Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration

Jason Altschuler*, Jonathan Weed*, and Philippe Rigollet†

Massachusetts Institute of Technology

Abstract. Computing optimal transport distances such as the earth mover's distance is a fundamental problem in machine learning, statistics, and computer vision. Despite the recent introduction of several algorithms with good empirical performance, it is unknown whether general optimal transport distances can be approximated in near-linear time. This paper demonstrates that this ambitious goal is in fact achieved by Cuturi's *Sinkhorn Distances*, and provides guidance towards parameter tuning for this algorithm. This result relies on a new analysis of Sinkhorn iterations that also directly suggests a new algorithm Greenkhorn with the same theoretical guarantees. Numerical simulations illustrate that Greenkhorn significantly outperforms the classical Sinkhorn algorithm in practice.

1. INTRODUCTION

Computing distances between probability measures on metric spaces, or more generally between point clouds, plays an increasingly preponderant role in machine learning [SL11, MJ15, LG15, JSCG16, ACB17], statistics [FCCR16, PZ16, SR04, BGKL17] and computer vision [RTG00, BvdPPH11, SdGP+15]. A prominent example of such distances is the *earth mover's distance* introduced in [WPR85] (see also [RTG00]), which is a special case of Wasserstein distance, or optimal transport (OT) distance [Vil09].

While OT distances exhibit a unique ability to capture geometric features of the objects at hand, they suffer from a heavy computational cost that has been prohibitive in large scale applications until the recent introduction to the machine learning community of *Sinkhorn Distances* by Cuturi [Cut13]. Combined with other numerical tricks, these recent advances have enabled the treatment of large clouds of points in computer graphics such as triangle meshes [SdGP+15] and high-resolution neuroimaging data [GPC15]. Sinkhorn Distances mainly rely on the idea of *entropy penalization*, which has been implemented in similar problems at least since Schrödinger [Sch31, Leo14]. This powerful idea has been successfully applied to a variety of contexts not only as a statistical tool for model

*This work was supported in part by NSF Graduate Research Fellowship DGE-1122374.

†This work was supported in part by NSF CAREER DMS-1541099, NSF DMS-1541100, DARPA W911NF-16-1-0551, ONR N00014-17-1-2147 and a grant from the MIT NEC Corporation.

selection [JRT08, RT11, RT12] and online learning [CBL06], but also as an optimization gadget in first-order optimization methods such as mirror descent and proximal methods [Bub15].

Related work. Computing an OT distance amounts to solving the following linear system:

$$\min_{P \in \mathcal{U}_{r,c}} \langle P, C \rangle \,, \tag{1}$$

where

$$\mathcal{U}_{r,c} := \left\{ P \in \mathbb{R}_+^{n \times n} \,:\, P\mathbf{1} = r \,, P^\top \mathbf{1} = c \right\},$$

is the *transport polytope*, $\mathbf{1}$ is the all-ones vector in $\mathbb{R}^n$, $C \in \mathbb{R}_+^{n \times n}$ is a given *cost matrix*, and $r \in \mathbb{R}^n, c \in \mathbb{R}^n$ are given vectors with positive entries that sum to one. Typically C is a matrix containing pairwise distances, but in this paper we allow C to be any positive dense matrix with bounded entries since our results are more general. For brevity, this paper focuses on square matrices C and P, since extensions to the rectangular case are straightforward.

This paper is at the intersection of two lines of research: a practical one that pursues fast algorithms for optimal transport problems and a theoretical one that aims at finding (near) linear time approximation algorithms for simple problems that are already known to run in polynomial time.

Noticing that (1) is a linear program with $O(n)$ linear constraints and certain graphical structure, one can use the recent Lee-Sidford linear solver to find a solution in time $\widetilde{O}(n^{2.5})$ [LS14], improving over the previous standard of $O(n^{3.5})$ [Ren88]. While no practical implementation of the Lee-Sidford algorithm is known, it provides a theoretical benchmark for our methods. Their result is part of a long line of work initiated by the seminal paper of Spielman and Teng [ST04] on solving linear systems of equations, that has provided a building block for near-linear time approximation algorithms in a variety of combinatorially structured linear problems. Our work fits into this line of work in the sense that it provides the first near-linear time guarantee to approximate (1). However, our work presents a striking difference: we analyze algorithms that are also practically efficient.

Practical algorithms for computing OT distances include Orlin's algorithm for the *Uncapacitated Minimum Cost Flow* problem via a standard reduction. Akin to interior point methods, it has a provable complexity of $O(n^3 \log n)$. This cubic dependence on the dimension is also observed in practice, thereby preventing large-scale applications. To overcome the limitations of such general solvers, various ideas ranging from graph sparsification [PW09] to metric embedding [IT03, GD04, SJ08] have been proposed over the years to deal with particular cases of OT distance. At the same time, recent years have witnessed the development of scalable methods for general OT that leverage the idea of entropic regularization [Cut13, BCC+15, GCPB16]. However, their apparent practical efficacy still lacks theoretical guarantees. In particular, the existence of algorithms to compute or approximate general OT distances in time nearly linear in the input size n^2 is an open question. Therefore, new tools are needed to develop provably near-linear time algorithms for OT distance computation.

Our contribution. The contribution of this paper is twofold. First we demonstrate that, with an appropriate choice of parameters, the algorithm for Sinkhorn Distances introduced in [Cut13] is in fact a *near-linear time* approximation algorithm for computing OT distances between discrete measures. This is the first proof that such near-linear time results are achievable for optimal transport. Core to our work is a new analysis of the Sinkhorn iteration algorithm, which we show converges in a number of iterations independent of the dimension n of the matrix to balance using a new and arguably more natural analysis of these iterations. In particular, this analysis directly suggests a greedy variant of Sinkhorn iterations that also provably runs in near-linear time and significantly outperforms the classical algorithm in practice. Finally, while most approximation algorithms output an approximation of the optimum *value* of the linear program (1), we also describe a simple rounding algorithm that provably outputs a feasible solution to (1). Specifically, for any $\varepsilon > 0$ and bounded, positive cost matrix C, we describe an algorithm that runs in time $\widetilde{O}(n^2/\varepsilon^4)$ and outputs $\hat{P} \in \mathcal{U}_{r,c}$ such that

$$\langle \hat{P}, C \rangle \leq \min_{P \in \mathcal{U}_{r,c}} \langle P, C \rangle + \varepsilon$$

