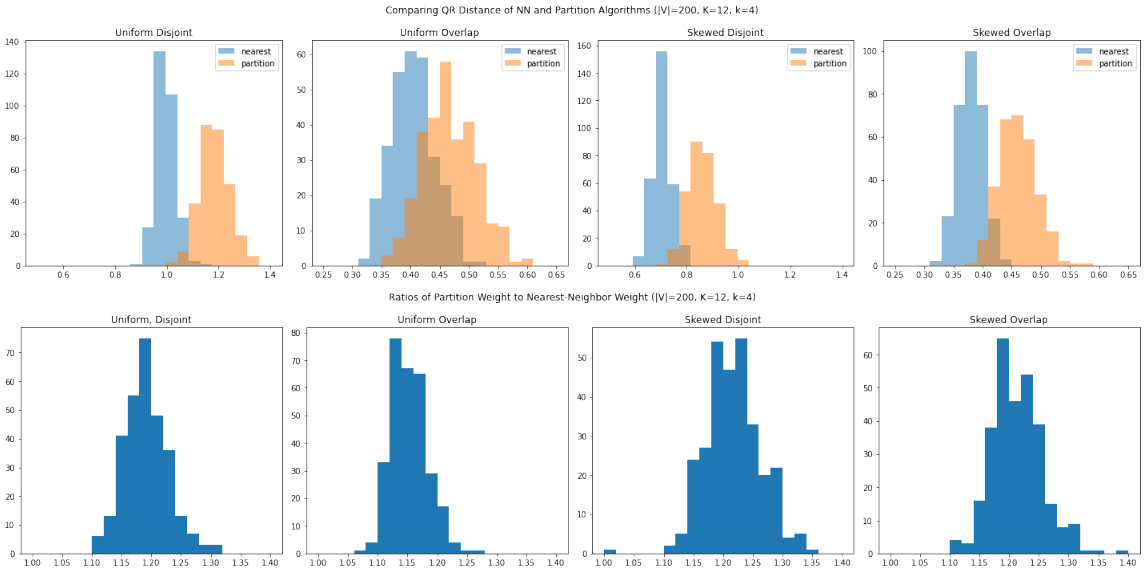

Figure 4: Comparing calculated QR Wasserstein distances over various graphs with kNN-connecting and Partition-connecting

3.2.1 Efficiency

Even if the Partition Algorithm may be random in its predictions, its value mainly lies in the speed of its calculation. Within the examples of Figure 2, we see the edge count drastically reduced by around half when connecting 500 points in $\mathbb{R}^2$. This translates to much faster quadratic programming as seen in all four cases of Figure 5. On average, the it took 1.8 seconds to calculate Optimal Transport on a kNN-connected graph with 200 vertices while it required an average of on .45 seconds on a Partition-connected graph. This difference only becomes more pronounced with more points to connect: on 300 points, kNN runs in 3.23s and Partition runs in 0.64s; on 500 points, kNN runs in 9.42s and Partition runs in .89s.

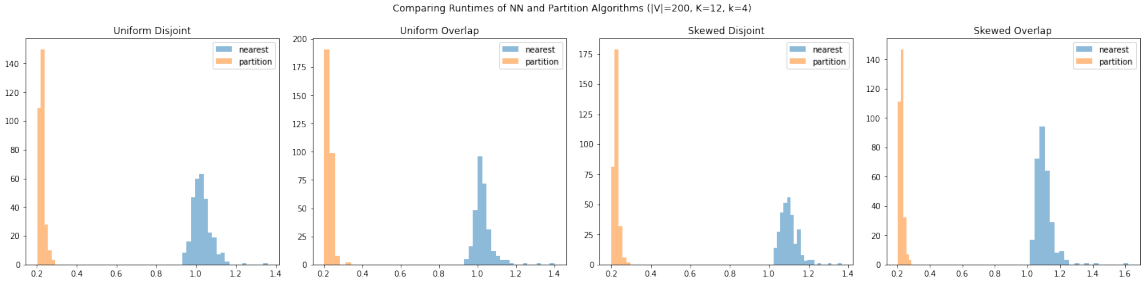

Figure 5: Comparing run-time of an Optimal Transport Algorithm on a kNN-connected graph vs. a Partition-connected graph

3.2.2 Re-scaling

Another issue that arises when using the Partition Algorithm to estimate the Wasserstein Distance is an unintentional increase in strength of the regularization factor $\frac{\alpha}{2}\sum_e w_e^2$. Canonically, α is used to control the "dispersion" of the transport, where low α-values induce sparser, concentrated transports while large

α-values encourage homogeneous, low-density transports (See Figure 6). In the objective function, we see this as α governing the ratio between the true transport weight $\sum_i w_i l_i$ and the regularizer $\sum_i w_i^2$.

