A SINKHORN–NEWTON METHOD FOR ENTROPIC OPTIMAL TRANSPORT

Christoph Brauer* Christian Clason† Dirk Lorenz* Benedikt Wirth‡

February 2, 2018

Abstract We consider the entropic regularization of discretized optimal transport and propose to solve its optimality conditions via a logarithmic Newton iteration. We show a quadratic convergence rate and validate numerically that the method compares favorably with the more commonly used Sinkhorn–Knopp algorithm for small regularization strength. We further investigate numerically the robustness of the proposed method with respect to parameters such as the mesh size of the discretization.

1 INTRODUCTION

The mathematical problem of optimal mass transport has a long history dating back to its introduction in Monge [10], with key contributions by Kantorovivc [6] and Kantorovivc & Rubinvsteuin [7]. It has recently received increased interest due to numerous applications in machine learning; see, e.g., the recent overview of Kolouri, Park, Thorpe, Slepcev & Rohde [9] and the references therein. In a nutshell, the (discrete) problem of optimal transport in its Kantorovich form is to compute for given mass distributions a and b with equal mass a transport plan, i.e., an assignment of how much mass of a at some point should be moved to another point to match the mass in b. This should be done in a way such that some transport cost (usually proportional to the amount of mass and dependent on the distance) is minimized. This leads to a linear optimization problem which has been well studied, but its application in machine learning has been problematic due to large memory requirement and long run time. Recently, Cuturi [2] proposed a method that overcomes the memory requirement by so-called entropic regularization that has found broad applications; see, e.g., Carlier, Duval, Peyré & Schmitzer [1], Cuturi & Doucet [3], and Frogner, Zhang, Mobahi, Araya & Poggio [5]. The resulting iteration resembles the so-called Sinkhorn–Knopp method from Sinkhorn & Knopp [11] for matrix balancing and allows for a simple and efficient implementation.

*Institute of Analysis and Algebra, TU Braunschweig, 38092 Braunschweig, Germany (ch.brauer@tu-braunschweig.de, d.lorenz@tu-braunschweig.de)

†Faculty of Mathematics, University Duisburg-Essen, 45117 Essen, Germany (christian.clason@uni-due.de)

‡Institute for Numerical and Applied Mathematics, University of Münster, Einsteinstraße 62, 48149 Münster, Germany (benedikt.wirth@uni-muenster.de)

1.1 OUR CONTRIBUTION

In this work, we show that the Sinkhorn–Knopp method can be viewed as an approximate Newton method and derive a full Newton method for entropically regularized optimal transport problems that is demonstrated to perform significantly better for small entropic regularization parameters. Here, compared to Cuturi [2], the key idea is to apply a logarithmic transform to the variables.

This paper is organized as follows. In Section 2, we state the Kantorovich formulation of optimal transport together with its dual which serves as the basis of the derived algorithm. Afterwards, we establish local quadratic convergence and discuss the relation of the proposed Newton method to the Sinkhorn–Knopp iteration. The performance and parameter dependence of the proposed method are illustrated with numerical examples in Section 3. Section 4 contains the proof of the key estimate for quadratic convergence, and Section 5 concludes the paper.

1.2 NOTATION

In the following, $\mathbb{1}_n$ represents the n-dimensional vector with all ones and $\mathbb{1}_{n,m}$ refers to the $n \times m$ matrix with all ones. Moreover, $\Sigma_n := \{a \in \mathbb{R}^n_+ : \mathbb{1}_n^\top a = 1\}$ denotes the probability simplex in $\mathbb{R}^n_+$ whose elements are called *probability vectors*, or equivalently, *histograms*. For two histograms $a \in \Sigma_n$ and $b \in \Sigma_m$,

$$U(a,b) := \{P \in \mathbb{R}^{n\times m}_+ : P\mathbb{1}_m = a,\ P^\top \mathbb{1}_n = b\} \tag{1.1}$$

is the set of admissible *coupling matrices*. In the context of optimal transport, the elements of $U(a,b)$ are also referred to as *transport plans*. Histograms a and b can be viewed as mass distributions, and an entry P_{ij} of a transport plan $P \in U(a,b)$ can be interpreted as the amount of mass moved from a_i to b_j.

