Weighted Adaptive Optimal Transport Using Quasi Newton Methods

Vaish Gajaraj *[1], Esteban G. Tabak[2], and Debra Laefer[3]

[1,2]Courant Institute of Mathematical Sciences, NYU
[3]Center of Urban Science and Progress, NYU

Abstract

Given two collections of points, x_i and y_j, weighted for intensity, an adaptive, adversarial methodology is developed for the optimal transport of one set of points to the other set of points. Such a map provides a pairing from the initial set of points to the final set of points, a necessary precondition for measuring local change between sets of points. We consider each set of independent points as discrete samplings of probability distributions, and seek to find an optimal map $T(x)$ from the initial distribution onto the final distribution. In our formulation, such an optimal map method minimizes a transportation cost and the relative entropy between $T(x_i)$ and y_j. By reformulating relative entropy into a mini-max problem, the methodology does not require any predetermined knowledge of the data. We split our initial and final set of points into several intermediate distributions, and solve for the optimal transport map between consecutive, intermediate distributions. Composing these simple maps constructs a transportation between dissimilar, complex distributions. This procedure is iterated to guarantee the global solution is optimal, as given by McCann's Interpolant Theorem. The procedure is illustrated through examples in two dimensions.

1 Motivation

Our problem originates from civil engineering, specifically infrastructure management. There is a need for a universal change detection algorithm to identify differences between two scanned images, acquired through LiDAR scanning or other conventional photo-scanning technology. Such an algorithm would be useful in finding changes between two images of city landscapes, and how those images evolved over time. The goal of this project was to take an adaptive optimal transport approach and analyze its effectiveness at pushing forward an

*Project completed with the guidance and support of Courant's AM- SURE program

initial image into a target image, and thereby create an optimal pairing from the original map to the final map. Adaptive optimal transport seeks to build an optimal map from the initial distribution of points, x_i , to a final distributions of points, y_j. An optimal map is defined as the map that minimizes some cost function between the two distributions. Our cost function is a reformulated version of relative entropy. The final optimal map will give information on how the pixels from the original picture, x_i moved in relationship to the pixels of the final picture, y_j. By understanding how these initial pixels have moved, we can identify areas of change automatically and on large scales.

2 Introduction to Optimal Transport

2.1 Monge Problem

The origin of optimal transport comes from Gasparde Monge's desire to transport physical materials between two sites at a minimal cost. Given a source distribution $\rho(x)$ and a target distribution $\mu(x)$, the optimal transport map, $T(x)$, is the one that minimizes the expected cost of transportation. Such a map will "push forward" the distribution in μ to ρ using T. To push forward ρ using T onto μ (denoted as $T_{\#}\rho = \mu$) means the distribution ρ will fall into the same probability distribution as μ. In the case of pictures, it would mean that the original picture maps as best it can to the final picture. The general form of this problem is as follows:

$$\min_{T} \left\{ \int c(x, T(x))\mu(x)\, dx \;\middle|\; T_{\#}\rho = \mu \right\}$$

The reformulations of the problem by Kantorovich nearly two centuries later, generalized the formulation to the case for when two measures ρ and μ have discrete support, and allowed for this problem to be solved through linear programming.

Kantorovich's formulation helped find optimal pairings between supply and demand. In order to provide a more flexible framework for data science applications, sample-based techniques to solve the OT problem were developed in [4, 5, 6]. A central question to address when posing sample-based OT problems is the meaning of the push-forward condition $T_{\#}\rho = \mu$ when ρ and μ are only known through samples x_i and y_j respectively. This is the case in problems dealing with pictures or LiDAR samplings. In the formulations from [4, 5, 6], the problem of optimal transport was first reformulated into an adversarial problem, computing the push forward between pre-determined sets of features over the two sample sets. In the example of an image, a certain feature corresponds to a particular attribute of the original image (e.g. shadows in an image). The problem was then further relaxed into an equality of empirical means. This approach however raised the trouble of feature selection, and requires knowledge of what features to look for.

The method described in [Essid et al., 2018] incorporates feature selection into the formulation of the optimal transport problem. This process is limited

to situations of a fixed number of points. We redid this approach considering the case when certain points carry intrinsic information with it (weights). Not only does this approach not require any prior knowledge of the data, but it also moves points in a less naive way, that is, the optimal transport is informed by not only the location of the final points but also their weights. This technique has applications in change detection for images, and other data sets with weighted pixels.