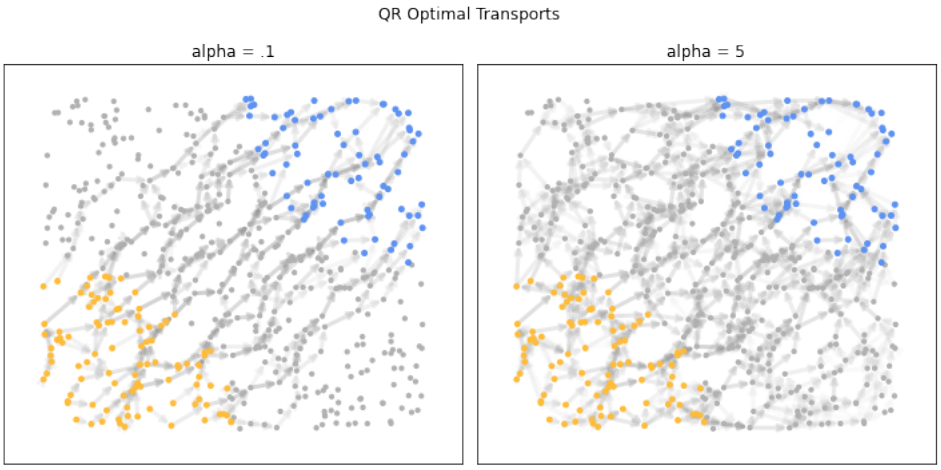

Figure 6: Comparing optimal transports for a low and high α value

However, when applying the same objective function on a graph with fewer edges (like the Partition graph), $\sum_i w_i l_i$ stays roughly the same while $\sum_i w_i^2$ increases. Intuitively, this is because any sufficiently connected graph will transport the same mass over approximately the same distance, albeit on different edges; however, since the mass is aggregated onto fewer edges, the quadratic regularizing factor will increase to reflect this density. In particular, if the Partition Algorithm reduces the number of edges by a factor of c and we assume naively that the weights on the vanished edges are evenly distributed onto the remaining edges, one can show that the regularizer then grows by a factor of c. A proper estimation of Wasserstein distance using the Partition Algorithm, thus requires a down-scaling of α. We calculate the magnitude of this scaling empirically.

3.2.3 Simulation

Our simulations consist of first calculating total weight using a kNN-connected graph and a fixed α, tracking Total QR Wasserstein Distance along with the Linear Term and the Quadratic Regularizer that make up the total weight. We then collect the same data on a Partition-connected graph, except with α scaled to various levels. Our goal is to find a c such that the Regularizer quantities after kNN-connecting with α and after Partition-connecting with $c\alpha$ are approximately equal.

So far, our QR Wasserstein distance for our graphs depends on 6 main parameters: k, K, $|V|$, α, point distribution, and mass distribution. Thus, we simulate the following scenarios:

- $k = 2, 3, 4, 6$
- $K = 12, 8$
- $|V| = 300, 150$
- initial $\alpha = 1, 4, .2$
- Uniformly-distributed points with disjoint masses vs. Beta-distributed points with overlapping masses

Throughout the simulations, untested parameters were held constant at default values of: $k = 4$, $K = 12$, $|V| = 300$, $\alpha = 1$, and uniform points with disjoint masses.