We refer to the Frobenius inner product of two matrices $P, P' \in \mathbb{R}^{n\times m}$ as $\langle P, P'\rangle := \sum_{ij} P_{ij}P'_{ij}$. At the same time, $\langle a, a'\rangle := \sum_i a_i a'_i$ denotes the standard dot product of two vectors $a, a' \in \mathbb{R}^n$. Finally, $\mathrm{Diag}(a) \in \mathbb{R}^{n\times n}$ is defined as the diagonal matrix with $\mathrm{Diag}(a)_{ii} := a_i$ and $\mathrm{Diag}(a)_{ij} := 0$ for $i \neq j$, and $a \odot a' := \mathrm{Diag}(a)a'$ is the Hadamard product (i.e., the component-wise product) of a and a'.

2 SINKHORN–NEWTON METHOD

In this section we derive our Sinkhorn–Newton method. We start by introducing the problem of entropically regularized optimal transport in Section 2.1. Afterwards, in Section 2.2, we present our approach, which is essentially applying Newton's method to the optimality system associated with the transport problem and its dual, before we discuss its local quadratic convergence in Section 2.3. In Section 2.4, we finally establish a connection between our Newton iteration and the Sinkhorn–Knopp type iteration introduced by Cuturi [2].

2.1 PROBLEM SETTING

Let $a \in \Sigma_n$ and $b \in \Sigma_m$ be given histograms together with a non-negative cost matrix $C \in \mathbb{R}^{n \times m}$. The entropically regularized Kantorovich problem of optimal mass transport between a and b is

$$\inf_{P \in U(a,b)} \langle C, P \rangle + \varepsilon \langle P, \log P - \mathbb{1}_{n,m} \rangle, \tag{P_ε}$$

where the logarithm is applied componentwise to P and $\varepsilon > 0$ is the regularization strength. The variables P_{ij} indicate how much of a_i ends up in b_j, while C_{ij} is the corresponding transport cost per unit mass. Abbreviating $K \coloneqq \exp(-C/\varepsilon)$, standard convex duality theory leads us to the dual problem

$$\sup_{f \in \mathbb{R}^n, g \in \mathbb{R}^m} -\langle a, f \rangle - \langle b, g \rangle - \varepsilon \langle \mathrm{e}^{-f/\varepsilon}, K \mathrm{e}^{-g/\varepsilon} \rangle, \tag{D_ε}$$

where f and g are the dual variables and the exponential function is applied componentwise. The problems (P_ε) and (D_ε) are linked via the optimality conditions

$$P = \operatorname{Diag}(\mathrm{e}^{-f/\varepsilon}) K \operatorname{Diag}(\mathrm{e}^{-g/\varepsilon}) \tag{2.1a}$$

$$a = \operatorname{Diag}(\mathrm{e}^{-f/\varepsilon}) K \mathrm{e}^{-g/\varepsilon} \tag{2.1b}$$

$$b = \operatorname{Diag}(\mathrm{e}^{-g/\varepsilon}) K^\top \mathrm{e}^{-f/\varepsilon}. \tag{2.1c}$$

The first condition (2.1a) connects the optimal transport plan with the dual variables. The conditions (2.1b) and (2.1c) simply reflect the feasibility of P for (P_ε), i.e., for the mass conservation constraints in (1.1).

2.2 ALGORITHM

Finding dual vectors f and g that satisfy (2.1b) and (2.1c) is equivalent to finding a root of the function

$$F(f, g) \coloneqq \begin{pmatrix} a - \operatorname{Diag}(\mathrm{e}^{-f/\varepsilon}) K \mathrm{e}^{-g/\varepsilon} \\ b - \operatorname{Diag}(\mathrm{e}^{-g/\varepsilon}) K^\top \mathrm{e}^{-f/\varepsilon} \end{pmatrix}, \tag{2.2}$$

i.e., to solving $F(f, g) = 0$. A Newton iteration for this equation is given by

$$\begin{pmatrix} f^{k+1} \\ g^{k+1} \end{pmatrix} = \begin{pmatrix} f^k \\ g^k \end{pmatrix} - J_F(f^k, g^k)^{-1} F(f^k, g^k). \tag{2.3}$$

The Jacobian matrix of F is

$$J_F(f, g) = \frac{1}{\varepsilon} \begin{bmatrix} \operatorname{Diag}(P \mathbb{1}_m) & P \\ P^\top & \operatorname{Diag}(P^\top \mathbb{1}_n) \end{bmatrix}, \tag{2.4}$$

where we used (2.1a) to simplify the notation. Performing the Newton step (2.3) requires finding a solution of the linear equation system

$$J_F(f^k, g^k) \begin{pmatrix} \delta f \\ \delta g \end{pmatrix} = -F(f^k, g^k). \tag{2.5}$$

