Let's Get Movin': Frontiers of Optimal Transport

Jiafeng Chen
School of Engineering and Applied Sciences
Harvard University
jiafengchen@college.harvard.edu

Francisco Rivera
School of Engineering and Applied Sciences
Harvard University
frivera@college.harvard.edu

1 Introduction and Background

1.1 The Optimal Transport Problem and Basic Properties

Let $\mathbf{a} \in \mathbb{R}^n_{\geq 0}$ and $\mathbf{b} \in \mathbb{R}^m_{\geq 0}$ be two vectors of non-negative real numbers such that $\mathbf{a}^T \mathbb{1} = \mathbf{b}^T \mathbb{1} = 1$, where a_i and b_j represents the amount of "mass" at different positions. We are interested in a matrix $\mathbf{P} \in \mathbb{R}^{n \times m}_{\geq 0}$ where P_{ij} prescribes the amount of mass flowing from a_i to b_j. Naturally, we must have $\sum_j P_{ij} = a_i$ and $\sum_i P_{ij} = b_j$, so that mass is preserved. Let $\mathbf{C} \in \mathbb{R}^{n \times m}$ be a *cost matrix*, where C_{ij} represents the cost of transporting one unit of a_i to b_j. The *Kantorovich optimal transport problem* is the following linear program:

$$\min_{\mathbf{P} \in \mathbf{U}(\mathbf{a}, \mathbf{b})} \sum_{i,j} P_{ij} C_{ij} \tag{Kantorovich}$$

where

$$\mathbf{U}(\mathbf{a}, \mathbf{b}) = \{\mathbf{P} \in \mathbb{R}^{n \times m}_{\geq 0} : \mathbf{P}\mathbb{1} = \mathbf{a}, \mathbf{P}^T \mathbb{1} = \mathbf{b}\}.$$

Remark 1 (A probabilistic view). Let X, Y be discrete random variables with support $\mathbf{x} \in \mathcal{X}^m$ and $\mathbf{y} \in \mathcal{Y}^n$, such that their marginal distributions are prescribed by $P(X = x_i) = a_i$ and $P(Y = y_j) = b_j$. Let $P_{ij} = P(X = x_i, Y = y_j)$, making $\mathbf{P} = (P_{ij})_{ij}$ a joint distribution. Let $C : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a cost function. Then the (Kantorovich) is equivalent to the following:

$$\min_{\mathbf{P}} \mathbb{E}_{(X,Y) \sim \mathbf{P}} \left[C(X, Y) \right],$$

where the expectation is taken over valid joint distributions of (X, Y) respecting the marginal distributions. Note that when $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ and $C(x, y) = \|x - y\|^p$, then the optimal value of the above problem for X, Y is the p-Wasserstein distance between the marginal distributions μ_X, μ_Y (up to a power of $1/p$), for discrete, d-dimensional distributions with finite support.

Duality It can be shown that the dual problem of the above LP takes the following form:

$$\max_{(\mathbf{f}, \mathbf{g}) \in \mathbb{R}^m \times \mathbb{R}^n} \sum_i a_i f_i + \sum_j b_j g_j$$
$$\text{subjected to } f_i + g_j \leq C_{ij} \text{ for all } i, j.$$

The dual problem has a neat economic intuition. Suppose Kevin needs to transport $\mathbf{a}$ to $\mathbf{b}$ but does not understand how. Francisco, his profit-maximizing colleague, offers him a deal, where Kevin will pay f_i for Francisco to pick up a unit of mass at a_i and pay g_j for Francisco to dropoff a unit of mass at b_j. At minimum, Kevin knows that if any $f_i + g_j > C_{ij}$, then Francisco is ripping him off, as

Preprint. Work in progress.

transporting one unit from a_i to b_j costs exactly C_{ij}. If, on the other hand, $f_i + g_j \leq C_{ij}$ is satisfied, then given any $\mathbf{P}$, we have

$$\sum_{i,j} C_{ij}P_{ij} \geq \sum_{i,j} f_i P_{ij} + \sum_{i,j} g_j P_{ij} = \sum_i f_i a_i + \sum_j g_j b_j, \qquad \text{(Weak duality)}$$

which means that Kevin cannot lose by taking Francisco's deal. Strong duality, which indeed holds, here imply that Francisco's optimal profit is zero, and the trade is fair.

Given optimal solutions to the primal and dual, $\mathbf{P}$, $(\mathbf{f}, \mathbf{g})$, respectively, *complementary slackness* implies that

$$P_{ij}(C_{ij} - f_i - g_j) = 0 \qquad \text{(Complementary slackness)}$$

holds, which means that $\mathbf{P}_{ij} \neq 0 \implies C_{ij} = f_i - g_j$ and vice versa.

