Complexity analysis of Sinkhorn-like algorithms for discrete optimal transport.

Yvann Le Fay

January 2024

Abstract

Inspired by the matching of supply and demand in logistic problems in an optimal way, the optimal transport (OT) problem consists in finding the minimal cost of transporting one distribution a to another b. This problem has recently gained a lot of attention because of its tremendous applications in machine learning for comparing distributions. However, solving the optimal transport problem is difficult. In the discrete case, the target distributions are probability vectors with size n, the OT problem becomes a large (n^2) constrained linear program which can be solved by the simplex method or the interior-point method. However, those methods suffer from an overall cubic cost with respect to n. To mitigate this issue, we instead resort to numerical schemes for solving approximate OT formulations that lead to the optimal cost up to accuracy ε. In particular, the most commonly used approximate formulation is the entropy-regularised optimal transport. Under the entropic regularisation, the problem reduces to computing two dual variables u, v maximising a strictly concave problem, for which there exists an algorithm called the Sinkhorn algorithm and which is well-known to be of complexity $O(n^2 \log(n)/\varepsilon^2)$. In this report, we are interested in two Sinkhorn-like algorithms, a greedy version of the Sinkhorn algorithm, Greekhorn, and the Adaptative primal-dual accelerated mirror descent (APDAMD) algorithm. In particular, we numerically validate the theoretical finding on the upper-bound complexities of those two algorithms. Indeed, we show that the complexity of the Greenkhorn algorithm achieves a complexity of $O(n^2 \log(n)/\varepsilon^2)$, while the APDAMD algorithm achieves a complexity of $O(n^{2.5} \log(n)/\varepsilon)$.

1 Introduction

Let $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^n$ be two histograms and let $\alpha = \sum_{i=1}^n a_i \delta_{x_i}$ and $\beta = \sum_{j=1}^n b_j \delta_{y_j}$ be two discrete measures with support respectively $\{x_1, \ldots, x_n\}$ and $\{y_1, \ldots, y_n\}$ and weights, the a_i's, and the b_j's respectively. Let $C = (C_{i,j})_{1 \leq i,j \leq n}$ be a cost matrix where $C_{i,j} \geq 0$. The discrete OT problem consists in finding a transport plan $X^* \in \mathbb{R}_+^{n \times n}$ such that it minimizes the total cost of transporting α to β, i.e.,

$$X^* \in \operatorname{argmin}_{X \in \mathbb{R}_+^{n \times n}} \langle C, X \rangle \quad \text{such that } X 1_n = a \text{ and } X^\top 1_n = b, \tag{1}$$

where $\langle C, X \rangle = \sum_{1 \leq i,j \leq n} C_{i,j} X_{i,j}$. Problem (1) can be solved using optimisation algorithms for linear programs such as the Network Simplex algorithm or more recently, interior point methods [see, e.g. Boyd and Vandenberghe, 2004, Ch. 11]. However, all those methods are well-known to have an overall cubic complexity with respect to n [Peyré and Cuturi, 2020, Ch. 3], thus making the initial discrete OT problem time-expensive. To mitigate the computational cost of X^*, we instead resort to approximately solving (1), up to a precision $\varepsilon > 0$ by which we mean, finding $\hat{X} \in \mathbb{R}_+^{n \times n}$ satisfying the marginal constraints, such that

$$\langle C, \hat{X} \rangle \leq \langle C, X^* \rangle + \varepsilon. \tag{2}$$

Such a transport plan $\hat{X}$ is called a *ε-approximate transport plan.* We typically proceed by adding to the initial objective, a regularisation term, thus making the objective strongly concave. The regularised formulation we are interested in this paper is the entropy-regularized formulation:

$$\hat{X} \in \operatorname{argmin}_{X \in \mathbb{R}_+^{n \times n}} \langle C, X \rangle - \eta H(X) \quad \text{such that } X 1_n = a \text{ and } X^\top 1_n = b, \tag{3}$$

where $H(X) = -\sum_{i,j} X_{i,j}(\log(X_{i,j}) - 1)$ is the entropy penalty and $\eta > 0$ is the penalisation parameter which crucially depends upon ε. The first-order condition on the potentials u and v derived from the Lagrangian of (3) can be expressed as a fixed-point equation which can be iterated. This gives rise to the so-called Sinkhorn Algorithm [Cuturi, 2013, Altschuler et al., 2017] which has been shown to converge [Franklin and Lorenz, 1989].