3 Adaptive Optimal Transport

In this formulation of optimal transport, we seek to both push forward the points x_i to y_j, such that some $T(x_i)$ and y_j are from the same distribution. We measure the similarity of distributions through their relative entropy. In addition, the problem seeks to minimize the cost in doing such a push forward. The cost defined in this problem is the distance squared cost, or the Wasserstein distance.

In our approach, we use a quadratic cost function:$\phi(x, y) = \|x - y\|^2$.

Problem 1. *Find the map* $T : \mathbf{R}^d \to \mathbf{R}^d$ *such that*

$$T(x) = \min_f \left\{ \int \|x - f(x)\|^2 \mu(x)\, dx \;\middle|\; f_{\#}\rho = \mu \right\}$$

As stated in Brenier's Theorem [Villani, 2003], such a map f does exist, and for the quadratic cost, the optimal solution is necessarily the gradient of some convex function $\phi(x), y = T(x) = \nabla\phi(x)$. Our problem transforms therefore into this slightly modified form:

Problem 2. *Find the map* $T : \mathbf{R}^d \to \mathbf{R}^d$ *such that*

$$T(x) = \min_\phi \left\{ \int \|x - \nabla\phi(x)\|^2 \mu(x)\, dx \;\middle|\; \nabla\phi_{\#}\rho = \mu \right\}$$

Problem 3. *Find the convex function whose gradient will correctly satisfy the push forward condition.*

$$\nabla\phi_{\#}\rho = \mu \tag{1}$$

If we can find such a ϕ that allows us to move the distribution ρ to be as close to μ as possible, we effectively make the cost function $T(x)$ zero. This is not a well defined question in its nature, especially when considering dimensions higher than one [Galichon, 2018]. As iterated in [Villani, 2003], however, a unique solution to problem 2 of is some convex function. Finding this function ϕ is where the extension to relative entropy, or Kullback-Leibler Divergence, comes in.

3.1 Kullback-Leibler Divergence

Definition 3.1. *For any two given probability distributions ν and δ, the Kullback-Liebler Divergence, denoted $D_{KL}(\nu\|\delta)$ to be*

$$D_{KL}(\nu\|\delta) := \int \ln\left(\frac{d\nu}{d\delta}\right) d\rho$$

Intuitively, KL Divergence provides a quantitative representation of how similar two probability distributions are. A KL-Divergence that is zero means that the two distributions that are identical. The goal of the push forward condition is to make $T(x_i)$ and y_j as similar to each other as possible, therefore the problem can be thought of as a minimization problem for KL-Divergence.

As previously stated, this minimizer exists for the Wasserstein distance, as given by Brenier's theorem. However, the same function that minimizes Wasserstein distances also provides a unique minimizer for the KL Divergence.

Theorem 3.1. *φ is a solution to problem 3 if and only if φ and satisfies*

$$\varphi = \arg\min_{\phi} \{D_{KL}(\nabla\phi_{\#}\rho\|\mu) \mid \psi \text{ is convex}\} \tag{2}$$

A proof of this can be found in [Essid et al., 2018].

If we accomplish push forward as best as possible, this φ would be ~ 0. This concludes the minimization portion of our formulation, where we seek to minimize over the continuous function space ϕ. Included in the adversarial formulation is the maximization problem. As proved in [3], the Kullback-Liebler divergence can be reformulated as a variational problem.

Theorem 3.2.

$$D_{KL}(\nu\|\mu) = 1 + \max_{g}\left\{\int g(z)d\nu(z) - \int e^{g(y)}d\mu(y) \;\middle|\; g \text{ borel measurable}\right\} \tag{3}$$

Notice here we introduce a new variable ν to represent our variational formulation. It is valuable to note that the variational reformulation of the Kullback-Leibler divergence is a consequence of the convexity of of $x \to -log(x)$. When taking the Legendre-Fenchel transform twice for the defined KL-Divergence, this variational form is realized.

Notice that the convexity of the relative entropy gives an avenue to include some convex ϕ to solve 2. This will come up shortly.