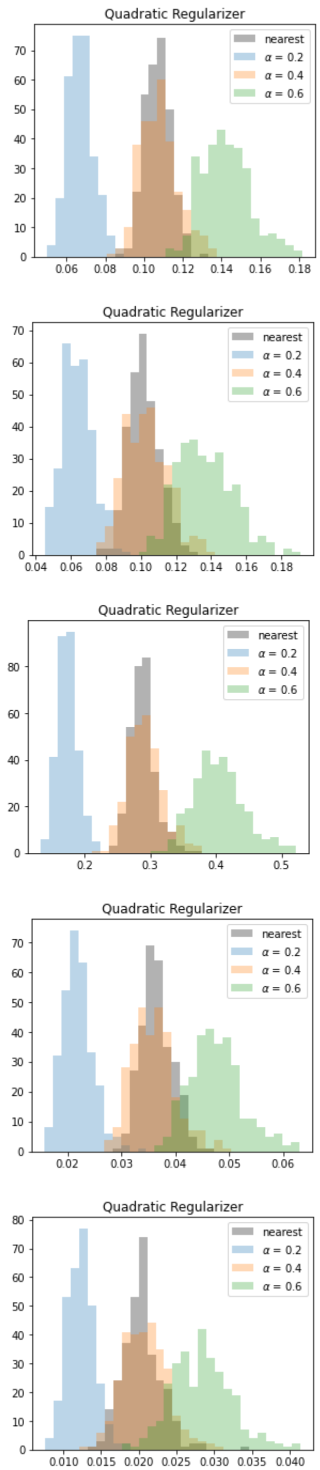

Figure 7: Simulation results from changing number of nodes, initial α, and point distribution

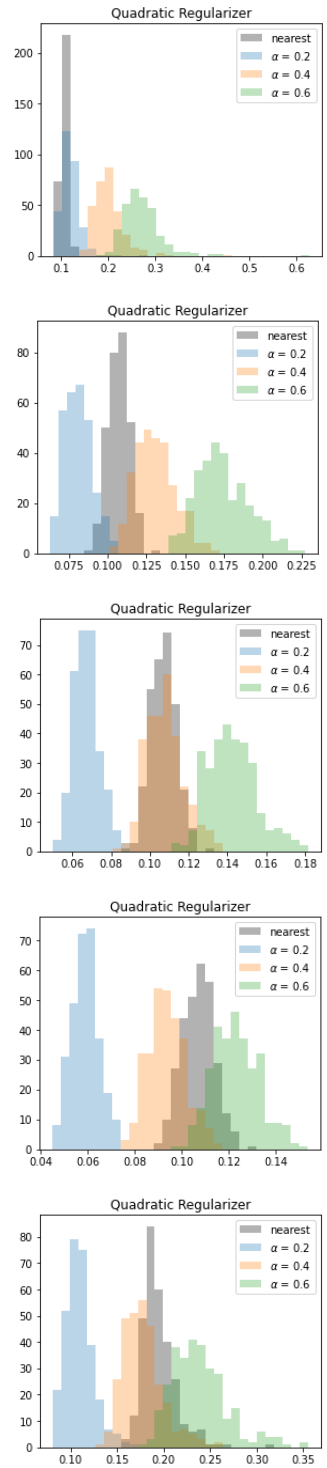

Figure 8: Simulation results from changing k and K (Note the shift in the grey histograms)

We focus first on the latter 3 scenarios that tested for variability in node count, initial α, and point/mass structure. Results are shown in Figure 7 above.

Surprisingly, none of these changes seem to influence α-scaling. We see in the third column of histograms that throughout all the cases, a scaling of 0.4α maintained the regularizing quantities to be around the same for both the kNN QR Distance and the Partition QR Distance.

We now turn to 2 remaining parameters: K and k. Recall that within the Partition graph, K represents the neighborhood size that each node could connect to while k represents the number of random connections each node actually makes. When these values were changed, Figure 8 shows that we indeed see a quite dramatic change in the α-scaling, where c seems to shrink along with the ratio of k/K. We can estimate that $c \approx .15$ when $k = 2, K = 12$ and $c \approx .5$ when $k = 4, 8$ and when $k = 6, 12$ but further research must be conducted to find the true relationship.

4 The "Football" Algorithm

We take an aside from estimation via a Partition Algorithm to pursue another algorithmic approach to estimating QR Wasserstein distance, employing a geometric result rather than strict computation.