Previous related work Complexity analysis of the Sinkhorn algorithm has been conducted by Altschuler et al. [2017], leading to the bound of $O(n^2 \log(n)/\varepsilon^3)$ arithmetic operations to compute a ε-approximate transport plan. This bound was later refined to $O(n^2 \log(n)/\varepsilon^2)$ in Dvurechensky et al. [2018]. Furthermore, Dvurechensky et al. [2018] derived a new algorithm called *Alternative primal-dual accelerated gradient descent* (APDAGD) which has been shown to exhibit a complexity $O(n^{2.5}\sqrt{\log(n)}/\varepsilon)$ [Lin et al., 2019], thus, paving the way for algorithms with substantially better performance with respect to ε, i.e., $O(1/\varepsilon)$-rate.

Contributions The rest of the report is organized as follows:

1. In section 2, we review a greedy version of the Sinkhorn algorithm, namely, the Greenkhorn Algorithm [Altschuler et al., 2017]. We focus on the recent work of Lin et al. [2019, 2022] which shows that the Greenkhorn algorithm matches the best-known complexity bound of the Sinkhorn algorithm.

2. In section 3, we review the *Adaptive Primal-Dual Accelerated Mirror Descent* (APDAMD) algorithm developed by Lin et al. [2019, 2022]. This algorithm is a generalisation of the *Adaptive Primal-Dual Accelerated Gradient Descent* [Dvurechensky et al., 2018, APDAGD] which is known to exhibit a $O(1/\varepsilon)$-complexity rate.

3. In section 4, we numerically assess on a simple unidimensional Gaussian case, the theoretical complexity bounds derived by Lin et al. [2019, 2022] for the Grinkhorn and APDAMD algorithms, with respect to both n and ε.

2 Greenkhorn algorithm

The Sinkhorn Algorithm [Cuturi, 2013] is an iterative algorithm to approximate the Sinkhorn projection matrix from which, we can recover a transport plan. At each iteration, it performs a modification of all the rows or columns, thus leading to $O(n^2)$ operations. The Greenkhorn algorithm introduced in Altschuler et al. [2017] has a modified iteration step such that only one row or column is modified at each iteration, which is done in $O(n)$-operations, and other needed quantities depending on the projection matrix can be computed in $O(1)$ operations. The following complexity bound has been derived:

Theorem 2.1 (Th. 12 [Lin et al., 2022])**.** *The Greenkhorn algorithm outputs a ε-approximate transport plan in $O(n^2 \log(n)/\varepsilon^2)$ operations.*

This bound which is essentially the same complexity as the Sinkhorn Algorithm [Dvurechensky et al., 2018], shows that it is not clear that performing only one row or column at each iteration, asymptotically improve the performance of the Greenkhorn algorithm. Indeed, one Greenkhorn update is more expensive than one Sinkhorn update, as Sinkhorn updates all rows or all columns simultaneously, whereas Greenkhorn needs to keep track of which row and which column to update next.