Given two random variables Z distributed through ν and Y distributed as our original μ with $\nu << \mu$ we can equivalently express the formula in 3.2 through probabilistic expectation values:

$$D_{\mathrm{KL}}(\nu\|\mu) = 1 + \max_{g}\left\{\mathbb{E}[g(Z)] - \mathbb{E}\left[e^{g(Y)}\right] \;\middle|\; g \text{ borel measuable}\right\}$$

We can take this one step further. If we define our ν variable as a result of the push forward condition as given by 1, it follows that $\nu \sim \nabla\phi_{\#}\rho$. Therefore, we can parameterize Z through the push forward condition. Given some X distributed through ρ, $Z \sim \nabla\phi_{\#}(X)$. As such, we have introduced the push forward constraint into our problem! The new form is as follows:

Theorem 3.3.

$$D_{KL}(\nabla\phi_{\#}\rho\|\mu) = 1 + \max_{g}\left\{\mathbb{E}[g(\nabla\phi(X))] - \mathbb{E}\left[e^{g(Y)}\right] \;\middle|\; g \textit{ borel measurable}\right\}$$

By adding our push forward condition, we have also solved the issue of finding a convex ϕ for which the original cost function would be satisfied. Previously, we stated that minimizing the KL Divergence is necessarily how one might solve the cost minimization problem. Now, we have found a way to include that ϕ into the construction of KL divergence, a problem previously disconcerting.

In this variational formulation however, our maximization function g will remain convex throughout. The first part of the equation containing $g(\nabla\phi(X))$ is essentially linear while the second part, $e^{g(Y)}$ is a negative convex function. As a result, this guarantees that at least a part of this Lagrangian is convex. This is not necessarily the case for ϕ. In the following section, we will discuss how such a convexity is enforced in order to solve the optimization problem.

With our original problem in mind, we can assume then that the random variables X and Y are given independent samples $x_1...x_i$ and $y_1...y_j$. From there we postulate the following approximation of both the samples and their expectations to sums:

$$D_{\mathrm{KL}}(\nabla\phi_{\#}\rho\|\mu) \approx 1 + \max_{g}\left\{\frac{1}{n}\sum_{i} g(\nabla\psi(x_i)) - \frac{1}{m}\sum_{j} e^{g(y_j)}\right\} \tag{4}$$

In the formulation in this paper, we take this sample based formulation further, by considering each sample's weight, represented by w_i or w_j respectively:

$$D_{\mathrm{KL}}(\nabla\phi_{\#}\rho\|\mu) \approx 1 + \max_{g}\left\{\sum_{i} w_i^x g(\nabla\psi(x_i)) - \sum_{j} w_j^y e^{g(y_j)}\right\} \tag{5}$$

These weights were normalized such that $\sum_i w_i^x = 1$ and $\sum_j w_j^y = 1$.

When the minimizing constraint from 3.1 is composed with the maximizing constraint we arrive at the following optimization equation:

$$\min_{\phi}\max_{g} L[\phi, g] \approx \arg\min_{\phi}\left\{\max_{g}\left\{\sum_{i} w_i^x g(\nabla\phi(x_i)) - \sum_{j} w_j^y e^{g(y_j)}\right\}\right\} \tag{6}$$

Where L is our Lagrangian function that defines the optimization problem. This will be discussed in the following section.

3.2 A Further Look at Mini-Max

The minimax problem is defined by two functions $\nabla\phi$ and g. These two functions should move as follows:

- $\nabla\phi(x_i)$ This family of functions essentially qualifies the push forward condition, pushing forward the points x_i to where the relative entropy is smallest compared to y_j.
- $g(y_j)$ An intuitive understanding of this function is one that highlights areas of differences. Given a function ϕ, g will try and maximize the Kullbach-Libeler Divergence. By the action of simultaneously minimizing and maximizing, it is possible to find a saddle point solution.

The optimization problem is solved when g becomes constant (~ 0) on the support of the distributions. At that point, ϕ does not need to push forward the points x_i anymore, and it receives no new highlighting from g.

4 Discretizing the Problem

In [Essid et al., 2018], the authors offer a methodology to tackling this problem in a discrete setting. The framework we use to solve our problem is through first making 9 into a standard numerical optimization problem, and then solving this equation for many local maps created by interpolating the two data sets. This entire process is iterated again and again until the original data points move linearly to their final location.