Algorithm 1 Sinkhorn-Newton method in primal variable

1: **Input:** $a \in \Sigma_n$, $b \in \Sigma_m$, $C \in \mathbb{R}^{n \times m}$
2: **Initialize:** $P^0 = \exp(-C/\varepsilon)$, set $k = 0$
3: **repeat**
4: Compute approximate histograms

$$a^k = P^k \mathbb{1}_m, \quad b^k = (P^k)^\top \mathbb{1}_n.$$

5: Compute updates δf and δg by solving

$$\frac{1}{\varepsilon} \begin{bmatrix} \operatorname{Diag}(a^k) & P^k \\ (P^k)^\top & \operatorname{Diag}(b^k) \end{bmatrix} \begin{bmatrix} \delta f \\ \delta g \end{bmatrix} = \begin{bmatrix} a^k - a \\ b^k - b \end{bmatrix}.$$

6: Update P by

$$P^{k+1} = \operatorname{Diag}(\mathrm{e}^{-\delta f/\varepsilon}) P^k \operatorname{Diag}(\mathrm{e}^{-\delta g/\varepsilon}).$$

7: $k \leftarrow k + 1$
8: **until** some stopping criteria fulfilled

The new iterates are then given by

$$f^{k+1} = f^k + \delta f \tag{2.6a}$$
$$g^{k+1} = g^k + \delta g. \tag{2.6b}$$

If one is only interested in the optimal transport plan, then it is actually not necessary to keep track of the dual iterates f^k and g^k after initialization (in our subsequent experiments, we use $f^0 = g^0 = 0$ and hence, $P^0 = K$). This is true because (2.5) can be expressed entirely in terms of

$$P^k := \operatorname{Diag}(\mathrm{e}^{-f^k/\varepsilon}) K \operatorname{Diag}(\mathrm{e}^{-g^k/\varepsilon}), \tag{2.7}$$

and thus, using (2.6) and (2.7), we obtain the multiplicative update rule

$$\begin{aligned} P^{k+1} &= \operatorname{Diag}(\mathrm{e}^{-[f^k+\delta f]/\varepsilon}) K \operatorname{Diag}(\mathrm{e}^{-[g^k+\delta g]/\varepsilon}) \\ &= \operatorname{Diag}(\mathrm{e}^{-\delta f/\varepsilon}) P^k \operatorname{Diag}(\mathrm{e}^{-\delta g/\varepsilon}). \end{aligned} \tag{2.8}$$

In this way, we obtain an algorithm which only operates with primal variables, see Algorithm 1. In applications where the storage demand for the plans P^k is too high and one is only interested in the optimal value, there is another form which does not form the plans P^k, but only the dual variables f^k and g^k and which can basically operate matrix-free. We sketch it as Algorithm 2 below.

Algorithm 2 Sinkhorn-Newton method in dual variables

1: **Input:** $a \in \Sigma_n$, $b \in \Sigma_m$, function handle for application of K and $K^\top$
2: **Initialize:** $a^0 \in \mathbb{R}^n$, $b^0 \in \mathbb{R}^m$, set $k = 0$
3: **repeat**
4: Compute approximate histograms

$$a^k = \mathrm{e}^{-f^k/\varepsilon} \odot K \mathrm{e}^{-g^k/\varepsilon}, \quad b^k = \mathrm{e}^{-g^k/\varepsilon} \odot K^\top \mathrm{e}^{-f^k/\varepsilon}.$$

5: Compute updates δf and δg by solving

$$M \begin{bmatrix} \delta f \\ \delta g \end{bmatrix} = \begin{bmatrix} a^k - a \\ b^k - b \end{bmatrix}$$

where the application of M is given by

$$M \begin{bmatrix} \delta f \\ \delta g \end{bmatrix} = \frac{1}{\varepsilon} \begin{bmatrix} a^k \odot \delta f + \mathrm{e}^{-f^k/\varepsilon} \odot K(\mathrm{e}^{-g^k/\varepsilon} \odot \delta g) \\ b^k \odot \delta g + e^{-g^k/\varepsilon} \odot K^\top(\mathrm{e}^{-f^k/\varepsilon} \odot \delta f) \end{bmatrix}.$$

6: Update f and g by

$$f^{k+1} = f^k + \delta f, \quad g^{k+1} = g^k + \delta g.$$

7: $k \leftarrow k + 1$
8: **until** some stopping criteria fulfilled

2.3 CONVERGENCE AND NUMERICAL ASPECTS

In the following, we first argue that (2.5) is solvable. Then we show that the sequence of Newton iterates converges locally at a quadratic rate as long as the optimal transport plan satisfies $P \geq c \cdot \mathbb{1}_{n,m}$ for some constant $c > 0$.