3 APDAMD algorithm

The objective (3) can be cast as a more general constrained linear program:

$$\min f(x) \quad \text{s.t } Ax = b, \tag{4}$$

by letting $x = \text{vec}(X)$, $c = \text{vec}(C)$, A implicitly defined by $Ax = \begin{pmatrix} X1_n \\ X^\top 1_n \end{pmatrix}$ and $b = \begin{pmatrix} r \\ c \end{pmatrix}$ and with objective function f defined by

$$f(x) = c^\top x - \eta H(x), \tag{5}$$

where H is the entropy regularisation. Then objective (3) is equivalent to (4). Let λ be the dual variable, the dual problem is given by

$$\min_{\lambda \in \mathbb{R}^{2n}} \left(\varphi(\lambda) \coloneqq \langle \lambda, b \rangle - \min_{x \in \mathbb{R}^{n^2}} \left(f(x) + \langle A^\top \lambda, x \rangle \right) \right). \tag{6}$$

Solving (6) can be performed through a gradient descent (GD), leading to the APDAGD [Dvurechensky et al., 2018]. A generalisation consists in replacing the gradient descent by a mirror descent with penalty, the Bregman divergence B_ϕ, for some potential convex function ϕ. This gives rise to the APDAMD algorithm [Lin et al., 2019] for solving (3). We explicit an instance of the APDAMD algorithm 1 incorporating a line search algorithm 2 with $\phi(\lambda) = \frac{1}{2n}||\lambda||^2$. The dual objective φ satisfies

$$\varphi(\lambda) = \langle \lambda, b \rangle - \langle A^\top \lambda, x(\lambda) \rangle - f(x(\lambda)), \tag{7}$$

where $x(\lambda) = \text{argmin}_{x \in \mathbb{R}^{n^2}} f(x) + \langle A^\top \lambda, x \rangle = \text{diag}(e^{-u/\eta}) e^{-C/\eta} \text{diag}(e^{-v/\eta})$ with λ being the concatenation of u and v.

Theorem 3.1 (Th. 17 [Lin et al., 2022]). *The APDAMD Algorithm 1 outputs a ε-approximate transport plan in $O(n^{2.5} \log(n)/\varepsilon)$ operations.*

This bound is an improvement by one order of magnitude on ε, however, the asymptotical performance with respect to n is expected to be worse by a factor $\sqrt{n}$.

4 Experiments

In this section, we numerically assess the complexity bounds 2.1 and 3.1 for the Greenkhorn and APDAMD algorithms with respect to both ε and n. The JAX implementation to reproduce the experiments is available at `https://github.com/ylefay/greenkhorn_apdamd`. It includes:

- the Sinkhorn algorithm, the Greenkhorn algorithm as well as the APDAMD algorithm with $\phi(\lambda) = \frac{1}{2n}||\lambda||^2$.
- Creation of multidimensional Gaussian histograms and the Euclidean-Wasserstein distances between Gaussians.

Algorithm 1: APDAMD(C, ε, r, c)

$\eta \leftarrow \varepsilon/(4\log(n))$
$\varepsilon' \leftarrow \varepsilon/(8||C||_\infty)$
$r \leftarrow (1-\varepsilon'/8)r + \varepsilon'/(8n)1_n$ // relaxing the constraints

$c \leftarrow (1-\varepsilon'/8)c + \varepsilon'/(8n)1_n$
$L \leftarrow 1; x, z, \lambda \leftarrow 0_{2n}; \bar{\alpha} = 0$
while $||X1_n - r||_1 + ||X^\top 1_n - c|| \geq \varepsilon'$ **do**
  $M \leftarrow L/2$
  $M, z, \lambda, \bar{\alpha}, \bar{\alpha}_1, \alpha \leftarrow \text{linesearch}(M, z, \lambda, \bar{\alpha})$
  $L \leftarrow M/2$
  $u, v \leftarrow \lambda[:n], \lambda[n:]$ // recover dual solution