Notation. We denote non-negative real numbers by $\mathbb{R}_+$, the set of integers $\{1, \ldots, n\}$ by $[n]$, and the d-dimensional simplex by $\Delta_d := \{x \in \mathbb{R}_+^d : \sum_{i=1}^d x_i = 1\}$. For two probability distributions $p, q \in \Delta_d$ such that p is absolutely continuous w.r.t. q, we define the entropy $H(p)$ of p and the Kullback-Leibler divergence $\mathcal{K}(p\|q)$ between p and q respectively by

$$H(p) = \sum_{i=1}^d p_i \log \frac{1}{p_i}, \qquad \mathcal{K}(p\|q) := \sum_{i=1}^d p_i \log \left(\frac{p_i}{q_i}\right).$$

We use $\mathbf{1}$ to denote the all-ones vector in $\mathbb{R}^n$. For a matrix $A = (A_{ij})$, we denote by $\exp(A)$ the matrix with entries $(e^{A_{ij}})$. For $A \in \mathbb{R}^{n \times n}$, we denote its row and columns sums by $r(A) := A\mathbf{1} \in \mathbb{R}^n$ and $c(A) := A^\top \mathbf{1} \in \mathbb{R}^n$, respectively. We write $\|A\|_\infty = \max_{ij} |A_{ij}|$. For two matrices of the same dimensions, we denote the Frobenius inner product of A and B by $\langle A, B \rangle = \sum_{ij} A_{ij} B_{ij}$. For a vector $x \in \mathbb{R}^n$, we write $\mathbf{D}(x) \in \mathbb{R}^{n \times n}$ to denote the diagonal matrix with entries $(\mathbf{D}(x))_{ii} = x_i$.

For any two nonnegative sequences $(u_n)_n, (v_n)_n$, we write $u_n = \widetilde{O}(v_n)$ if there exist positive constants C, c such that $u_n \leq C v_n (\log n)^c$. For any two real numbers, we write $a \wedge b = \min(a, b)$.

2. OPTIMAL TRANSPORT IN NEAR-LINEAR TIME

In this section, we describe the main algorithm studied in this paper. Pseudocode appears in Algorithm 1.

The core of our algorithm is the computation of an *approximate Sinkhorn projection* of the entrywise-exponentiated matrix $A = \exp(-\eta C)$ (Step 1). We discuss this step and its connection to entropic penalization in Section 2.1. Since our approximate Sinkhorn projection is not guaranteed to lie in the feasible set, we round our approximation to ensure that it lies in $\mathcal{U}_{r,c}$ (Step 2). More details about the rounding procedure appear in Section 2.2.

Our main theorem about Algorithm 1 is the following accuracy and runtime guarantee.

Algorithm 1 APPROXOT(C, r, c, ε)

$\eta \leftarrow \frac{4 \log n}{\varepsilon}$, $\varepsilon' \leftarrow \frac{\varepsilon}{4\|C\|_\infty}$

\\ Step 1: Approximately project onto $\mathcal{U}_{r,c}$

1: $A \leftarrow \exp(-\eta C)$

2: $B \leftarrow \textsc{Proj}(A, \mathcal{U}_{r,c}, \varepsilon')$

\\ Step 2: Round to feasible point in $\mathcal{U}_{r,c}$

3: Output $\hat{P} \leftarrow \textsc{round}(B, \mathcal{U}_{r,c})$

THEOREM 1. *Algorithm 1 returns a point $\hat{P} \in \mathcal{U}_{r,c}$ satisfying*

$$\langle \hat{P}, C \rangle \leq \min_{P \in \mathcal{U}_{r,c}} \langle P, C \rangle + \varepsilon$$

in $O(n^2 + S)$ operations, where S is the number of operations of the subroutine $\textsc{Proj}(A, \mathcal{U}_{r,c}, \varepsilon')$. In particular, if $\|C\|_\infty \leq L$, then S can be $O(n^2 L^3 (\log n) \varepsilon^{-3})$, so that Algorithm 1 requires $O(n^2 L^3 (\log n) \varepsilon^{-3})$ operations.

For simplicity, we state Theorem 1 in terms of elementary arithmetic operations, and do not consider bit complexity issues arising from the taking of exponentials in Step 1. It can be easily shown [KLRS08] that the maximum bit complexity throughout the execution of our algorithm is $O(L(\log n)/\varepsilon)$. As a result, factoring in bit complexity leads to a runtime of $O(n^2 L^4 (\log n)^2 \varepsilon^{-4})$, which is truly near-linear.

2.1 Approximate Sinkhorn projection

The core of our algorithm is the entropic penalty proposed by Cuturi [Cut13]:

$$P_\eta := \operatorname*{argmin}_{P \in \mathcal{U}_{r,c}} \left\{ \langle P, C \rangle - \eta^{-1} H(P) \right\}, \tag{2}$$

where H is the entrywise entropy. The solution to (2) can be characterized explicitly by analyzing its first-order conditions for optimality.

THEOREM 2. *[Cut13] For any cost matrix C and $r, c \in \Delta_n$, the minimization progam (2) has a unique minimum at $P_\eta \in \mathcal{U}_{r,c}$ of the form $P_\eta = XAY$, where $A = \exp(-\eta C)$ and $X, Y \in \mathbb{R}_+^{n \times n}$ are both diagonal matrices. The matrices (X, Y) are unique up to a constant factor.*

We call the matrix P_η appearing in Theorem 2 the *Sinkhorn projection* of A, denoted $\Pi_{\mathcal{S}}(A, \mathcal{U}_{r,c})$, after Sinkhorn, who proved uniqueness in [Sin67]. Computing $\Pi_{\mathcal{S}}(A, \mathcal{U}_{r,c})$ exactly is impractical, so we implement instead an approximate version $\textsc{Proj}(A, \mathcal{U}_{r,c}, \varepsilon')$ that outputs a matrix $B = XAY$ which may not lie in $\mathcal{U}_{r,c}$ but satisfies the condition $\|r(B) - r\|_1 + \|c(B) - c\|_1 \leq \varepsilon'$. We stress that this condition is very natural from a statistical standpoint, since it requires that $r(B)$ and $c(B)$ are close to the target marginals r and c in *total variation distance*. Prior work on approximate Sinkhorn projection focuses on the weaker condition $\|r(B) - r\|_2 + \|c(B) - c\|_2 \leq \varepsilon'$. Not only do such bounds lack statistical meaning, but they also fail to yield useful approximation guarantees for OT distances. We discuss this issue and give an algorithm to compute $\textsc{Proj}(A, \mathcal{U}_{r,c}, \varepsilon')$ in Section 3.