4.1 Solving Several Local Maps

Transporting from one highly complex set of points to another very different set of points is infeasible given our current equation. We would need to solve our equation between sets of points that are nearby to one another. In order to accomplish this, we interpolate the first and the last set of points together. For N collections of points between the initial set of points x_i and the final target set y_j:

$$z_k = \frac{N-k}{N}x_i + \frac{k}{N}y_j \tag{7}$$

This comes with a problem however, in that we are arbitrarily interpolating these data sets together to achieve several intermediary sets of points from which we can apply our reformulation of KL Divergence. To address this, we use a methodology described in [Kuang and Tabak, 2017]. This paper describes a closed form optimal transport between two Gaussian distributions. Obviously, highly complex data is rarely Gaussian, but doing this for our data would allow

for a better guess when computing an optimal transport between two distributions. We treat both our initial data and our final data as Gaussian functions being optimally transported onto one another as specified in more depth in [Kuang and Tabak, 2017]. From solving the optimal transport problem we gain an image of the original data set x_i onto y_j and conversely. We name these images T_x and T_y. The takeaway is that we are no longer simply taking portions of x_i and y_j in 7, but rather we are interpolating points between mixtures of x_i with T_x and y_j with T_y.

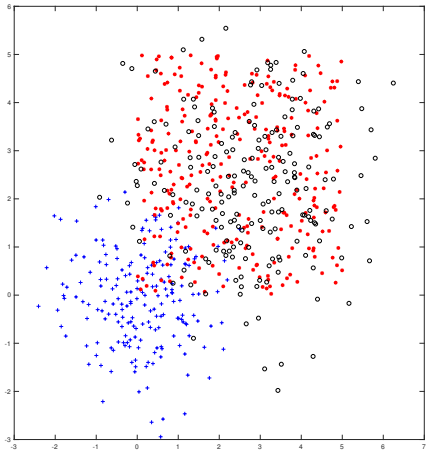

Figure 1: An example of the solved optimal transport between Gaussian. Here x_i is blue, T_x is black and the target y_j is red. y_j is clearly not Gaussian but regardless in our first step we attempt to map as if they were. We attain disappointing results as a result.

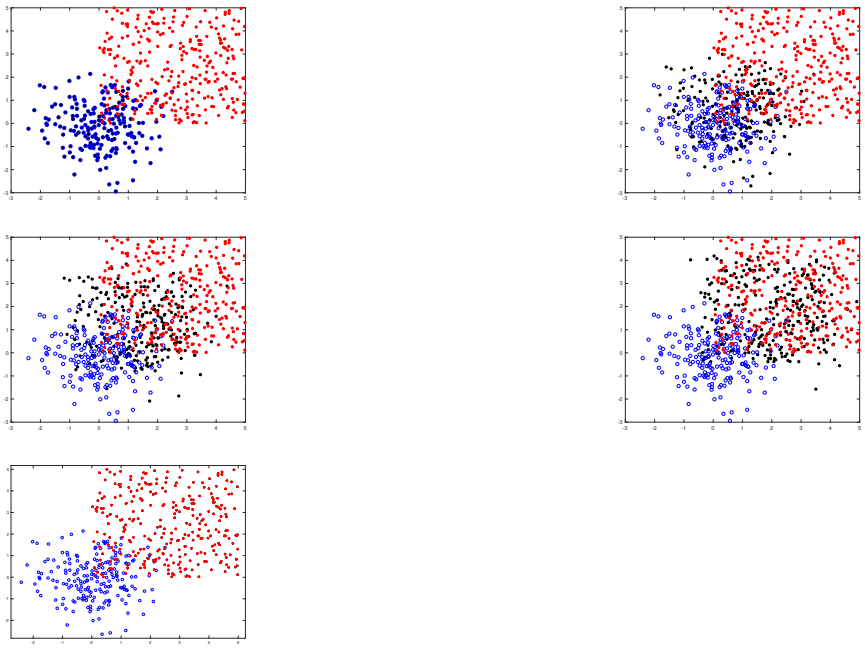

Figure 2: An example of equation 7 in use where N = 4. There are four z_k maps being constructed between the original blue x_i data and the final y_j data. The data being interpolated is a mixture of T_x and x_i and y_j with T_y, which provides us intermediary clouds that are simpler to solve.