4.1 Visual Intuition

The objective function of our optimal transport problem is $|\mathbf{w}| \cdot \mathbf{l} + \frac{\alpha}{2}\mathbf{w} \cdot \mathbf{w}$, where $|\mathbf{w}|$ is the component-wise absolute value of $\mathbf{w}$. This is necessary since we allow for negative entries in our construction. If we apply the same absolute value to our regularization term and set the variable x as the output to be minimized, we can "complete the square" to frame the question geometrically:

$$\begin{aligned} \frac{\alpha}{2}|\mathbf{w}| \cdot |\mathbf{w}| + |\mathbf{w}| \cdot \mathbf{l} &= x \\ |\mathbf{w}| \cdot |\mathbf{w}| + \frac{2}{\alpha}|\mathbf{w}| \cdot \mathbf{l} + \frac{1}{\alpha^2}\mathbf{l} \cdot \mathbf{l} &= \frac{2x}{\alpha} + \frac{1}{\alpha^2}\mathbf{l} \cdot \mathbf{l} \\ \left(|\mathbf{w}| + \frac{\mathbf{l}}{\alpha}\right) \cdot \left(|\mathbf{w}| + \frac{\mathbf{l}}{\alpha}\right) &= \frac{2x}{\alpha} + \frac{\|\mathbf{l}\|^2}{\alpha^2} \\ \left\| |\mathbf{w}| + \frac{\mathbf{l}}{\alpha} \right\| &= \sqrt{\frac{2x}{\alpha} + \frac{\|\mathbf{l}\|^2}{\alpha^2}} \end{aligned} \tag{1}$$

Letting $r = \sqrt{\frac{2x}{\alpha} + \frac{\|\mathbf{l}\|^2}{\alpha^2}}$, we see that r grows monotonically with x, so we can minimize x by minimizing r.

The problem is now reminiscent of a Ridge Regression problem, where we search for a "first intersection" on ball of radius r, centered at $-\frac{\mathbf{l}}{\alpha}$. However, our problem differs quite significant in its use of the component-wise absolute value. Visually, this alters our search zone into less of a ball around $-\frac{\mathbf{l}}{\alpha}$ and more of a "football" around the origin; the ball is restricted to the positive quadrant and then reflected across all axes. Figure 9 demonstrates this in 2 dimensions.

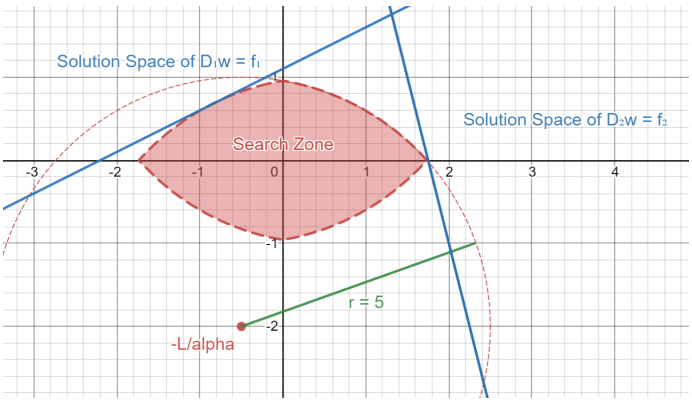

Figure 9: Visualization of a 2D search zone intersecting 2 solution spaces

We formalize this idea by defining the set of valid radii

$$R := \left\{ r \in \mathbb{R} : \text{ the system } \begin{bmatrix} \left\| |\mathbf{w}| + \frac{1}{\alpha} \right\| & = r \\ Dw & = f \end{bmatrix}_w \text{ has a solution} \right\}.$$

and define $r_{min} = \inf R$. We claim (and prove in section 4.4) that $r_{min} \in R$, $r = r_{min}$ elicits a unique solution $\mathbf{w}_{min}$ to the system and that $\mathbf{w}_{min}$ is the weight vector for a QR Optimal Transport on the graph described by D.

Functionally, this creates an interesting halfway-point between the familiar simplex of LASSO Regression and sphere of Ridge Regression. The solving process thus exhibit traits from both methods: the sparsity granted by LASSO and the local differentiability of ridge regression. This development may be motivation enough for further investigation (e.g. in feature selection, regularization, etc.).