THEOREM 3. *Let $\|C\|_\infty \leq L$. There exists an implementation of the procedure $\textsc{Proj}(A, \mathcal{U}_{r,c}, \varepsilon')$ requiring $O(n^2 L^3 (\log n) \varepsilon^{-3})$ elementary arithmetic operations that outputs a matrix $B = XAY$ where $X, Y \in \mathbb{R}_+^{n \times n}$ are diagonal matrices and*

$$\|r(B) - r\|_1 + \|c(B) - c\|_1 \leq \varepsilon'.$$

PROOF. Theorems 5 and 6 imply that both the SINKHORN and GREENKHORN algorithms yield a matrix B of the desired accuracy in $O(n^2(\varepsilon')^{-2}\log\frac{s}{\ell})$ elementary arithmetic operations, where s is the sum of the entries of A and ℓ is the smallest entry of A. Since the matrix C is nonnegative, $s \le n^2$. The smallest entry of A is $e^{-\eta\|C\|_\infty}$, so $\log 1/\ell = \eta\|C\|_\infty$. We obtain

$$S = O(n^2(\varepsilon')^{-2}(\log n + \eta\|C\|_\infty)\eta\|C\|_\infty)\,,$$

and plugging in the values of η and ε' finishes the proof. □

2.2 Rounding to a feasible point

The rounding procedure we implement is very simple, and is based on the observation that calculating the optimal transport with respect to the total variation distance is computationally cheap.

THEOREM 4. *If $r, c \in \Delta_n$ and $F \in \mathbb{R}_+^{n\times n}$, then there exists $G \in \mathcal{U}_{r,c}$ satisfying*

$$\|G - F\|_1 \le \|r(F) - r\|_1 + \|c(F) - c\|_1\,.$$

Such a G can be computed in $O(n^2)$ time by Algorithm 2.

A proof of Theorem 4 appears in the Appendix.

Algorithm 2 ROUND$(F, \mathcal{U}_{r,c})$

1: $X \leftarrow$ diagonal with $X_{ii} = \frac{r_i}{r_i(F)} \wedge 1$
2: $F \leftarrow XF$
3: $Y \leftarrow$ diagonal with $Y_{jj} = \frac{c_j}{c_j(F)} \wedge 1$
4: $F \leftarrow FY$
5: $\mathrm{err}_r \leftarrow r - r(F)$, $\mathrm{err}_c \leftarrow c - c(F)$
6: Output $G \leftarrow F + \mathrm{err}_r \mathrm{err}_c^\top / \|\mathrm{err}_r\|_1$

2.3 Proof of Theorem 1

We have already established that Steps 1 and 2 run in S and $O(n^2)$ time, respectively, so the runtime guarantee is immediate.

Let B be the output of PROJ$(A, \mathcal{U}_{r,c}, \varepsilon')$, and let $P^* \in \operatorname{argmin}_{P\in\mathcal{U}_{r,c}}\langle P, C\rangle$ be an optimal solution to the original OT program.

We first show that $\langle B, C\rangle$ is not much larger than $\langle P^*, C\rangle$. To that end, write $r' := r(B)$ and $c' := c(B)$. Since $B = XAY$ for positive diagonal matrices X and Y, Theorem 2 implies B is the optimal solution to

$$\min_{P\in\mathcal{U}_{r',c'}} \langle P, C\rangle - \eta^{-1}H(P)\,. \tag{3}$$

By Theorem 4, there exists a matrix $P' \in \mathcal{U}_{r',c'}$ such that

$$\|P' - P^*\|_1 \le \|r' - r\|_1 + \|c' - c\|_1\,.$$

Moreover, since B is an optimal solution of (3), we have

$$\langle B, C\rangle - \eta^{-1}H(B) \le \langle P', C\rangle - \eta^{-1}H(P')\,.$$

Thus, by Hölder's inequality

$$\begin{aligned}\langle B, C\rangle - \langle P^*, C\rangle &= \langle B, C\rangle - \langle P', C\rangle + \langle P', C\rangle - \langle P^*, C\rangle \\ &\le \eta^{-1}(H(B) - H(P')) + (\|r' - r\|_1 + \|c' - c\|_1)\|C\|_\infty \\ &\le 2\eta^{-1}\log n + (\|r' - r\|_1 + \|c' - c\|_1)\|C\|_\infty\,, \end{aligned} \tag{4}$$

where we have used the fact that $0 \le H(B), H(P') \le 2\log n$.

Theorem 4 implies that the output $\hat{P}$ of $\textsc{round}(B, \mathcal{U}_{r,c}, \varepsilon')$ satisfies

$$\|B - \hat{P}\|_1 \le \|r' - r\|_1 + \|c' - c\|_1 .$$

Together with (4) and Hölder's inequality, it yields

$$\langle \hat{P}, C\rangle \le \min_{P \in \mathcal{U}_{r,c}} \langle P, C\rangle + 2\eta^{-1}\log n + 2(\|r' - r\|_1 + \|c' - c\|_1)\|C\|_\infty .$$

Applying the guarantee of $\textsc{Proj}(A, \mathcal{U}_{r,c})$ yields

$$\langle \hat{P}, C\rangle \le \min_{P \in \mathcal{U}_{r,c}} \langle P, C\rangle + \frac{2\log n}{\eta} + 2\varepsilon'\|C\|_\infty .$$

Plugging in the values of η and ε' prescribed in Algorithm 1 yields the claim.

3. LINEAR-TIME APPROXIMATE SINKHORN PROJECTION

Given a matrix A, Sinkhorn proposed a simple iterative algorithm to approximate the Sinkhorn projection $\Pi_{\mathcal{S}}(A, \mathcal{U}_{r,c})$, which is now known as the Sinkhorn-Knopp algorithm or RAS method. Despite the simplicity of this algorithm and its good performance in practice, it has been difficult to analyze. As a result, recent work showing that $\Pi_{\mathcal{S}}(A, \mathcal{U}_{r,c})$ can be approximated in near-linear time [AZLOW17, CMTV17] has bypassed the Sinkhorn-Knopp algorithm entirely. Though these results come with provable convergence guarantees, the algorithms they propose are not unimplementable in practice. In our work, we obtain a new analysis of the simple and practical Sinkhorn-Knopp algorithm, showing that also approximates $\Pi_{\mathcal{S}}(A, \mathcal{U}_{r,c})$ in near-linear time.

Pseudocode for the Sinkhorn-Knopp algorithm appears in Algorithm 3. In brief, it is an alternating projection procedure which renormalizes the rows and columns of A in turn so that they match the desired row and column marginals r and c. At each step, it prescribes to either modify all the rows by multiplying row i by $r_i / r_i(A)$ for $i \in [n]$, or to do the analogous operation on the columns. (We interpret the quantity $0/0$ as 1 in this algorithm if ever it occurs.)