Our goal is to transport the x_i coordinates to the y_j coordinates through the use of these intermediary sets of points z_k. Each iteration of the optimal transport algorithm will push forward our blue points step by step, saving them in positions that gradually move towards the red coordinates.

$$T_{composed} = T_N \circ T_{N-1} \circ ... \circ T_1 \tag{8}$$

Once we push forwards from our initial set of points to the final mapping, we have accomplished a rough estimation of how the original points have evolved from the first image to the last image. In our applications to city maps, this would equivalently be an estimation of how the original image has changed to the final image.

4.2 Parameterizing the Lagrangian

The transported map of initial samples corresponds to the value of $\nabla\phi(x)$. In our formulation of this problem, we assigned normalized weights to each point

rather than dividing though by the number of points for each distribution:

$$\min_{\phi} \max_{g} L[\phi, g] \approx \arg\min_{\phi} \left\{ \max_{g} \left\{ \sum_{i} w_i^x g(\nabla \phi(x_i)) - \sum_{j} w_j^y e^{g(y_j)} \right\} \right\} \quad (9)$$

Where w_i and w_j correspond to the weights of each of the points x_i and y_j. The reasoning for this approach was to find a more efficient way of mapping points of varying intensity, such as pixels in an image. For the case of black and white images, we would assign each point a certain intensity corresponding to how bright the pixel is. Such an approach would allow us to pair pixels based on both the location of each pixel and its grey scale.
While this formula is great in a theoretical sense, it is not possible to minimize or maximize under infinite function spaces such as ϕ or g. Therefore we elect to parameterize our Lagrangian function. From Brenier's theorem, one of the constraints of our problem is that it must be convex. Therefore we must select a function space that is convex when two distributions are nearby. Notice that because the sets of points we developed from interpolation are nearby one another, we can suppose that their optimal map is a perturbation of the identity. Therefore we suppose that the function ϕ is of the form:

$$\phi(x) = \frac{1}{2}||x||^2 \quad (10)$$

This allows us to have some reliability that we will land on a solution to the optimization problem after each iteration. Our function spaces for ϕ and g should be able to detecting and correct global displacements and scaling. In addition, we would like our functions to be able to handle non-linear transformations. To do all of this, we take the approach generalized by [Essid et al., 2018], where:

$$\phi(x) = \frac{1}{2} x^\intercal (I + A_0) x + a_1 \cdot x + \phi_{nl}(x) \quad (11)$$

and

$$g(z) = \frac{1}{2} z^\intercal (B_0) z + b_1 \cdot x + b_2 + g_{nl}(x) \quad (12)$$

where $A_0, B0$ are symmetric matrices in R^{dd}, a_1, b_1 are vectors in R^d , $b_2 \in R$ is a scalar, and ϕ_{nl} and g_{nl} stand for additional non-linear functions. These non linear functions serve the purpose of allowing our mappings to fit to more complex shapes. In our algorithm we chose isotropic Gaussian functions of the form

$$\phi_{nl} = \alpha exp(-\frac{v||x - \overline{x}||}{2}) \quad (13)$$

Where α is a vector in R^d and v is some scalar value acting on the means.
Parameterizing the Lagrangian with vectors $\alpha \in \mathbb{R}^a, \beta \in \mathbb{R}^b$ we set $\phi(x) = \phi_\alpha(x)$ and $g(y) = g_\beta(y)$. We seek to solve the minimax problem in $\alpha \in \mathbb{R}^a, \beta \in \mathbb{R}^b$ for

the Lagrangian:

$$\min_{\phi}\max_{g} L[\alpha,\beta] = \arg\min_{\alpha}\left\{\max_{\beta}\left\{\sum_{i} w_i^x g_\beta(\nabla\phi_\alpha(x_i)) - \sum_{j} w_j^y e^{g_\beta(y_j)}\right\}\right\} \tag{14}$$

This function space selection of ϕ and g adapt with each iteration, changing their coefficient values as per the evolving shape of the set of points. This is why we consider our optimal transport as an adaptive optimal transport.