1.2 Applications

2 Exact Methods

2.1 Linear Programming

The Kantorovich Optimal Transport problem as formulated is an LP. This means that a natural first-step to benchmarking the problem difficulty is to use a generic LP solver. The only difficulty we face when doing this is that we must flatten our arguments from matrices to vectors in order to fit the API of a generic LP solver, such as `cvxopt.solvers.lp` which we employ. Recall that our problem inputs are $\mathbf{a} \in \mathbb{R}^n_{\geq 0}$, $\mathbf{b} \in \mathbb{R}^m_{\geq 0}$ with sum of entries equal to 1 and $\mathbf{C} \in \mathbb{R}^{n,m}$. We flatten C into a vector $\mathbf{c} \in \mathbb{R}^{nm}$ and define the matrix

$$\mathbf{A} := \begin{bmatrix} I_n \otimes \mathbf{1}_m \\ \mathbf{1}_n \otimes I_m \end{bmatrix}$$

which allows us to formulate our LP as,

$$\min \mathbf{c}^\top \mathbf{x} \text{ such that } \mathbf{x} \geq 0, \mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}.$$

We may then benchmark the performance of this data by generating data. We generate random square $n \times n$ cost matrices by making each entry i.i.d. $\chi^2(1)$, and perform optimal transport on

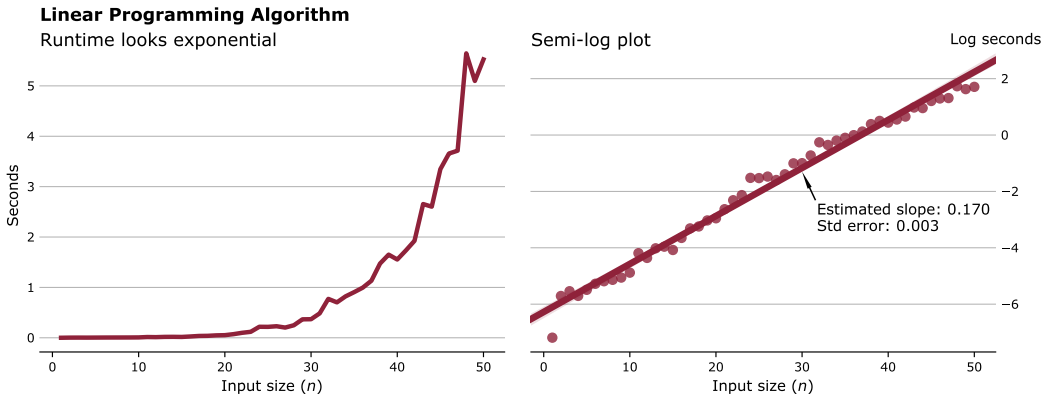

Figure 1: LP Solver Runtime

$\mathbf{a} = \mathbf{b} = \mathbf{1}_n/n$. The runtimes for different values of n are documented in Fig. 1. While this gives us a baseline to how quickly we can compute optimal transport problems, solving the problem for n bigger than a couple dozen quickly becomes an expensive task. Thus, we may look elsewhere for more efficient solutions that either solve a special case or provide an approximate solution.

2.2 Hungarian Algorithm

One such special case of the optimal transport problem is when $\mathbf{a} = \mathbf{b} \propto \mathbf{1}_n$ as in our simulations to test the performance of the generic LP solver. In this case, mass is being matched in unit pieces, which makes the problem more tractable. The Hungarian algorithm solves this special-case.

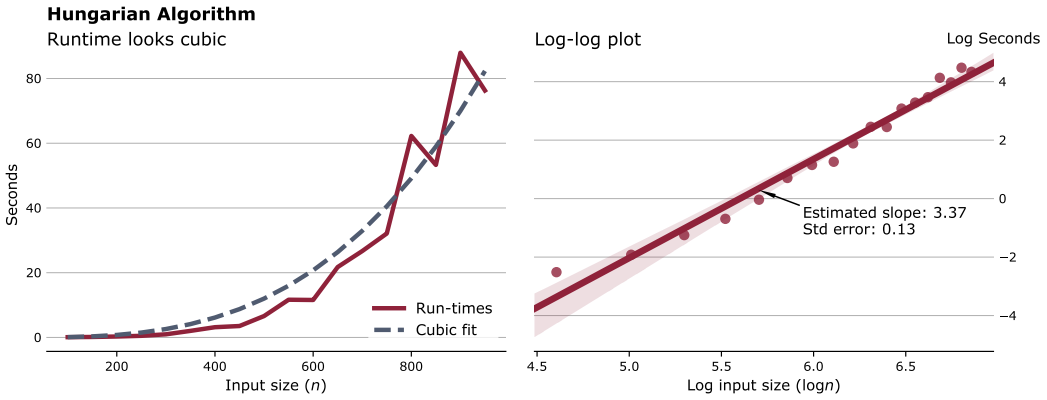

Figure 2: Hungariant Algorithm runtime

3 Inexact Methods

3.1 Entropic Regularization and Sinkhorn's Algorithm

Let

$$H(\mathbf{P}) = -\sum_{i,j} P_{ij}(\log P_{ij} - 1)$$

be the *entropy* of the transport matrix $\mathbf{P}$. Observe that

$$\nabla^2_{\mathbf{P}}(-H(\mathbf{P})) - I \text{ is positive semidefinite,}$$

and so $-H(\mathbf{P})$ is 1-strongly convex. We regularize the objective of (Kantorovich) by $-\epsilon H(\mathbf{P})$:

$$\min_{\mathbf{P}\in\mathbf{U}(\mathbf{a},\mathbf{b})} \sum_{i,j} P_{ij}C_{ij} - \epsilon H(\mathbf{P}) \qquad \text{(Entropic Regularization)}$$

so as to make the objective ϵ-strongly convex. *Entropic regularization* approximates (Kantorovich) by solving (Entropic Regularization), where

$$\arg\min_{\mathbf{P}\in\mathbf{U}(\mathbf{a},\mathbf{b})} \sum_{i,j} P_{ij}C_{ij} - \epsilon H(\mathbf{P}) \underset{n\to\infty}{\longrightarrow} \arg\min_{\mathbf{P}\in\mathbf{U}(\mathbf{a},\mathbf{b})} \sum_{i,j} P_{ij}C_{ij},$$

where the convergence is towards the maximal-entropy solution of (Kantorovich). The convergence property makes (Entropic Regularization) an attractive alternative to solving optimal transport; other reasons why (Entropic Regularization) is useful is explored in a seminal article by [Cut13].

Leveraging duality theory, the Lagrangian of the regularized problem is

$$\mathcal{L}(\mathbf{P}, \mathbf{f}, \mathbf{g}) = \sum_{i,j} P_{ij} C_{ij} + \epsilon \sum_{i,j} P_{ij} (\log P_{ij} - 1) + \sum_i f_i \left(a_i - \sum_j P_{ij} \right) + \sum_j g_j \left(b_j - \sum_i P_{ij} \right),$$

where the first-order condition on $\mathbf{P}$ shows that the optimum $\mathbf{P}^\star$ satisfies

$$C_{ij} + \epsilon \log P^\star_{ij} - f^\star_i - g^\star_j = 0 \implies \mathbf{P}^\star = \operatorname{diag}(\exp(\mathbf{f}^\star/\epsilon)) \exp(-\mathbf{C}/\epsilon) \operatorname{diag}(\exp(\mathbf{g}^\star/\epsilon)) \quad (1)$$

for some $\mathbf{f}^\star, \mathbf{g}^\star$, where the exp is taken component-wise. Suppose $\mathbf{f}^\star, \mathbf{g}^\star$ is such that $\mathbf{P}^\star$ defined by (1) is within the constraints $\mathbf{U}(\mathbf{a}, \mathbf{b})$, then the value of the Lagrangian $\mathcal{L}(\mathbf{P}^\star, \mathbf{f}^\star, \mathbf{g}^\star)$ equals the value of (Entropic Regularization), and since $\mathbf{P}^\star$ is chosen to maximize the strongly concave function $\mathbf{P} \mapsto \mathcal{L}(\mathbf{P}, \mathbf{f}^\star, \mathbf{g}^\star)$, we have that $(\mathbf{P}^\star, \mathbf{f}^\star, \mathbf{g}^\star)$ solves both programs.

Before proceeding, we define some notation that is a hybrid of [PC+17] and [AWR17]. Let $\mathbf{A} = \exp(-\mathbf{C}/\epsilon)$, $\mathbf{u} = \exp(\mathbf{f}/\epsilon)$, $\mathbf{v} = \exp(\mathbf{g}/\epsilon)$, $\mathbf{x} = \log \mathbf{u}$, $\mathbf{y} = \log \mathbf{v}$, $\mathbf{X} = \operatorname{diag}(\mathbf{u})$, $\mathbf{Y} = \operatorname{diag}(\mathbf{v})$, where $\exp, \log$ are taken componentwise. Thus, we aim to find vectors $\mathbf{u}, \mathbf{v}$, such that

$$\mathbf{P}^\star = \mathbf{X}\mathbf{A}\mathbf{Y}.$$

satisfies the constraints $\mathbf{U}(\mathbf{a}, \mathbf{b})$. In terms of $\mathbf{u}, \mathbf{v}$, the constraints are

$$\mathbf{u} \odot (\mathbf{A}\mathbf{v}) = \mathbf{a} \quad \text{and} \quad \mathbf{v} \odot (\mathbf{A}^T \mathbf{u}) = \mathbf{b} \quad (2)$$