It is clear that, if Algorithm 3 terminates, then its output B satisfies $\mathrm{dist}(B, \mathcal{U}_{r,c}) \le \varepsilon'$. (The choice of metric in which to measure $\mathrm{dist}(A, \mathcal{U}_{r,c})$ will be discussed further below.) If m is the number of nonzero entries in A, then each iteration of Sinkhorn can be performed in $O(m)$ time. Therefore the total running time of Algorithm 3 is linear in m so long as the number of iterations depends only on ε' but not on n or m.

Before this work, the best analysis of the RAS method appeared in [KLRS08]. They defined

$$\mathrm{dist}(A, \mathcal{U}_{r,c}) = \|r(A) - r\|_2 + \|c(A) - c\|_2 ,$$

Algorithm 3 $\textsc{Sinkhorn}(A, \mathcal{U}_{r,c}, \varepsilon')$

Initialize $k \leftarrow 0$
$A^{(0)} \leftarrow A/\|A\|_1$, $X^{(0)} \leftarrow I$, $Y^{(0)} \leftarrow I$
while $\mathrm{dist}(A^{(k)}, \mathcal{U}_{r,c}) > \varepsilon$ **do**
  $k \leftarrow k + 1$
  if k odd **then**
    $X \leftarrow$ diagonal with $X_{ii} = \frac{r_i}{r_i(A^{(k-1)})}$
    $X^{(k)} \leftarrow X^{(k-1)}X$, $Y^{(k)} \leftarrow Y^{(k-1)}$
  else
    $Y \leftarrow$ diagonal with $Y_{jj} = \frac{c_j}{c_j(A^{(k-1)})}$
    $Y^{(k)} \leftarrow Y^{(k-1)}Y$, $X^{(k)} \leftarrow X^{(k-1)}$
  $A^{(k)} = X^{(k)}AY^{(k)}$
Output $B \leftarrow A^{(k)}$

and showed that if A is strictly positive with $\min_{ij} A_{ij} = \ell$ and $\sum_{ij} A_{ij} = s$, then Algorithm 3 outputs a matrix B satisfying

$$\|r(B) - r\|_2 + \|c(B) - c\|_2 \le \varepsilon' \tag{5}$$

in $O\big(\rho(\varepsilon')^{-2}\log(s/\ell)\big)$ iterations, where $\rho > 0$ is such that $r_i, c_i \le \rho$ for all $i \in [n]$. While this result appears to show that we can obtain an ε'-approximate scaling of A in only $\widetilde{O}((\varepsilon')^{-2})$ iterations, this impression is misleading. Indeed, the ℓ_2 norm is not an appropriate measure of closeness between probability vectors, since very different distributions on large alphabets can nevertheless have small ℓ_2 distance: for example, $(n^{-1}, \dots, n^{-1}, 0, \dots, 0)$ and $(0, \dots, 0, n^{-1}, \dots, n^{-1})$ in Δ_{2n} have ℓ_2 distance $\sqrt{2/n}$ even though they have disjoint support. As noted above, for statistical problems, including computation of the OT distance, it is more natural to measure distance in ℓ_1 norm.

The best ℓ_1 guarantee available from previous work implies that a matrix B can be obtained satisfying

$$\|r(B) - r\|_1 + \|c(B) - c\|_1 \le \varepsilon'$$

in $O\big(n\rho(\varepsilon')^{-2}\log(s/\ell)\big)$ iterations, where the extra factor of n is the price to pay to convert an ℓ_2 bound to an ℓ_1 bound. Note that $\rho \ge 1/n$, so $n\rho$ is always larger than 1. In the extreme where r or c contains an entry of constant size, $n\rho = \Omega(n)$. However, if $r = c = \mathbf{1}_n/n$ are uniform distributions, then $n\rho = 1$ and no dependence on the dimension appears. Our new analysis allows to a dimension-independent bound on the number of iterations beyond the uniform case.

THEOREM 5. *Algorithm 3 with* $\operatorname{dist}(A, \mathcal{U}_{r,c}) = \|r(A) - r\|_1 + \|c(A) - c\|_1$ *outputs a matrix* B *satisfying* $\operatorname{dist}(B, \mathcal{U}_{r,c}) \le \varepsilon'$ *in* $O\big((\varepsilon')^{-2}\log(s/\ell)\big)$ *iterations.*

Comparing our result with the bound on ℓ_2 distance, we see what our bound is always stronger, by up to a factor of n. Moreover, our analysis is extremely short. Our improved results and simplified proof follow directly from the fact that we carry out the analysis entirely in with respect to the KullbackLeibler divergence, a common measure of statistical distance. This measure possesses a close connection to the total-variation distance via Pinsker's inequality (Lemma 3, below), from which we obtain the desired ℓ_1 bound. A full proof appears in Section 3.1.

We also propose a new algorithm, GREENKHORN (for "Greedy Sinkhorn"), which enjoys precisely the same bound as SINKHORN, but which works better in many practical situations (see Section 4 for experimental results). We emphasize that previous analyses of Sinkhorn iteration did not apply to GREENKHORN, but our new analysis handles the GREENKHORN algorithm with only trivial modifications.

3.1 New analysis of Sinkhorn iteration

We analyze Sinkhorn iterations by considering the following auxiliary function, which has appeared in much of the literature on Sinkhorn projections [KLRS08, CMTV17, KK96, KK93]. Given a matrix A and desired row and column sums r and c, we define $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ by

$$f(x, y) = \sum_{ij} A_{ij} e^{x_i + y_j} - \langle r, x\rangle - \langle c, y\rangle .$$

It is easy to check that a minimizer (x^*, y^*) of f yields the Sinkhorn projection of A: writing $X = \mathbf{D}(\exp(x^*))$ and $Y = \mathbf{D}(\exp(y^*))$, first order optimality conditions imply that XAY lies in $\mathcal{U}_{r,c}$, and therefore $XAY = \Pi_{\mathcal{S}}(A, \mathcal{U}_{r,c})$.

Since the first step of Algorithm 3 renormalizes A to have total mass 1, we can assume in our analysis that $s = 1$ at the price of replacing ℓ by ℓ/s. This simplification is valid because our bound involves the scale-invariant quantity s/ℓ. Likewise, if r_i or c_j is zero for some $i, j \in [n]$, then the corresponding rows and columns of $A^{(k)}$ contain only zeroes throughout the execution of the algorithm. We therefore restrict our attention to the submatrix indexed by positive entries of r and c.