5 Algorithm

In ([Essid et al., 2018]) a procedure is given to update the α and β terms using a second order gradient descent algorithm, where each new update is implicitly defined by the prior update. This method however is computationally cumbersome, therefore in our approach we use the low rank quasi-Newtonian update schema described in [Essid et al., 2019]. This method implicitly calculates gradients, however rather than computing the Hessian during each iteration, works out a rank-one approximation of the Hessian for each step. Using gradient descent, we update the α and β terms as follow:

$$\begin{pmatrix}\alpha_{n+1}\\ \beta_{n+1}\end{pmatrix} = \begin{pmatrix}\alpha_n\\ \beta_n\end{pmatrix} - \eta(B^*)G^n \tag{15}$$

Where $eta \to 0$ is the learning rate standard in gradient descent, B^* is the approximated Hessian described in [Essid et al., 2019] and G^n is the gradient of the Lagrangian given the current α's and β's.

5.1 Backwards step

Now we have a frame work to push forward our initial x_i points to a target y_j set of points. Simply pushing forward our points to a final location does not complete our problem however. Recall that we preform an optimal transport only on sets of nearby interpolated points. This does not equate to an optimal transport from the original set of points to the final set of points however. An ideal optimal transport would move each point linearly from the initial position to the final position. After the first map was pushed forward, any individual point would move in a more or less jagged, non-linear progression. Although the whole set of points are moving towards the final position, any single point can move backwards, or jagged from its final position. A process detailed in [Kuang and Tabak, 2017] provides a method of iterating through this global algorithm many times such that the mapping from the initial set of points to the final set is linear. It involves applying McCann's Interpolant theorem, where through iterating an optimal mapping process, the transformation moves the points linearly.
A pseudo-code of this process of interpolating, solving the minimax optimal

transport problem, and iterating through this process is as follows.

Algorithm 1 Optimal Transport between Points

▷ *Step 1: Initialize intermediate sets of points*
N ← number of intermediate sets of points between x_i and y_j
$z_0 \leftarrow x, T_x$, $z_N \leftarrow y, T_y$ ▷ Mix initial and final with Gaussian maps
for k = 1 : N **do**
  $z_{k,i} \leftarrow \frac{N-k}{N} z_0 + \frac{k}{N} z_N$
end for
while Final Map Has Not Converged **do** ▷ Step 2: Forward Step
  for $k = 1 : N$ **do**
    z_k = Solve Quasi-Newtonian(z_{k-1},z_k)
  end for
end while
FirstMap = z_N ▷ Step 3: Make Global Mapping Linear
while Maps Do Not Transport Linearly **do**
  for $k = 1 : N - 1$ **do**
    $z_{k,i} \leftarrow \frac{N-k}{N} + \frac{k}{N} \cdot$ FirstMap
  end for
  for $k = 1 : N$ **do**
    z_k = Solve Quasi-Newtonian(z_{k-1},z_k)
  end for
end while
return z_N

To solve the minimax problem between nearby sets of points, we employ a standard Quasi Newton method described in [Essid et al., 2019], such that we are minimizing the gradient of the Lagrangian defined by the two nearby collections of points. In the pseudo-code, this refers to the function titled: Solve Quasi-Newtonian. In addition, before beginning step 2, our α parameters were initialized to accentuate the identity function 10, such that we could guarantee that each iteration of gradient descent will converge. Our β parameters were initialized to fit to the closed form optimal transport function similarly used for T_x and T_y. Is similarly used and described in detail in both [Essid et al., 2018] and [Kuang and Tabak, 2017]. As the transport of Gaussian is already solved, we initialize the β parameters in this manner so that our more complex distribution has a foundational "guess" to start off with. The final set of β parameters can then build off of this initial guess to mold to a more complex distribution.

Using this algorithm, we can get towards a proper mapping from our initial set of points to our final set of points.

6 Results

After applying our algorithm to an initial data set, we can gain a mapping from an initial distribution to a final distribution. These are a few examples of preforming this transformation, where on the left we have the initial distribution of points, and on the right we have the final distribution of points. In many cases, the initial distribution has less points than the target map being pushed to. This is simulating cases where an image from the year 2000 is mapped to am image from 2015. An image from 2000 would have significantly less pixels from a map from 2015. Even with this limitation, the algorithm would find a way of mapping points optimally. In each of these examples, there are 100 less points between the initial distribution and the final distribution.