4.2 Algorithm

We now attempt to find the smallest r such that a $\mathbf{w} \in \mathbb{R}^{|E|}$ satisfies both $\left\| |\mathbf{w}| + \frac{1}{\alpha} \right\| = r$ and $D\mathbf{w} = \mathbf{f}$, i.e. the smallest r that allows for an intersection between the football and the solution space, denoted by the affine subspace S. We attempt to do this by:

1. Identifying in which quadrant the non-zero part of the solution lies
2. Identifying the dimensions along which the solution is 0
3. Projecting an adjusted $-\frac{1}{\alpha}$ onto the solution space, accounting for the zeros in the solution

Step 1 is done by considering $\text{proj}_S \mathbf{0}$. We claim that if $\text{proj}_S \mathbf{0}$ lies in a closed quadrant of $\mathbb{R}^{|E|}$, then so too does $\mathbf{w}$. (This also turned out to be false, but it's too late for me to rewrite everything. The algorithm still works as an approximation.) This is because we can find an r such that $\text{proj}_S \mathbf{0}$ is a solution to $\left\| |\mathbf{w}| + \frac{1}{\alpha} \right\| = r$ and $D\mathbf{w} = \mathbf{f}$. If r is minimal, we have found the optimal $\mathbf{w}$ and we are done. If r is not minimal, we can shrink r and still find solutions to the system. However, it can be shown that solutions converge to a point within the closed quadrant as r approaches its minimal value.

Step 2 can be attempted by naively extending the 2-dimensional intuition from Figure 9 to $|E|$ dimensions. We compare $\text{proj}_S \mathbf{0}$ with $\text{proj}_S \frac{1}{\alpha}$ and assert (incorrectly) that $\mathbf{w}_e = 0$ whenever the e-th

component of $\text{proj}_S \mathbf{0}$ and $\text{proj}_S \frac{\mathbf{l}}{\alpha}$ are opposite in sign. For a counterexample, see Figure 10. Despite the limitations of this claim, simulated results show that this method still produces a decent approximation for $\mathbf{w}$.

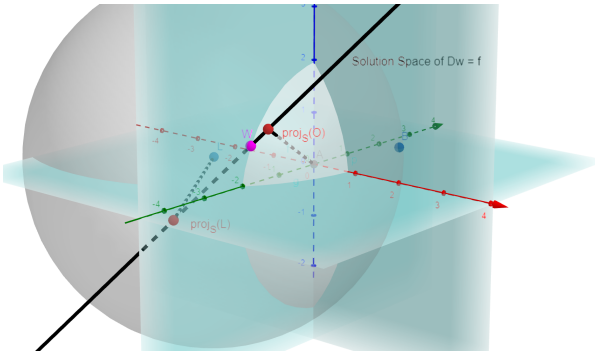

Figure 10: Counter-example to the projection-method in higher dimensions. Note that $\text{proj}_S(\mathbf{O})$ and $\text{proj}_S(\mathbf{L})$ have z-values with opposite signs, yet the z-value of $\mathbf{w}$ is not zero.

In Step 3, since we know the quadrant in which $\mathbf{w}$ lies, we project a properly reflected $-\frac{\mathbf{l}}{\alpha}$ onto S, while requiring the components found in Step 2 to be 0. The reflection is necessary because if $\mathbf{w}$ is in a determined quadrant, it's necessary that $-\frac{\mathbf{l}}{\alpha}$ lies in the opposite quadrant to be properly projected.

We thus create the following Algorithm:

Algorithm 1: Football Algorithm

Result: Weight vector for QR Optimal Transport
l = lengths vector;
D = incidence matrix;
f = sink - source;

origin_on_S = projection of $\vec{0}$ onto S;
new_l = l/α adjusted in signs to be opposite of origin_on_S;

has_sign_change = sign(origin_on_S) != sign(new_l);

sliced_id = rows of the $|E| \times |E|$ identity matrix where has_sign_change is True;
D_aug = D augmented with sliced_id; *# this forces the solution to have zeros*
f_aug = f augmented with zeros ;

estimated_w = projection of new_l onto the solution space of D_aug @ w = f_aug;

return estimated_w

This algorithm can only be an approximation for $\mathbf{w}$ because the method used to find its 0-entries is not completely accurate in higher dimensions. As such, this algorithm is most easily improved by a more accurate identification of 0-components. Until a better method is found, the current algorithm's quick calculation for an approximate solution has the potential for speeding up iterative QP solvers that benefit from a close initialization. Figure 11 demonstrates the the approximation by overlaying the first 300 weights of the the estimated weights on top of the first 300 true weights. We also analyze the algorithm's performance through simulation in the following section.