Algorithm 3 updates one of the diagonal matrices $X^{(k)}$ and $Y^{(k)}$ at each step. Write x^k for the vector whose ith entry is $\log X^{(k)}_{ii}$, and similarly let y^k be the vector with entries $\log Y^{(k)}_{jj}$. We call these vectors the *scaling vectors* corresponding to $A^{(k)}$.

The proof of Theorem 5 relies on the following Lemmas that relate the successive improvements of the function f to the Kullback-Leibler divergence between target and current row/column sums. Similar ideas can be traced back at least to [GY98] where an analysis of Sinkhorn iterations for bi-stochastic targets is sketched in the context of a different problem, detecting the existence of a perfect matching in a bipartite graph.

LEMMA 1. *If $k \geq 2$, then*

$$f(x^{k-1}, y^{k-1}) - f(x^k, y^k) = \mathcal{K}(r \| r(A^{(k-1)})) + \mathcal{K}(c \| c(A^{(k-1)}))\,.$$

PROOF. Assume without loss of generality that k is odd, so that $c(A^{(k-1)}) = c$ and $r(A^{(k)}) = r$. (If k is even, interchange the roles of r and c.) By definition,

$$\begin{aligned} f(x^{k-1}, y^{k-1}) - f(x^k, y^k) &= \sum_{ij} \left(A^{(k-1)}_{ij} - A^{(k)}_{ij}\right) + \langle r, x^k - x^{k-1}\rangle + \langle c, y^k - y^{k-1}\rangle \\ &= \sum_i r_i (x^k_i - x^{k-1}_i) \\ &= \mathcal{K}(r \| r(A^{(k-1)}) + \mathcal{K}(c \| c(A^{(k-1)})\,, \end{aligned}$$

where we have used that: $\|A^{(k-1)}\|_1 = \|A^{(k)}\|_1 = 1$ and $Y^{(k)} = Y^{(k-1)}$; for all i,

$$r_i(x^k_i - x^{k-1}_i) = r_i \log \frac{X^{(k)}_{ii}}{X^{(k-1)}_{ii}} = r_i \log \frac{r_i}{r_i(A^{(k-1)})}\,;$$

and $\mathcal{K}(c \| c(A^{(k-1)})) = 0$ since $c = c(A^{(k-1)})$. □

The next lemma has already appeared in the literature and we defer its proof to the Appendix.

LEMMA 2. *If A is a positive matrix with total mass s, then*

$$f(x^1, y^1) - \min_{x,y \in \mathbb{R}} f(x, y) \leq f(0, 0) - \min_{x,y \in \mathbb{R}} f(x, y) \leq \log \frac{s}{\ell}\,.$$

LEMMA 3 (Pinsker's Inequality [Tsy09]). *For any $p, q \in \Delta_{n^2}$ such that p is absolutely continuous with respect to q, we have*

$$\|p - q\|_1 \leq \sqrt{2\mathcal{K}(p\|q)}\,.$$

PROOF OF THEOREM 5. Let k^* be the first iteration such that

$$\|r(A^{(k^*)}) - r\|_1 + \|c(A^{(k^*)}) - c\|_1 \leq \varepsilon\,.$$

Pinsker's inequality implies that for any $k < k^*$, we have

$$\varepsilon^2 < (\|r(A^{(k)}) - r\|_1 + \|c(A^{(k)}) - c\|_1)^2 \leq 4(\mathcal{K}(r\|r(A^{(k)}) + \mathcal{K}(c\|c(A^{(k)}))\,,$$

so Lemmas 1 and 2 implies that we terminate in

$$k^* \leq 4\varepsilon^{-2}\log\left(\frac{s}{\ell}\right)$$

steps, as claimed. □

3.2 Greedy sinkhorn

In addition to a new analysis of SINKHORN, we propose a new algorithm which enjoys the same convergence guarantee but with better performance in practice. Instead of alternating updates of all rows and columns of A, GREENKHORN simply updates the best single row or column at each step, thus updating only $O(n)$ entries of A, rather than $O(n^2)$ per iteration. Our analysis shows GREENKHORN might require n times more iterations than SINKHORN, so that the runtime guarantees of the two algorithms are the same. However, GREENKHORN tends to make much faster progress in practice.

This algorithm is an extremely natural modification of the RAS method, but previous analyses of RAS cannot be modified to extract any meaningful performance guarantees. On the other hand, our new analysis applies to GREENKHORN with only trivial modifications.

Pseudocode for GREENKHORN appears in Algorithm 4. As in SINKHORN,

$$\operatorname{dist}(A, \mathcal{U}_{r,c}) = \|r(A) - r\|_1 + \|c(A) - c\|_1\,.$$

Violations of the row and column constraints are measured by the distance function $\rho : \mathbb{R}_+ \times \mathbb{R}_+ \to [0, +\infty]$ given by

$$\rho(a, b) = b - a + a\log\frac{a}{b}\,.$$

Algorithm 4 GREENKHORN$(A, \mathcal{U}_{r,c}, \varepsilon')$

```
A ← A/‖A‖₁
while dist(A, U_{r,c}) > ε do
    I ← argmax_i ρ(r_i, r_i(A))
    J ← argmax_j ρ(c_j, c_j(A))
    if ρ(r_I, r_I(A)) > ρ(c_J, c_J(A)) then
        Rescale Ith row of A by r_I/r_I(A)
    else
        Rescale Jth row of A by c_J/c_J(A)
Output B ← A
```

Since ρ is not symmetric, it is not a metric; however, the function ρ is nonnegative and satisfies $\rho(a, b) = 0$ iff $a = b$.

We note that after $r(A)$ and $c(A)$ are computed once at the beginning of the algorithm, GREENKHORN can be implemented such that each iteration runs in only $O(n)$ time.

THEOREM 6. *The algorithm* GREENKHORN *yields a matrix satisfying* $\operatorname{dist}(B, \mathcal{U}_{r,c}) \leq \varepsilon'$ *in* $O(n(\varepsilon')^{-2}\log(s/\ell))$ *iterations. Since each iteration requires* $O(n)$ *operations, such a matrix can be found in* $O(n^2(\varepsilon')^{-2}\log(s/\ell))$ *arithmetic operations.*

The analysis requires the following Lemma, which is an easy modification of Lemma 1.

LEMMA 4. *Let* A' *and* A'' *be successive iterates of* GREENKHORN*, with corresponding scaling vectors* (x', y') *and* (x'', y'')*. If* A'' *was obtained from* A' *by updating row* I*, then*

$$f(x', y') - f(x'', y'') = \rho(r_I, r_I(A'))\,,$$

and if it was obtained by updating column J*, then*

$$f(x', y') - f(x'', y'') = \rho(c_J, c_J(A'))\,.$$

We also require the following extension of Pinsker's inequality (proof in Appendix).