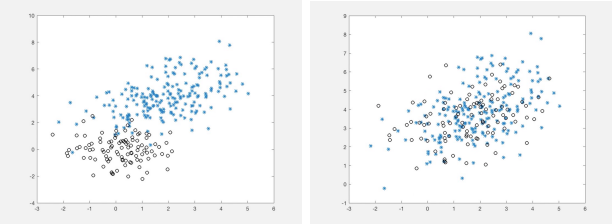

Figure 3: A transportation from a Gaussian to another Gaussian using the algorithm. We used 50 separate intermediate interpolated sets. The blue points are the target distribution and the black points show a before and after mapping onto this target.

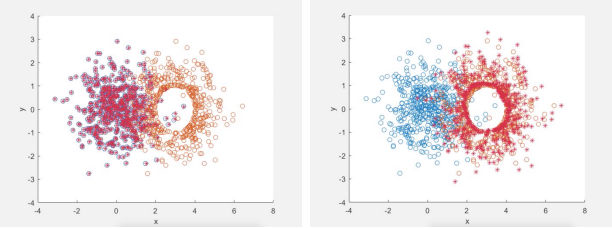

Figure 4: A transportation from a Gaussian to an annular distribution using the algorithm. We used 100 separate intermediate interpolated sets. The blue coordinates are the initial distribution, the orange points are the target, and the red points show how they have mapped.

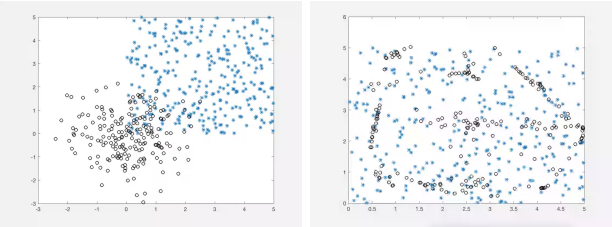

Figure 5: A transportation from a Gaussian to a rectangular distribution of points. We used 300 separate intermediate interpolated sets.The blue points are the target distribution and the black points show a before and after mapping onto this target.

7 Discussion

Although our method works for somewhat complicated data sets, there are some limitations to the validity of our approach. In5 for example, our points seem to 'clump together' and find difficulty when making extreme changes in the shape of the initial distribution and final distribution. A possible explanation for this clumping may be from the Gaussian functions being used in the Lagrangian. These Gaussian terms have the ability to pull or push together groups of points during the mini max process. A future project may involve experimentation with other function spaces for the non-linear terms of the Lagrangian. Another possible approach to solving issues regarding clumping might be to initialize our α's and β's in a less-Gaussian manner. Initializing them this way has the disadvantage of adding bias towards fitting our distributions to Gaussian distributions, even if they are more complex. As it currently stands, we initialize the parameters to an optimal Gaussian transport, which may not be reasonable for very complex collections of points. Another future project might explore novel initializing procedures for our function.

Now that we have developed a system with which to map points onto each other, the next step in developing a change detection algorithm is to identify what it means for a collection of points to have local change. A future project might include applying this system to a collection of weighted points representing an image, and finding a way of identifying how that image has evolved from its initial representation to its final representation.

8 Conclusion

In this paper, we developed a strategy to optimally map weighted points to some target set of points. Thank you to all who helped in the organization of the 2019 AM-SURE program and my advisors, Dr. Tabak and Dr. Laefer, in helping me complete my project. Thank you also to the National Science Foundation for providing a portion of the funding for this project.

References

[Essid et al., 2018] Essid, M., Laefer, D., and Tabak, E. G. (2018). Adaptive optimal transport. *arXiv preprint arXiv:1807.00393*.

[Essid et al., 2019] Essid, M., Tabak, E., and Trigila, G. (2019). An implicit gradient-descent procedure for minimax problems. *arXiv preprint arXiv:1906.00233*.

[Galichon, 2018] Galichon, A. (2018). *Optimal transport methods in economics*. Princeton University Press.

[Kuang and Tabak, 2017] Kuang, M. and Tabak, E. (2017). Preconditioning of optimal transport. *SIAM Journal for Scientific Computing*.

[Villani, 2003] Villani, C. (2003). *Topics in optimal transportation*. Number 58. American Mathematical Soc.