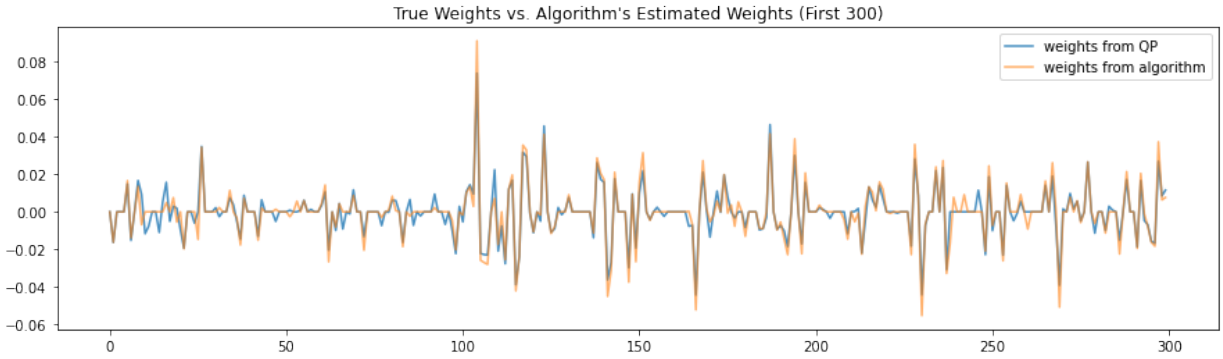

Figure 11: Histograms comparing true Wasserstein distance (blue) and its estimation via the Football Algorithm (orange)

4.3 Simulated Results

We compare results by finding the optimal weight vector using both the traditional quadratic programming approach and the new algorithm. On a random point cloud, we create a connected graph with kNN (k = 12) and use disjoint source/sink distributions. Figure 12 summarizes our results. We see that, by virtue of being only an approximation, the algorithm's weights are sub-optimal by a factor of around 1.2. If necessary, this could be systematically scaled up to estimate the true QR Wasserstein distance, but the main power of this method comes from closely approximating the optimal weight vector.

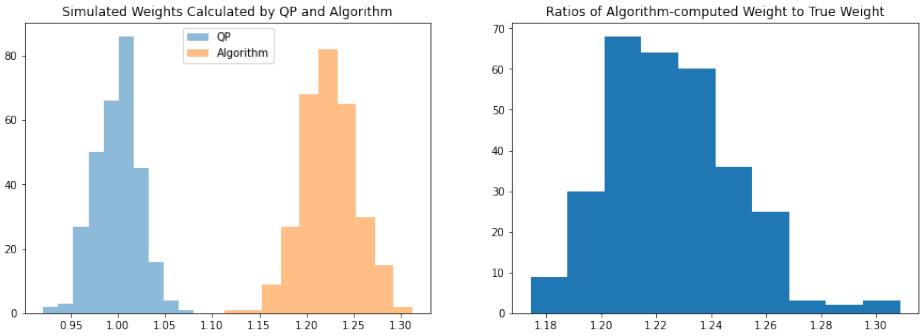

Figure 12: Histograms comparing true Wasserstein distance (blue) and its estimation via the Football Algorithm (orange)

Even more impressive is the computation time for getting the estimate. Requiring only three least-squares calculations on very sparse matrices, the algorithm performs 43 times faster than the QP approach, on average. The histogram for computation time is presented in Figure 13 below.

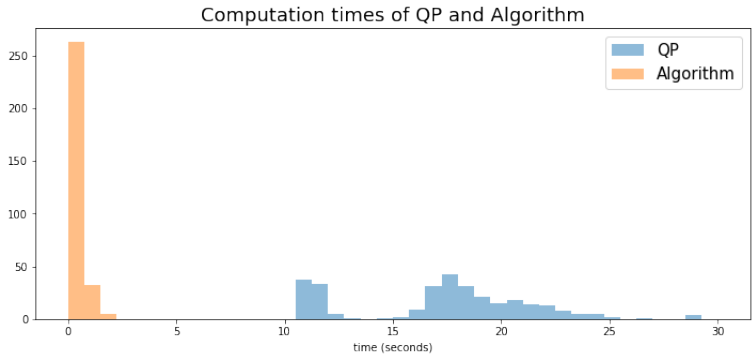

Figure 13: Histogram comparing original optimal transport run-time (blue) and run-time of Football Algorithm (orange)

4.4 Theorems

We prove the aforementioned claims here.