LEMMA 5. *For any* $\alpha \in \Delta_n, \beta \in \mathbb{R}^n_+$*, define* $\rho(\alpha, \beta) = \sum_i \rho(\alpha_i, \beta_i)$*. If* $\rho(\alpha, \beta) \leq 1$*, then*

$$\|\alpha - \beta\|_1 \leq \sqrt{7\rho(\alpha, \beta)}\,.$$

PROOF OF THEOREM 6. We follow the proof of Theorem 5. If $\|r(A) - r\|_1 + \|c(A) - c\|_1 > \varepsilon$, then we make at least

$$\frac{1}{2n}(\rho(r, r(A)) + \rho(c, c(A))) \geq \frac{1}{14n}(\|r(A) - r\|_1^2 + \|c(A) - c\|_1^2) \geq \frac{1}{28n}\varepsilon^2$$

progress at each step, so we terminate in at most $28n\varepsilon^{-2}\log(s/\ell)$ iterations. □

4. EMPIRICAL RESULTS

Cuturi [Cut13] already gave experimental evidence that using SINKHORN to solve (2) outperforms state-of-the-art techniques for optimal transport. In this section, we provide strong empirical evidence that our proposed GREENKHORN algorithm significantly outperforms SINKHORN.

We consider transportation between pairs of $m \times m$ greyscale images, normalized to have unit total mass. The target marginals r and c represent two images in a pair, and $C \in \mathbb{R}^{m^2 \times m^2}$ is the matrix of ℓ_1 distances between pixel locations. Therefore, we aim to compute the earth mover's distance.

We run experiments on two datasets: *real images*, from MNIST, and *synthetic images*, as in Figure 2.

4.1 MNIST

We first compare the behavior of GREENKHORN and SINKHORN on real images. To that end, we choose 10 random pairs of images from the MNIST dataset, and for each one analyze the performance of APPROXOT when using both GREENKHORN and SINKHORN for the approximate projection step. We add negligible noise 0.01 to each background pixel with intensity 0. Figure 1 paints a clear picture: GREENKHORN significantly outperforms SINKHORN both in the short and long term.

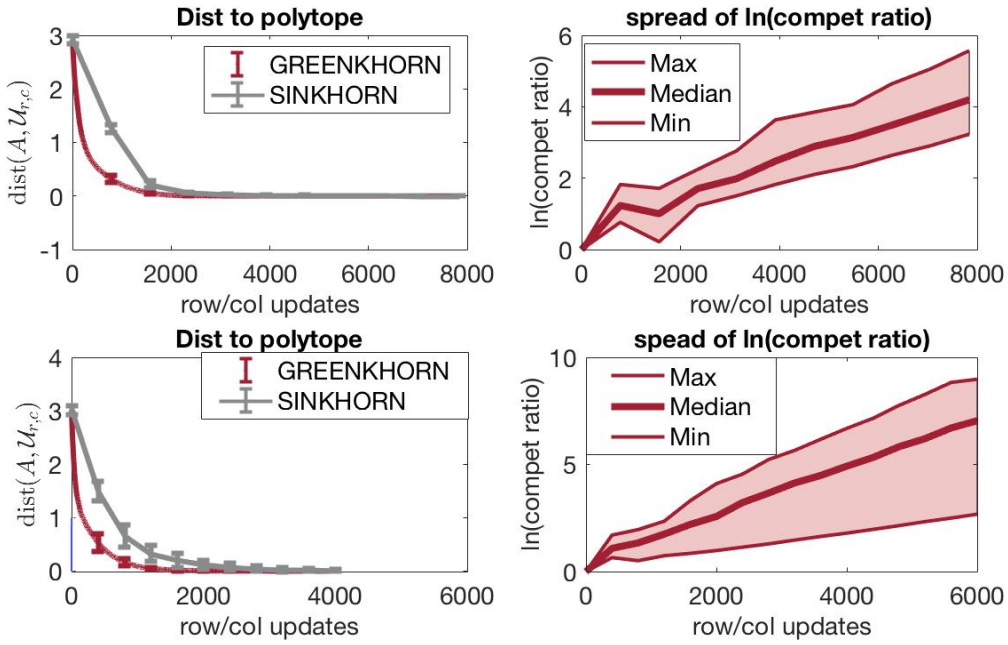

Figure 1: Comparison of GREENKHORN and SINKHORN on pairs of MNIST images of dimension 28×28 (top) and random images of dimension 20×20 with 20% foreground (bottom). Left: distance $\text{dist}(A, \mathcal{U}_{r,c})$ to the transport polytope (average over 10 random pairs of images). Right: maximum, median, and minimum values of the competitive ratio $\ln(\text{dist}(A_S, \mathcal{U}_{r,c})/\text{dist}(A_G, \mathcal{U}_{r,c}))$ over 10 runs.

4.2 Random images

To better understand the empirical behavior of both algorithms in a number of different regimes, we devised a synthetic and tunable framework whereby we generate images by choosing a randomly positioned "foreground" square in an otherwise black background. The size of this square is a tunable parameter varied between 20%, 50%, and 80% of the total image's area. Intensities of background pixels are drawn uniformly from $[0, 1]$; foreground pixels are drawn uniformly from $[0, 50]$. Such an image is depicted in Figure 2, and results appear in Figure 1.

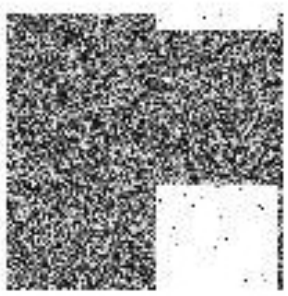

Figure 2: Synthetic image.

We perform two other experiments with random images in Figure 3. In the first, we vary the number of background pixels and show that GREENKHORN performs better when the number of background pixels is larger. We conjecture that this is related to the fact that GREENKHORN only updates salient rows and columns at each step, whereas SINKHORN wastes time updating rows and columns corresponding to background pixels, which have negligible impact. This demonstrates that GREENKHORN is a better choice especially when data is sparse,

which is often the case in practice.

In the second, we consider the role of the regularization parameter η. Our analysis requires taking η of order $\log n/\varepsilon$, but Cuturi [Cut13] observed that in practice η can be much smaller. Cuturi showed that SINKHORN outperforms state-of-the art techniques for computing OT distance even when η is a small constant, and Figure 3 shows that GREENKHORN runs faster than SINKHORN in this regime with no loss in accuracy.