(2) shows that, given $\mathbf{v}$, we can only set $\mathbf{u} = \frac{\mathbf{a}}{\mathbf{A}\mathbf{v}}$, where division is componentwise. This motivates the Sinkhorn-Knopp algorithm [SK67], which iteratively updates $\mathbf{u}, \mathbf{v}$ such that one of the constraints hold exactly in this fashion. In practical implementation, to prevent numerical underflow, the algorithm should work in log-space with $\mathbf{x}, \mathbf{y}$. Algorithm 1 presents a numerically stable implementation of Sinkhorn-Knopp that terminates when the ℓ_1 error on constraint violation gets sufficiently small. At each step of Algorithm 1, if k is odd, then $\mathbf{P}^{(k)}$ matches row-wise with $\mathbf{a}$, and if k is even, then $\mathbf{P}^{(k)}$ matches column-wise with $\mathbf{b}$.

Algorithm 1 Numerically stable Sinkhorn-Knopp algorithm as presented in [AWR17]

function SINKHORN($\mathbf{A}, \mathbf{a}, \mathbf{b}, \epsilon$)
  Initialize $k = 0$
  $\mathbf{P}^{(0)} \leftarrow \mathbf{A}, \mathbf{x}^0 \leftarrow 0, \mathbf{y}^0 \leftarrow 0$
  while $\left\|\mathbf{P}^{(k)}\mathbb{1} - \mathbf{a}\right\|_1 + \left\|(\mathbf{P}^{(k)})^T\mathbb{1} - \mathbf{b}\right\|_1 > \epsilon$ **do**
    $k \leftarrow k + 1$
    if k is odd **then**
      $\Delta\mathbf{x} \leftarrow \log \frac{\mathbf{a}}{\mathbf{P}^{(k-1)}\mathbb{1}}$
      $\mathbf{x}^k \leftarrow \mathbf{x}^{k-1} + \Delta\mathbf{x}, \mathbf{y}^k \leftarrow \mathbf{y}^{k-1}$
    end if
    if k is even **then**
      $\Delta\mathbf{y} \leftarrow \log \frac{\mathbf{b}}{(\mathbf{P}^{(k-1)})^T\mathbb{1}}$
      $\mathbf{x}^k \leftarrow \mathbf{x}^{k-1}, \mathbf{y}^k \leftarrow \mathbf{y}^{k-1} + \Delta\mathbf{y}$
    end if
    $\mathbf{P}^{(k)} \leftarrow \mathbf{X}^k\mathbf{A}\mathbf{Y}^k$
  end while
  return $\mathbf{P}^{(k)} / \left\|\mathbf{P}^{(k)}\right\|_1$ ▷ $\mathbf{P}^{(0)}$ is not normalized; $\mathbf{P}^{(k)}, k > 0$ is normalized.
end function

Note that Algorithm 1 does not output a matrix that satisfies the conditions $\mathbf{U}(\mathbf{a}, \mathbf{b})$ exactly. However, [AWR17, Algorithm 2] analyzes a rounding scheme that forces the output to be inside $\mathbf{U}(\mathbf{a}, \mathbf{b})$. The rounding scheme introduces error on the order of ϵ and thus does not pose any problems for solving the original program (Entropic Regularization). The rest of this section will be devoted to an analysis of Algorithm 1.

3.1.1 Convergence

A priori, it is unclear that Algorithm 1 converges, nor is the speed of convergence obvious. [AWR17] presents an extremely succinct analysis of Algorithm 1, which we reproduce in this section. Let

$$f(\mathbf{x}, \mathbf{y}) = \sum_{i,j} A_{ij} e^{x_i + y_j} - \sum_i a_i x_i - \sum_j b_j y_j.$$

The first order conditions for the optimality of f gives $\mathbf{x}, \mathbf{y}$ that ensures (2). Sinkhorn-Knopp can be viewed as coordinate descent on f: To optimize x_i^k, the first order condition yields

$$x_i^k = \log \frac{a_i}{\sum_j A_{ij} e^{y_j^{k-1}}} = \log \frac{a_i}{e^{x_i^{k-1}} \sum_j A_{ij} e^{y_j^{k-1}}} + x_i^{k-1} = \log \frac{a_i}{\mathbf{P}^{(k-1)} \mathbb{1}} + x_i^{k-1},$$

which is exactly the update rule in Algorithm 1.

Remark 2. We do not need to limit ourselves to using coordinate descent to minimize f. For instance [BCLW17] uses Newton's method, and achieves similar convergence performance as Algorithm 1.

The proof of [AWR17] uses the follow strategy. First, we calculate the step gain in the Sinkhorn-Knopp algorithm for optimizing f in terms of Kullback-Leibler divergences. Next we bound the distance of f's initial value from its optimal value. Lastly, using Pinsker's inequality, we show that if Algorithm 1 does not terminate, then the step size must be larger than $C\epsilon^2$, which then gives a bound of convergence in ϵ^{-2}.