  $x \leftarrow (\alpha \text{Diag}(e^{-u/\eta}) e^{-C/\eta} \text{Diag}(e^{-v/\eta}) + \bar{\alpha}x)/\bar{\alpha}_1$
  $\bar{\alpha} \leftarrow \bar{\alpha}_1$
  reshape x to obtain $\hat{X}$
end
// $\hat{X}$ approximately satisfy the constraints with the relaxed r and c
Apply algorithm 2 [Dvurechensky et al., 2018] to obtain X from $\hat{X}$
  // X approximately satisfy the initial constraints

return X

Algorithm 2: linesearch$(M, z, \lambda, \bar{\alpha})$

while $\varphi(\lambda) - \varphi(\mu) - (\lambda - \mu)\nabla\varphi(\mu) \leq M/2||\lambda - \mu||_\infty^2$ **do**
  $M \leftarrow 2M$
  $\alpha \leftarrow (1 + \sqrt{1 + 4nM\bar{\alpha}})/(2nM)$
  $\bar{\alpha}_1 \leftarrow \bar{\alpha} + \alpha$
  $\mu \leftarrow (\alpha z + \bar{\alpha}\lambda)/\bar{\alpha}_1$
  $\nabla\varphi(\mu) \leftarrow b - Ax(\mu)$ // explicit gradient of the dual function

  $z \leftarrow z - 2n\alpha\nabla\varphi(\mu)$ // explicit minimum

  $\lambda \leftarrow (\alpha z + \bar{\alpha}\lambda)/\bar{\alpha}_1$
end
return $M, z, \lambda, \bar{\alpha}, \bar{\alpha}_1, \alpha$

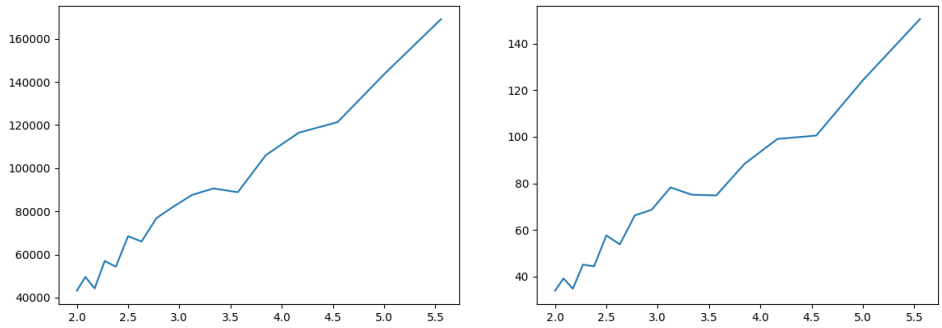

Figure 1: APDAMD. Left: number of iterations with respect to $1/\varepsilon$. Right: running time with respect to $1/\varepsilon$ (10 replications).

4.1 Methodology

For the sake of computational cost, we restrict ourselves to Gaussian distributions in low-dimension. We evaluate the time performance with respect to both ε and n using two proxies:

- The needed number of iterations for the algorithms to terminate.
- The effective running time. We replicate 10 times a run and compute the mean running time.

The target distributions r and c are unidimensional discrete histograms from normal distributions denoted by $\mathcal{N}(m_1, \sigma_1^2)$ and $\mathcal{N}(m_2, \sigma_2^2)$. When comparing for different ε, the sample size is set to $n = 100$, and the n-points for the histograms are taken uniformly in the regions of interest $[m_i - 2\sigma_i; m_i + 2\sigma_i]$ for $i \in \{1, 2\}$. When comparing for different n, ε is set to 0.5. The considered target distributions are $\mathcal{N}(0, 1)$ and $\mathcal{N}(1, 1.5)$. The cost matrix is the Euclidean cost matrix. In this special setup, there is a closed formula for the Wasserstein distance between two Gaussians given by Peyré and Cuturi [2020, Eq. 2.41]. We compare this distance to the unpenalised distance realised by the approximate transport plans.