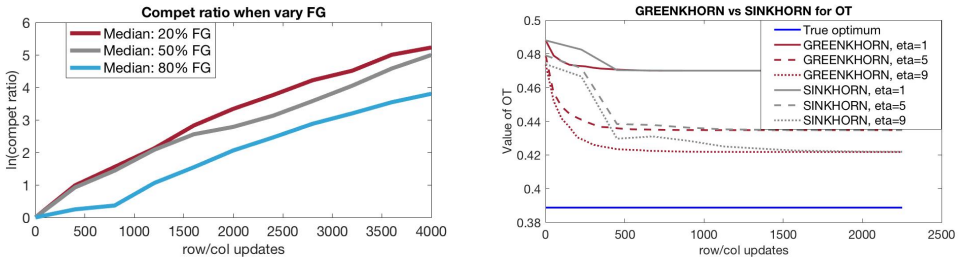

Figure 3: Left: Comparison of median competitive ratio for random images containing 20%, 50%, and 80% foreground. Right: Performance of GREENKHORN and SINKHORN for small values of η.

APPENDIX A: OMITTED PROOFS

A.1 Proof of Theorem 4

Let G be the output of $\text{ROUND}(F, \mathcal{U}_{r,c})$. The entries of F are nonnegative throughout, and at the end of the algorithm err_r and err_c are both nonnegative, with $\|\text{err}_r\|_1 = \|\text{err}_c\|_1 = 1 - \|F\|_1$. Therefore the entries of G are nonnegative and

$$r(G) = r(F) + r(\text{err}_r \text{err}_c^\top / \|\text{err}_r\|_1) = r(F) + \text{err}_r = r\,,$$

and likewise $c(G) = c$. This establishes that $G \in \mathcal{U}_{r,c}$.

Let $\Delta = 1 - \|F\|_1$ be the total amount of mass subtracted from F during the course of the algorithm. Since we only remove mass from F from rows and columns which are over weight, we have

$$\begin{aligned}\Delta &\le \sum_{i=1}^{n} (r(F)_i - r_i)_+ + \sum_{j=1}^{n} (c(F)_j - c_j)_+ \\ &\le \frac{1}{2}(\|r(F) - r\|_1 + \|c(F) - c\|_1)\,.\end{aligned}$$

We obtain

$$\begin{aligned}\|G - F\|_1 &\le \Delta + \|\text{err}_r \text{err}_c^\top\|_1 / \|\text{err}_r\|_1 \\ &= 2\Delta \le \|r(F) - r\|_1 + \|c(F) - c\|_1\,.\end{aligned}$$

Finally, we prove the $O(n^2)$ runtime bound follows by observing that each rescaling and computing the matrix $\text{err}_r \text{err}_c^\top / \|\text{err}_r\|_1$ both require at most $O(n^2)$ time. □

A.2 Proof of Lemma 2

The first inequality follows from the fact that rescaling the rows or columns of A always leads to improvement in the value of f. Then as in the proof of Lemma 1,

$$\begin{aligned} f(x^{(0)}, y^{(0)}) - f(x^{(1)}, y^{(1)}) &= \langle r, x^{(1)} \rangle + \langle c, y^{(1)} \rangle \\ &= \sum_{ij} A^{(1)}_{ij} \log \frac{A^{(1)}_{ij}}{A_{ij}} \\ &= \mathcal{K}(A^{(1)} \| A) \geq 0 \,. \end{aligned}$$

We now prove the second claim. Recall that we assume that we have rescaled A in such a way that $\|A\|_1 = 1$ and its smallest entry is ℓ/s. Since A is positive, [Sin67] shows that $\Pi_{\mathcal{S}}(A)$ exists and is unique. Let (x^*, y^*) be corresponding scaling factors. Then

$$f(x^{(0)}, y^{(0)}) - f(x^*, y^*) = \langle r, x^* \rangle + \langle c, y^* \rangle \,.$$

Since

$$A_{ij} e^{x^*_i + y^*_j} \leq \sum_{ij} A_{ij} e^{x^*_i + y^*_j} = 1 \,,$$

we have

$$x^*_i + y^*_j \leq \log \frac{s}{\ell} \,,$$

for all $i, j \in [n]$. Because r and c are both probability vectors,

$$\langle r, x^* \rangle + \langle c, y^* \rangle \leq \log \frac{s}{\ell} \,.$$

□

A.3 Proof of Lemma 4

We prove only the case where a row was updated, since the column case is exactly the same.

By definition,

$$f(x', y') - f(x'', y'') = \sum_{ij} (A'_{ij} - A''_{ij}) + \langle r, x'' - x' \rangle + \langle c, y'' - y' \rangle \,.$$

Observe that A' and A'' differ only in the Ith row, and x'' and x' differ only in the Ith entry, and $y'' = y'$. Hence

$$\begin{aligned} f(x', y') - f(x'', y'') &= r_I(A') - r_I(A'') + r_I(x''_I - x'_I) \\ &= \rho(r_I, r_I(A')) \,, \end{aligned}$$

where we have used the fact that $r_I(A'') = r_I$ and $x''_I - x'_I = \log(r_I / r_I(A'))$. □

A.4 Proof of Lemma 5

Let $s = \sum_i \beta_i$, and write $\bar{\beta} = \beta/s$. The definition of ρ implies

$$\begin{aligned}\rho(\alpha,\beta) &= \sum_i (\beta_i - \alpha_i) + \alpha_i \log \frac{\alpha_i}{\beta_i} \\ &= s - 1 + \sum_i \alpha_i \log \frac{\alpha_i}{s\bar{\beta}_i} \\ &= s - 1 - (\log s) \sum_i \alpha_i + \mathcal{K}(\alpha \| \bar{\beta}) \\ &= s - 1 - \log s + \mathcal{K}(\alpha \| \bar{\beta}) \,.\end{aligned}$$

Note that both $s - 1 - \log s$ and $\mathcal{K}(\alpha\|\bar{\beta})$ are nonnegative. If $\rho(\alpha,\beta) \le 1$, then in particular $s - 1 - \log s \le 1$, and it can be seen that $s - 1 - \log s \ge (s-1)^2/5$ in this range. Applying Lemma 3 (Pinsker's inequality) yields

$$\rho(\alpha,\beta) \ge \frac{1}{5}(s-1)^2 + \frac{1}{2}\|\alpha - \bar{\beta}\|_1^2 \,.$$

By the triangle inequality and convexity,

$$\begin{aligned}\|\alpha - \beta\|_1^2 &\le (\|\bar{\beta} - \beta\|_1 + \|\alpha - \bar{\beta}\|_1)^2 \\ &= (|s-1| + \|\alpha - \bar{\beta}\|_1)^2 \\ &\le \frac{7}{5}(s-1)^2 + \frac{7}{2}\|\alpha - \bar{\beta}\|_1^2 \,.\end{aligned}$$

The claim follows from the above two displays. □

Acknowledgments

We thank Michael Cohen, Adrian Vladu, John Kelner, and Marco Cuturi for helpful discussions.