For $k > 1$, we can calculate the gain of each step [AWR17, Lemma 2]:

$$\begin{aligned} f(\mathbf{x}^{k-1}, \mathbf{y}^{k-1}) - f(\mathbf{x}^k, \mathbf{y}^k) &= \left(\sum_{i,j} A_{ij} e^{x_i^{k-1} + y_i^{k-1}} - \sum_{i,j} A_{ij} e^{x_i^k + y_i^k} \right) \\ &\quad + \langle \mathbf{a}, \mathbf{x}^k - \mathbf{x}^{k-1} \rangle + \langle \mathbf{b}, \mathbf{y}^k - \mathbf{y}^{k-1} \rangle \\ &= (1 - 1) + \mathrm{KL}(\mathbf{a} || \mathbf{P}^{(k)} \mathbb{1}) + \mathrm{KL}(\mathbf{b} || (\mathbf{P}^{(k)})^T \mathbb{1}) \\ &\qquad \text{(Writing out the inner products)} \\ &= \mathrm{KL}(\mathbf{a} || \mathbf{P}^{(k)} \mathbb{1}) + \mathrm{KL}(\mathbf{b} || (\mathbf{P}^{(k)})^T \mathbb{1}). \end{aligned} \tag{3}$$

(3) gives an elegant formulation of the gain each step exactly as a function of the Kullback-Leibler divergence between the target distribution and the current marginal distribution.

Next, we claim that the total distance in f that we are required to travel is bounded above [AWR17, Lemma 3]. Note that Algorithm 1 behaves almost identically—producing the same $\mathbf{P}^{(k)}$ at each iteration—if the input $\mathbf{A}$ passed in is multiplied by a constant. Thus we may without loss of generality replace $\mathbf{A}$ with $\mathbf{A}_0 = \mathbf{A} / \|\mathbf{A}\|_1$. Let f^k denote $f(\mathbf{x}^k, \mathbf{y}^k)$ and $f^\star$ denote $\min f$. Then we have $f(0,0) - f^1 = \mathrm{KL}(\mathbf{a} || \mathbf{A}_0 \mathbb{1}) \geq 0$. Thus

$$f^1 - f^\star \leq f(0,0) - f^\star = \sum_i a_i x_i^\star + \sum_j b_j y_j^\star \leq \max_{i,j} (x_i^\star + y_j^\star).$$

Observe that $A_{0,ij} e^{x_i^\star + y_j^\star} \leq \sum A_{0,ij} e^{x_i^\star + y_j^\star} = 1$. Therefore $x_i^\star + y_j^\star \leq -\log A_{0,ij} \leq \log \frac{\|\mathbf{A}\|_1}{\min_{i,j} A_{ij}}$. Putting these together shows that

$$f^1 - f^\star \leq \log \frac{\|\mathbf{A}\|_1}{\min_{i,j} A_{ij}}. \tag{4}$$

Lastly, Pinsker's inequality [AWR17, Lemma 4], a well-known result in information theory, states that for any probability measures p, q,

$$\|p - q\|_1^2 \leq 2 \, \mathrm{KL}(p || q).$$

Applying Pinsker's inequality shows that every step in Algorithm 1 before termination has

$$\epsilon < \left\| \mathbf{P}^{(k)} \mathbb{1} - \mathbf{a} \right\|_1 + \left\| (\mathbf{P}^{(k)})^T \mathbb{1} - \mathbf{b} \right\|_1 \leq \left[4 \left(\mathrm{KL}(\mathbf{a} || \mathbf{P}^{(k)} \mathbb{1}) + \mathrm{KL}(\mathbf{b} || (\mathbf{P}^{(k)})^T \mathbb{1}) \right) \right]^{1/2}.$$

Therefore we improve f by more than $\frac{1}{4}\epsilon^2$ every step before termination. Since the total improvement is bounded by (4), we arrive at the main result of [AWR17]:

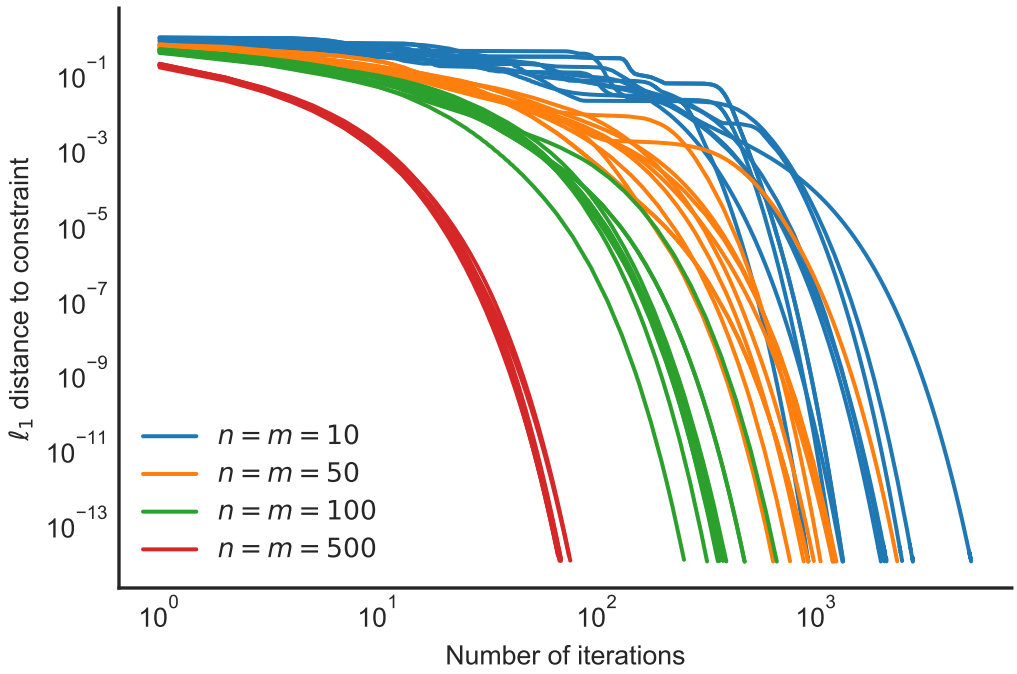

Figure 3: Numerical experiments on the speed of ℓ_1 convergence for Algorithm 1

Theorem 1 ([AWR17]). *Algorithm 1 terminates in at most* $4\epsilon^{-2} \log \frac{\|\mathbf{A}\|_1}{\min_{i,j} A_{ij}}$ *steps.*

3.1.2 Numerical Experiments

We implement Algorithm 1 and compute the ℓ_1 distance $\left\|\mathbf{P}^{(k)}\mathbb{1} - \mathbf{a}\right\|_1 + \left\|(\mathbf{P}^{(k)})^T\mathbb{1} - \mathbf{b}\right\|_1$ as a function of k. We generate random marginal distribution targets $\mathbf{a}, \mathbf{b}$ by sampling from a uniform distribution on the unit interval and normalizing by the sum. We generate random cost matrices by sampling each component from a uniform distribution. We plot the log-log plot in Fig. 3. Note that if the results of Theorem 1 is tight for these random matrices and constraints, then the log-log plot should look linear with slope -2. This is not what we observe in Fig. 3, suggesting that the average case—if not worst case—of Algorithm 1 may be better than $O(\epsilon^{-2})$. In fact, if we plot the log of error against k, we find that the relationship is linear, suggesting that, for these random matrices and distributions, the behavior of Algorithm 1 is $O(\log(1/\epsilon))$.

3.1.3 Towards a Tightness Analysis of Theorem 1

Section 3.1.2 suggests that the bound of ϵ^{-2} may be far too generous. In this section we sketch a potential direction in analyzing the worst-case performance of Algorithm 1. [Sas15] gives a reverse Pinsker's inequality: For discrete probability measures on finite support p, q,

$$\mathrm{KL}(p||q) \leq \log\left(1 + \frac{1}{q_{\min}}\|p - q\|_1^2\right) \leq \frac{1}{q_{\min}}\|p - q\|_1^2,$$

where $q_{\min}$ is the minimum of q on its support. The inequality suggests that before termination and after the ℓ_1 error reaches some δ, every step of Algorithm 1 improves f by at most

$$\mathrm{KL}(\mathbf{a}||\mathbf{P}^{(k)}\mathbb{1}) + \mathrm{KL}(\mathbf{b}||(\mathbf{P}^{(k)})^T\mathbb{1}) \leq \frac{C}{\min(\mathbf{P}^{(k)}\mathbb{1}, (\mathbf{P}^{(k)})^T\mathbb{1})}\delta^2,$$

for some constant C, where the minimum is taken element-wise. We can remove the dependence on k of the right-hand side by the following consideration. Let k' be the first time that the ℓ_1-error is below $\frac{1}{2}\min(\mathbf{a}, \mathbf{b})$. Then for every $k > k'$, $\min(\mathbf{P}^{(k)}\mathbb{1}, (\mathbf{P}^{(k)})^T\mathbb{1}) \geq \frac{1}{2}\min(\mathbf{a}, \mathbf{b})$, and so the bound becomes

$$\mathrm{KL}(\mathbf{a}||\mathbf{P}^{(k)}\mathbb{1}) + \mathrm{KL}(\mathbf{b}||(\mathbf{P}^{(k)})^T\mathbb{1}) \leq \frac{C'}{\min(\mathbf{a}, \mathbf{b})}\delta^2, \tag{5}$$