4.2 Results

For the APDAMD algorithm, Figures 1 and 2 confirm the theoretical complexity given in 3.1. Indeed, with respect to ε, Figure 1 shows that we recover the $O(1/\varepsilon)$-rate. With respect to n, Figure 2 validates that the number of iterations is $O(n^{1.5}\log(n))$ while the number of arithmetic operations, which is expected to be proportional to the running time, is $O(n^{2.5}\log(n))$. For the Greenkhorn algorithm, Figure 3 indicates that the $O(1/\varepsilon^2)$ complexity rate might not be optimal in this case. Figure 4 validates the complexity bound

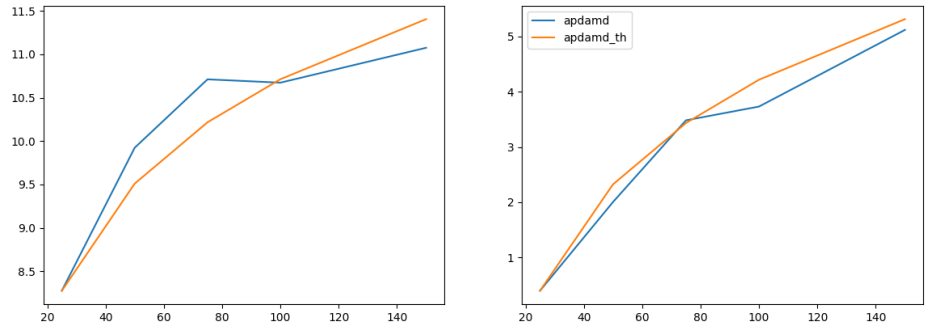

Figure 2: APDAMD. Left: Logarithm of the number of iterations with respect to n. Right: Logarithm of the running time with respect to n (10 replications). Fitted bounds in orange, left: $O(\log(n)n^{1.5})$, right: $O(\log(n)n^{2.5})$.

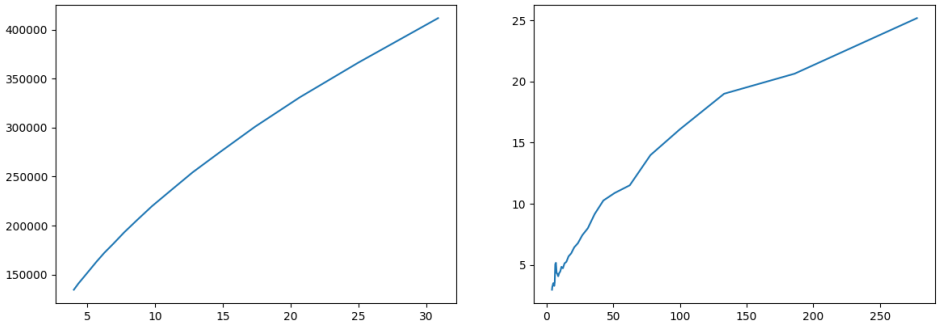

Figure 3: Greenkhorn. Left: number of iterations with respect to $1/\varepsilon^2$. Right: running time with respect to $1/\varepsilon^2$ (10 replications).

of $O(\log(n)n^2)$. On our toy model, the Greenkhorn algorithm outperforms the APDAMD and the Sinkhorn algorithm outperforms the Greenkhorn algorithm, see Table 1. This indicates that, while the APDAMD has a better asymptotic complexity bound with respect to ε, in practice, the Sinkhorn or Greenkhorn algorithms might be more suited. Moreover, with respect to n, the Greenkhorn and Sinkhorn algorithm are expected to beat the APDAMD, see Table 2 for time performances with respect to n. See Figures 3 and 4 for the non-penalised OT distances of the computed transport plans.