Lemma 4.1. *Let* $r_{min} := \inf R$. *Then* $r_{min} \in R$.

Proof. Construct a monotone decreasing sequence $\{r_i\}$ in R such that $\{r_i\} \to r_{min}$. By definition of R, for each r_i we have a $\mathbf{w}_i \in \mathbb{R}^{|E|}$ such that

$$\left\| |\mathbf{w}_i| + \frac{\mathbf{1}}{\alpha} \right\| = r_i \quad \text{and} \quad D\mathbf{w}_i = f$$

However, if W is the set of all possible $\mathbf{w}_i$, we have that W is bounded by the first constraint since $\{r_i\}$ is decreasing. Also, defining $\varphi(w) := \left\| |\mathbf{w}| + \frac{\mathbf{1}}{\alpha} \right\|$, see that

$$W = \left(\bigcap_{i \in \mathbb{N}} \phi^{-1}(\{r_i\}) \right) \cap D^{-1}(\{f\})$$

and thus W is closed by being an intersection of closed preimages (both ϕ and D are continuous). Conclude that W is compact, and define the convergent subsequence $\{\mathbf{w}_{i_j}\} \to \mathbf{w}_{min}$.

But now we have

$$r_{min} = \lim r_{i_j} = \lim \phi(\mathbf{w}_{i_j}) = \phi(\lim \mathbf{w}_{i_j}) = \phi(\mathbf{w}_{min})$$

using continuity of ϕ. Conclude that $r_{min} \in R$ because $\mathbf{w}_{min}$ is a solution. □

Lemma 4.2. *Setting* $r = r_{min}$ *produces a single solution to the system.*

Proof. Suppose $\mathbf{w}_1$ and $\mathbf{w}_2$ are two distinct solutions to the system $\begin{bmatrix} \left\| |\mathbf{w}| + \frac{\mathbf{1}}{\alpha} \right\| & = r_{min} \\ D\mathbf{w} & = f \end{bmatrix}_{\mathbf{w}}$. Then define $\mathbf{w}_m = \frac{\mathbf{w}_1 + \mathbf{w}_2}{2}$. See that we indeed have $Dw_m = \frac{1}{2}(Dw_1 + Dw_2) = \frac{1}{2}(f + f) = f$.

Next, prove an intermediate result: that $\left\| \frac{\mathbf{w}_1 + \mathbf{w}_2}{2} \right\|^2 \leq \frac{\|\mathbf{w}_1\|^2 + \|\mathbf{w}_2\|^2}{2}$. By Triangle Inequality, we know $\left\| \frac{\mathbf{w}_1 + \mathbf{w}_2}{2} \right\| \leq \frac{\|\mathbf{w}_1\| + \|\mathbf{w}_2\|}{2}$, so we have that

$$\left\| \frac{\mathbf{w}_1 + \mathbf{w}_2}{2} \right\|^2 \leq \frac{\|\mathbf{w}_1\|^2 + 2\|\mathbf{w}_1\|\|\mathbf{w}_2\| + \|\mathbf{w}_2\|^2}{4}$$

It now suffices to show that $2\|\mathbf{w}_1\|\|\mathbf{w}_2\| \leq \|\mathbf{w}_1\|^2 + \|\mathbf{w}_2\|^2$ but this is true since

$$0 \leq (\|\mathbf{w}_1\| - \|\mathbf{w}_2\|)^2 = \|\mathbf{w}_1\|^2 - 2\|\mathbf{w}_1\|\|\mathbf{w}_2\| + \|\mathbf{w}_2\|^2$$