for $k > k'$. Let k_ϵ be the step immediately before termination. Let k_δ be the first iteration such that $k_\delta > k'$ and that k_δ reaches ℓ_1-error of δ. If we can show that there is some inputs $\mathbf{A}, \mathbf{a}, \mathbf{b}, \epsilon$ for which we can bound the total improvement $f^{k_\eta} - f^{k_\epsilon}$ below by some quantity $M(\mathbf{A}, \mathbf{a}, \mathbf{b}, \epsilon, \delta)$, then for the inputs $\mathbf{A}, \mathbf{a}, \mathbf{b}, \epsilon$, Algorithm 1 has a lower bound of $\Omega\left(M(\mathbf{A}, \mathbf{a}, \mathbf{b}, \epsilon, \delta)\delta^{-2}\min(\mathbf{a}, \mathbf{b})\right)$. Here, δ should be implicitly a function of ϵ, e.g. $\delta = \sqrt{\epsilon}$, so that the bound is a function of ϵ. Choosing $\delta(\epsilon)$ such that the bound $M(\mathbf{A}, \mathbf{a}, \mathbf{b}, \epsilon, \delta)$ is tractable is a major difficulty, which we leave to future work.

3.2 Genetic Algorithm

Recent advances in machine learning points towards the versatility of genetic algorithms, even as a competitive alternative to deep learning based on gradient methods [SMC+17]. Inspired by these advances, we consider the following genetic algorithm

Algorithm 2 A toy genetic algorithm for optimal transport

```
function GeneticOT(φ, γ, k, a, b, T)
    initial ← ab^T
    family ← Array[φ], initialized to initial
    children ← Array[φγ], initialized to empty
    for generation in 1 : T do
        for person in family do
            for person′ in Subset(family, γ) do        ▷ Subset chooses a subset of size γ
                child ← Marry(person, person′)
                child ← Perturb(child, k)
                Add child to children
            end for
        end for
        Choose the φ elements in children with minimal cost for (Kantorovich) and set to family
    end for
    return element in family that achieves minimal cost
end function
```

Here Marry picks two probability matrices and averages them elementwise, and Perturb chooses random indices i_1, i_2, j_1, j_2 and moves mass in the probability matrix in such a way that preserves the marginal distributions and respects the constraint that $0 \leq p \leq 1$ for all components p. Such perturbations are repeated k times. ϕ is a parameter that controls the size of families, and γ is a parameter that decides the number of offsprings. At each generation, we take the top ϕ matrices that achieve the lowest cost. We hoped that the population drifts towards the optimal transport matrix as the number of generation increases.

The empirical results we obtain are less than encouraging, however. We plot the optimal cost each generation with $n = m = 4, \phi = 100, \gamma = 5$ for various values of k in Fig. 4. Curiously, the improvement in the first generation is always large, with a magnitude of improvement increasing in k, but the algorithm seems to achieve some non-optimal steady state as generation goes on. Of course, Algorithm 2 can potentially be improved by thinking about the design of Marry and Perturb more carefully, perhaps leveraging some information about $\mathbf{C}$.