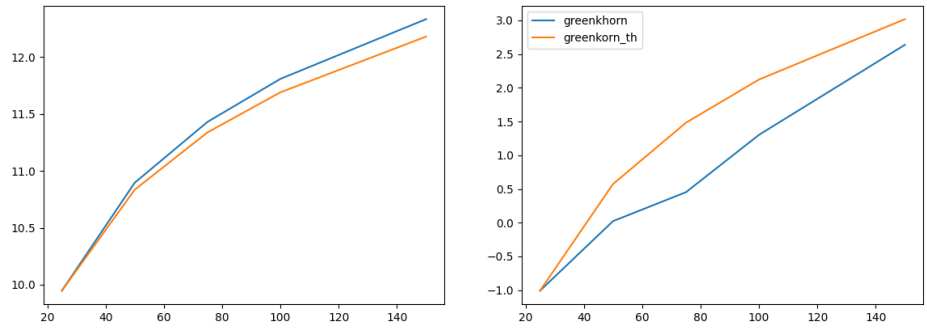

Figure 4: Greenkhorn. Left: Logarithm of the number of iterations with respect to n. Right: Logarithm of the running time with respect to n (10 replications). Fitted bounds in orange, left: $O(\log(n)n^1)$, right: $O(\log(n)n^2)$.

5 Discussion

In this report, we reviewed two algorithms for solving the entropy-regularised optimal transport. We assessed the complexity bounds with respect both to the size of the histograms n and the accuracy parameter ε. On our toy-model, we found that Grinkhorn's time performance is indistinguishable or worse from Sinkhorn's in practice. Furthermore, the overall time performance of the APDAMD is found to be worse compared to the previous two algorithms.

While we restricted ourselves to the potential function $\phi(\lambda) = \frac{1}{2n}||\lambda||^2$, it could be of interest to study the consequences on the convergence speed of different instances of the APDAMD algorithm depending on the chosen potential function ϕ. Indeed, Lin et al. [2019] predict the number of iterations needed for the APDAMD algorithm to converge depends on the module of the Bregman divergence of ϕ. More importantly, we did not delve into the question of parallelisation. Future work could include measuring the running time of those algorithms when using parallelised operations (such as matrix products) on GPUs. Besides, we have not analysed the violation constraints over the different iterations, which could be of interest when comparing convergence speeds. Finally, one could be interested in the running time with respect to the cost matrix C or the target distributions (e.g., the dimension of the latent spaces).

Remark on the main paper The initial paper Lin et al. [2019] contains several typos. The considered penalisation is $H(x) = -\sum_{i,j} X_{i,j} \log(X_{i,j})$ which is not the classical entropy regularisation and leads to a supplementary factor e^{-1} inside the dual function. In the APDAMD section, the definition of $x(\lambda)$ is a minimisation program over $\mathbb{R}^n$ instead of $\mathbb{R}^{n^2}$. The more recent

ε	Sinkhorn	Greenkhorn	APDAMD	Sinkhorn	Greenkhorn	APDAMD
0.18	2435	411741	169142	0.26	7.9	150
0.20	2172	366485	143618	0.23	6.5	124
0.22	1959	330904	121370	0.22	6.4	100
0.24	1782	300877	116475	0.23	6.1	99
0.26	1634	275139	106113	0.20	5.0	88
0.28	1508	254278	88876	0.25	5.1	75
0.30	1399	235387	90620	0.20	4.6	75
0.32	1304	219813	87628	0.19	4.2	78
0.34	1221	205450	82136	0.20	4.0	69
0.36	1147	193111	76897	0.19	3.8	66
0.38	1082	181637	66019	0.18	3.7	54
0.40	1023	172222	68472	0.18	3.5	58
0.42	970	163081	54328	0.19	3.4	44
0.44	922	154623	56972	0.17	3.0	45
0.46	879	147283	44233	0.17	2.9	35
0.48	839	140887	49570	0.17	2.8	39
0.50	802	134494	43180	0.17	2.7	34

Table 1: Three first columns: Number of iterations. The three last columns: Mean running time in seconds over 10 replications.

paper Lin et al. [2022] solves those issues. Moreover, we were not able to draw a clear conclusion on the advantage to use the Grinkhorn algorithm over the Sinkhorn algorithm in the simple unidimensional Gaussian case. The practical performance of each algorithms depends on the considered problem at fixed ε and n, this is an issue which is not treated by Lin et al. [2022] nor Lin et al. [2019]. It could be of interest, for assessing the practical advantage of those algorithms, to repeat the experiments of the two papers but plotting the constraint violations with respect to time, instead of only the number of row/column updates.