REFERENCES

[ACB17] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *ArXiv:1701.07875*, January 2017.

[AZLOW17] Z. Allen-Zhu, Y. Li, R. Oliveira, and A. Wigderson. Much faster algorithms for matrix scaling. *arXiv preprint arXiv:1704.02315*, 2017.

[BCC+15] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[BGKL17] J. Bigot, R. Gouet, T. Klein, and A. López. Geodesic PCA in the Wasserstein space by convex PCA. *Ann. Inst. H. Poincaré Probab. Statist.*, 53(1):1–26, 02 2017.

[Bub15] S. Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3-4):231–357, 2015.

[BvdPPH11] N. Bonneel, M. van de Panne, S. Paris, and W. Heidrich. Displacement interpolation using Lagrangian mass transport. *ACM Trans. Graph.*, 30(6):158:1–158:12, December 2011.

[CBL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, Cambridge, 2006.

[CMTV17] M. B. Cohen, A. Madry, D. Tsipras, and A. Vladu. Matrix scaling and balancing via box constrained Newton's method and interior point methods. *arXiv:1704.02310*, 2017.

[Cut13] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2292–2300. Curran Associates, Inc., 2013.

[FCCR16] R. Flamary, M. Cuturi, N. Courty, and A. Rakotomamonjy. Wasserstein discriminant analysis. *arXiv:1608.08063*, 2016.

[GCPB16] A. Genevay, M. Cuturi, G. Peyré, and F. Bach. Stochastic optimization for large-scale optimal transport. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3440–3448. Curran Associates, Inc., 2016.

[GD04] K. Grauman and T. Darrell. Fast contour matching using approximate earth mover's distance. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–220–I–227 Vol.1, June 2004.

[GPC15] A. Gramfort, G. Peyré, and M. Cuturi. *Fast Optimal Transport Averaging of Neuroimaging Data*, pages 261–272. Springer International Publishing, 2015.

[GY98] L. Gurvits and P. Yianilos. The deflation-inflation method for certain semidefinite programming and maximum determinant completion problems. Technical report, NECI, 1998.

[IT03] P. Indyk and N. Thaper. Fast image retrieval via embeddings. In *Third International Workshop on Statistical and Computational Theories of Vision*, 2003.

[JRT08] A. Juditsky, P. Rigollet, and A. Tsybakov. Learning by mirror averaging. *Ann. Statist.*, 36(5):2183–2206, 2008.

[JSCG16] W. Jitkrittum, Z. Szabó, K. P. Chwialkowski, and A. Gretton. Interpretable distribution features with maximum testing power. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 181–189, 2016.

[KK93] B. Kalantari and L. Khachiyan. On the rate of convergence of deterministic and randomized RAS matrix scaling algorithms. *Oper. Res. Lett.*, 14(5):237–244, 1993.

[KK96] B. Kalantari and L. Khachiyan. On the complexity of nonnegative-matrix scaling. *Linear Algebra Appl.*, 240:87–103, 1996.

[KLRS08] B. Kalantari, I. Lari, F. Ricca, and B. Simeone. On the complexity of general matrix scaling and entropy minimization via the RAS algorithm. *Math. Program.*, 112(2, Ser. A):371–401, 2008.

[Leo14] C. Leonard. A survey of the Schrödinger problem and some of its connections with optimal transport. *Discrete and Continuous Dynamical Systems*, 34(4):1533–1574, 2014.

[LG15] J. R. Lloyd and Z. Ghahramani. Statistical model criticism using kernel two sample tests. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 829–837, Cambridge, MA, USA, 2015. MIT Press.

[LS14] Y. T. Lee and A. Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{\text{rank}})$ iterations and faster algorithms for maximum flow. In *Proceedings of the 2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, FOCS '14, pages 424–433, Washington, DC, USA, 2014. IEEE Computer Society.

[MJ15] J. Mueller and T. Jaakkola. Principal differences analysis: Interpretable characterization of differences between distributions. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 1702–1710, Cambridge, MA, USA, 2015. MIT Press.

[PW09] O. Pele and M. Werman. Fast and robust earth mover's distances. In *2009 IEEE 12th International Conference on Computer Vision*, pages 460–467, Sept 2009.

[PZ16] V. M. Panaretos and Y. Zemel. Amplitude and phase variation of point processes. *Ann. Statist.*, 44(2):771–812, 04 2016.

[Ren88] J. Renegar. A polynomial-time algorithm, based on newton's method, for linear programming. *Mathematical Programming*, 40(1):59–93, 1988.

[RT11] P. Rigollet and A. Tsybakov. Exponential screening and optimal rates of sparse estimation. *Ann. Statist.*, 39(2):731–771, 2011.

[RT12] P. Rigollet and A. Tsybakov. Sparse estimation by exponential weighting. *Statistical Science*, 27(4):558–575, 2012.

[RTG00] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vision*, 40(2):99–121, November 2000.

[Sch31] E. Schrödinger. Über die Umkehrung der Naturgesetze. *Angewandte Chemie*, 44(30):636–636, 1931.

[SdGP+15] 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 Trans. Graph.*, 34(4):66:1–66:11, July 2015.

[Sin67] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967.

[SJ08] S. Shirdhonkar and D. W. Jacobs. Approximate earth mover's distance in linear time. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[SL11] R. Sandler and M. Lindenbaum. Nonnegative matrix factorization with earth mover's distance metric for image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1590–1602, Aug 2011.

[SR04] G. J. Székely and M. L. Rizzo. Testing for equal distributions in high dimension. *Inter-Stat (London)*, 11(5):1–16, 2004.

[ST04] D. A. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 81–90, New York, NY, USA, 2004. ACM.

[Tsy09] A. B. Tsybakov. *Introduction to nonparametric estimation*. Springer Series in Statistics. Springer, New York, 2009. Revised and extended from the 2004 French original, Translated by Vladimir Zaiats.

[Vil09] C. Villani. *Optimal transport*, volume 338 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 2009. Old and new.

[WPR85] M. Werman, S. Peleg, and A. Rosenfeld. A distance metric for multidimensional histograms. *Computer Vision, Graphics, and Image Processing*, 32(3):328 – 336, 1985.