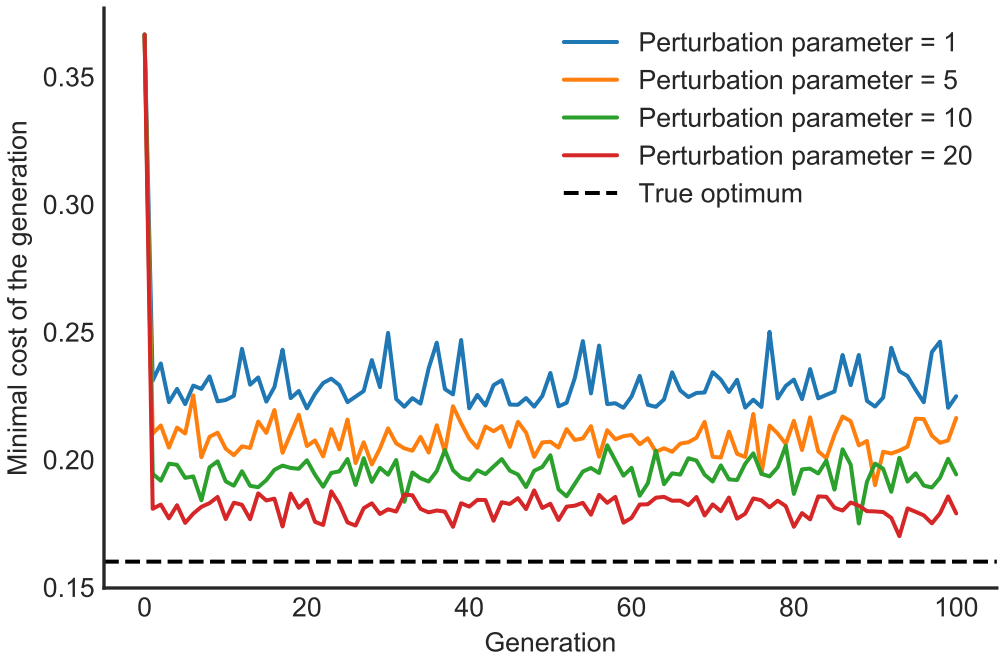

Figure 4: Non-convergence of Algorithm 2

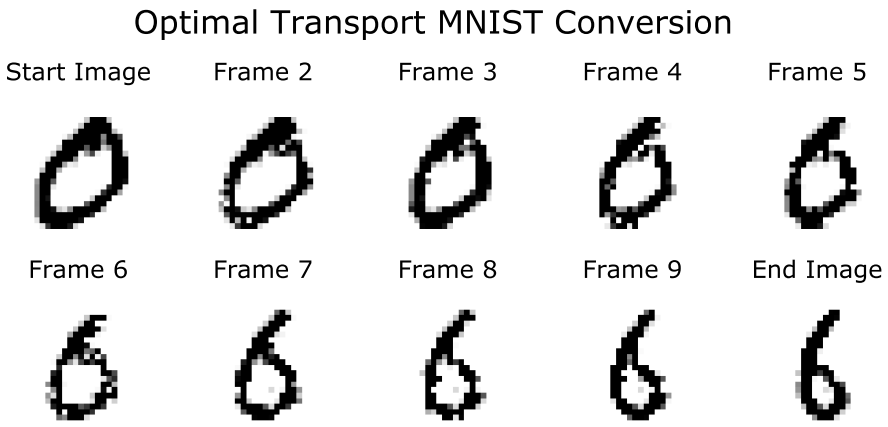

Figure 5: Optimal transport on MNIST images

4 Applications and Experiments

The applications of the optimal transport problem to the physical transportation of objects is practically explicit in the problem statement. This is but one of many applications, though. In particular, we explore the use of optimal transport in the image processing of MNIST images.

Optimal transport provides a natural language with which to talk about image similarity. If we consider the darkness of a pixel as "mass," the similarity of two images could be quantified as the

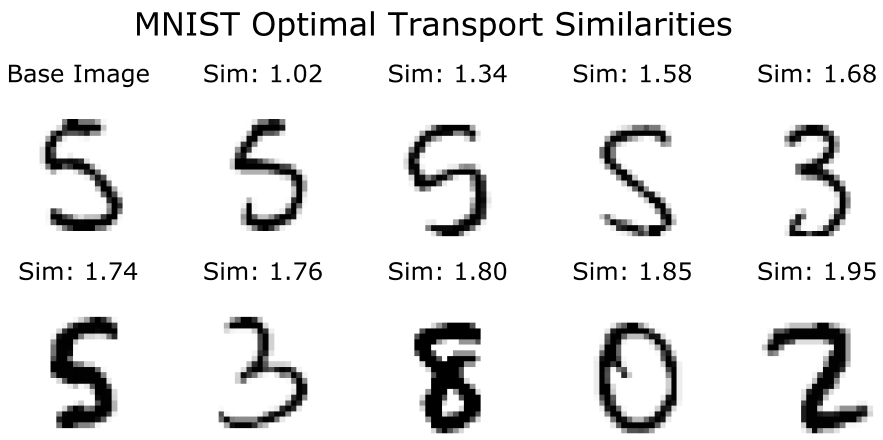

Figure 6: MNIST Similarity Metric

units of mass times distance that need to be moved in order to transform it into the second image. This requires us to normalize the image such that the pixel intensities sum to 1, but this is not a problem with MNIST images which do not differ significantly in their average brightness. The transport also requires a choice of cost function between two pixels. We opt for an L^2 penalization on the distance between the pixels, but alternative metrics such as L^1 perform similarly Fig. 5 illustrates what this transformation looks like for a "0" being transported into a "6."

To verify that the similarity is behaving as we expect, we can compare the similarity of a number "5" against its most similar images in a small sample of the MNIST dataset. The most similar images are also 5's, as depicted in Fig. 6.

5 Conclusion

References

[AWR17] Jason Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In *Advances in Neural Information Processing Systems*, pages 1961–1971, 2017.

[BCLW17] Christoph Brauer, Christian Clason, Dirk Lorenz, and Benedikt Wirth. A sinkhorn-newton method for entropic optimal transport. *arXiv preprint arXiv:1710.06635*, 2017.

[Cut13] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

[PC+17] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. Technical report, 2017.

[Sas15] Igal Sason. On reverse pinsker inequalities. *arXiv preprint arXiv:1503.07118*, 2015.

[SK67] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[SMC+17] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O Stanley, and Jeff Clune. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv preprint arXiv:1712.06567*, 2017.