Connexion with the course The Greenkhorn algorithm is a direct modification of the Sinkhorn for the regularised-optimal transport, which was studied in Lectures 4, 5, and 6. It leverages the dual formulation for OT which was studied in Lectures 4 and 6. Bregman divergences used in the APDAMD algorithm were introduced in Lecture 5 for generalising the Kullback-Leibler divergence.

n	Sinkhorn	Greenkhorn	APDAMD	Sinkhorn	Greenkhorn	APDAMD
25	502	20885	3925	0.2	0.4	1.5
50	644	53969	20395	0.2	1.0	7.4
75	734	92173	44871	0.2	1.6	32.5
100	802	134494	43180	0.25	3.7	41.8
150	903	227424	64552	0.3	14.0	167.6

Table 2: Fixed $\varepsilon = 0.5$. Three first columns: Number of iterations. The three last columns: Mean running time in seconds over 10 replications.

References

J. Altschuler, J. Niles-Weed, and P. Rigollet. Near-linear time approximation algorithms for optimal transport via sinkhorn iteration. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2, 3

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 2

M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/af21d0c97db2e27e13572cbf59eb343d-Paper.pdf. 2, 3

P. Dvurechensky, A. Gasnikov, and A. Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1367–1376. PMLR, 10–15 Jul 2018. 2, 3, 4, 5

J. Franklin and J. Lorenz. On the scaling of multidimensional matrices. *Linear Algebra and its Applications*, 114-115:717–735, 1989. ISSN 0024-3795. doi: https://doi.org/10.1016/0024-3795(89)90490-4. Special Issue Dedicated to Alan J. Hoffman. 2

T. Lin, N. Ho, and M. Jordan. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 3982–3991. PMLR, 09–15 Jun 2019. 2, 3, 4, 8, 9

ε	Sinkhorn	Greenkhorn	APDAMD
0.18	1.045	1.044	1.046
0.20	1.045	1.045	1.046
0.22	1.046	1.046	1.047
0.24	1.046	1.046	1.048
0.26	1.047	1.047	1.048
0.28	1.047	1.047	1.049
0.30	1.048	1.048	1.049
0.32	1.049	1.048	1.050
0.34	1.05	1.049	1.050
0.36	1.05	1.049	1.051
0.38	1.051	1.050	1.052
0.40	1.051	1.051	1.054
0.42	1.051	1.051	1.053
0.44	1.052	1.051	1.055
0.46	1.052	1.052	1.056
0.48	1.053	1.052	1.055
0.50	1.054	1.053	1.055

Table 3: Realised unpenalised distance $\langle C, \hat{X} \rangle$. The Wasserstein distance is $W_2^2(\mathcal{N}(0,1), \mathcal{N}(0,1.5)) \approx 1.051$.

T. Lin, N. Ho, and M. I. Jordan. On the efficiency of entropic regularized algorithms for optimal transport. *Journal of Machine Learning Research*, 23(137):1–42, 2022. 3, 4, 9

G. Peyré and M. Cuturi. Computational optimal transport, 2020. 2, 6

ε	Sinkhorn	Greenkhorn	APDAMD
25	1.060	1.060	1.061
50	1.0576	1.056	1.058
75	1.055	1.054	1.056
100	1.054	1.053	1.055
150	1.052	1.052	1.057

Table 4: Fixed $\varepsilon = 0.5$. Realised unpenalised distance $\langle C, \hat{X} \rangle$. The Wasserstein distance is $W_2^2(\mathcal{N}(0,1), \mathcal{N}(0,1.5)) \approx 1.051$.