After noting that $\mathbf{w}_1$ and $\mathbf{w}_2$ are distinct, and thus cannot be parallel, we write

$$\begin{aligned}
\left\| |\mathbf{w}_m| + \frac{\mathbf{l}}{\alpha} \right\|^2 &= \sum_i \left(\left| \frac{\mathbf{w}_{1i} + \mathbf{w}_{2i}}{2} \right| + \frac{\mathbf{l}_i}{\alpha} \right)^2 \\
&= \sum_i \left| \frac{\mathbf{w}_{1i} + \mathbf{w}_{2i}}{2} \right|^2 + \sum_i \frac{\mathbf{l}_i}{\alpha} |\mathbf{w}_{1i} + \mathbf{w}_{2i}| + \sum_i \frac{\mathbf{l}_i^2}{\alpha^2} \\
&< \left\| \frac{\mathbf{w}_1 + \mathbf{w}_2}{2} \right\|^2 + \sum_i \frac{\mathbf{l}_i}{\alpha} |\mathbf{w}_{1i}| + \sum_i \frac{\mathbf{l}_i}{\alpha} |\mathbf{w}_{2i}| + \sum_i \frac{\mathbf{l}_i^2}{\alpha^2} \\
&\leq \frac{\|\mathbf{w}_1\|^2 + \|\mathbf{w}_2\|^2}{2} + \sum_i \frac{\mathbf{l}_i}{\alpha} |\mathbf{w}_{1i}| + \sum_i \frac{\mathbf{l}_i}{\alpha} |\mathbf{w}_{2i}| + \sum_i \frac{\mathbf{l}_i^2}{\alpha^2} \\
&= \sum_i \left(\frac{\mathbf{w}_{1i}^2}{2} + \frac{\mathbf{l}_i}{\alpha} |\mathbf{w}_{1i}| + \frac{\mathbf{l}_i^2}{2\alpha^2} \right) + \sum_i \left(\frac{\mathbf{w}_{2i}^2}{2} + \frac{\mathbf{l}_i}{\alpha} |\mathbf{w}_{2i}| + \frac{\mathbf{l}_i^2}{2\alpha^2} \right) \\
&= \frac{1}{2} \sum_i \left(|\mathbf{w}_{1i}| + \frac{\mathbf{l}_i}{\alpha} \right)^2 + \frac{1}{2} \sum_i \left(|\mathbf{w}_{2i}| + \frac{\mathbf{l}_i}{\alpha} \right)^2 \\
&= \frac{1}{2} \left\| |\mathbf{w}_1| + \frac{\mathbf{l}}{\alpha} \right\|^2 + \frac{1}{2} \left\| |\mathbf{w}_2| + \frac{\mathbf{l}}{\alpha} \right\|^2 \\
&= r_{min}
\end{aligned}$$

But since we also have $D\mathbf{w}_m = f$, this contradicts the minimality of r_{min}. Conclude that the system can have at most one solution. □

Lemma 4.3. $\mathbf{w}_{min}$ *solves the Optimal Transport problem*

Proof. See the complete-the-square derivation (1) detailed in Section 4.1. It remains to be shown that $f(x) = \sqrt{\frac{2x}{\alpha} + \frac{\|\mathbf{l}\|^2}{\alpha^2}}$ is monotone increasing for positive x but this is evident by composition. □

5 Conclusion

The methods detailed above allow for efficient approximations of Quadratically-regularized Optimal Transport. With enough innovations into calculation speed, it is possible that the QR-variant could become a regularly used form of Optimal Transport, with notable advantages (sparsity, differentiability, convexity, etc.). There is much yet to be researched in both methods, namely a relationship between $\frac{k}{K}$ and the optimal α-scaling for the Partition Algorithm, and a deterministic way of finding zero-components in the Football Algorithm (as well as a better name). The insights gained by the Football approach are not limited to Optimal Transport either; rather, it seems they apply to any Quadratic Programming problem with only equality constraints. Toward the end of the project, significant effort was spent trying to connect the OT solutions' to a modified least-squares solution, but was eventually proven to be false.

5.1 Acknowledgements

This project would not have been possible without the support, guidance, and research topic provided by Professor Alex Cloninger. He made the long nights of scribbling and coding worth the excitement of sharing my results the next day. My thanks extends to the UCSD Math Department as well, for allowing me to develop my passion for math research through such an insightful Honors Program.

References

[1] Montacer Essid, Justin Solomon. *Quadratically-Regularized Optimal Transport on Graphs.* arXiv, 1704.08200, `https://arxiv.org/abs/1704.08200`, 2018.

[2] George C. Linderman, Gal Mishne, Yuval Kluger, Stefan Steinerberger. *Randomized Near Neighbor Graphs, Giant Components, and Applications in Data Science.* arXiv, 1711.04712, `https://arxiv.org/abs/1711.04712`, 2017.