Lemma 2.1. *For $f \in \mathbb{R}^n$ and $g \in \mathbb{R}^m$, the Jacobian matrix $J_F(f, g)$ is symmetric positive semi-definite, and its kernel is given by*

$$\ker\left[J_F(f,g)\right] = \operatorname{span}\left\{\begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_m \end{pmatrix}\right\}. \tag{2.9}$$

Proof. The matrix is obviously symmetric. For arbitrary $\varphi \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}^m$, we obtain from (2.4) that

$$\begin{pmatrix} \varphi^\top & \gamma^\top \end{pmatrix} J_F(f,g) \begin{pmatrix} \varphi \\ \gamma \end{pmatrix} = \frac{1}{\varepsilon} \sum_{ij} P_{ij}(\varphi_i + \gamma_j)^2 \geq 0, \tag{2.10}$$

which holds with equality if and only if we have $\varphi_i + \gamma_j = 0$ for all i, j. □

Hence, the system (2.5) can be solved by a conjugate gradient (CG) method. To see that, recall that the CG method iterates on the orthogonal complement of the kernel as long as

the initial iterate $(\delta f^0, \delta g^0)$ is chosen from this subspace, in this case with $\mathbb{1}_n^\top \delta f^0 = \mathbb{1}_m^\top \delta g^0$. Furthermore, the Newton matrix can be applied matrix-free in an efficient manner as soon as the multiplication with $K = \exp(-C/\varepsilon)$ and its transpose can be done efficiently, see Algorithm 2. This is the case, for example if C_{ij} only depends on $i-j$ and thus, multiplication with K amounts to a convolution. A cheap diagonal preconditioner is provided by the matrix

$$\frac{1}{\varepsilon}\begin{bmatrix} \operatorname{Diag}(P^k \mathbb{1}_n) & 0 \\ 0 & \operatorname{Diag}([P^k]^\top \mathbb{1}_m) \end{bmatrix}. \tag{2.11}$$

According to Deuflhard [4, Thm. 2.3], we expect local quadratic convergence as long as

$$\|J_F(y^k)^{-1}[J_F(y^k) - J_F(\eta)](y^k - \eta)\| \le \omega \|y^k - \eta\|^2 \tag{2.12}$$

holds for all $\eta \in \mathbb{R}^n \times \mathbb{R}^m$ and $k \in \mathbb{N}$, with an arbitrary norm and some constant $\omega > 0$ in a neighborhood of the solution. Here, we abbreviated $y^k := (f^k, g^k)$.

Theorem 2.2. *For any $k \in \mathbb{N}$ with $P^k_{ij} > 0$, (2.12) holds in the ℓ_∞-norm for*

$$\omega \le (\mathrm{e}^{\frac{1}{\varepsilon}} - 1)\left(1 + 2\mathrm{e}^{\frac{1}{\varepsilon}} \frac{\max\left\{\|P^k \mathbb{1}_m\|_\infty,\ \|[P^k]^\top \mathbb{1}_n\|_\infty\right\}}{\min_{ij} P^k_{ij}}\right) \tag{2.13}$$

when $\|y^k - \eta\|_\infty \le 1$.

We postpone the proof of Theorem 2.2 to Section 4.

Remark 2.3. In fact, one can show that necessarily $\omega \ge \mathrm{e}^{\frac{1}{\varepsilon}} - 1$. Indeed, if $y^k - \eta = (\varphi, 0) \in \mathbb{R}^n \times \mathbb{R}^n$, then one can explicitly compute

$$J_F(y^k)^{-1}[J_F(y^k) - J_F(\eta)](y^k - \eta) = ((\mathrm{e}^{\varphi/\varepsilon} - 1)\varphi, 0),$$

where the exponential and the multiplication are pointwise (the calculation is detailed in the proof of Theorem 2.2).

Hence, if (f^0, g^0) is chosen sufficiently close to a solution of $F(f, g) = 0$, then the contraction property of Newton's method shows that the sequence of Newton iterates (f^k, g^k), and hence P^k, remain bounded. If the optimal plan satisfies $P^* \ge c \cdot \mathbb{1}_{n,m}$ for some $c > 0$, we can therefore expect local quadractic convergence of Newton's method.

2.4 Relation to Sinkhorn–Knopp

Substituting $u := \mathrm{e}^{-f/\varepsilon}$ and $v := \mathrm{e}^{-g/\varepsilon}$ in (2.1) shows that the optimality system can be written equivalently as

$$P = \operatorname{Diag}(u) K \operatorname{Diag}(v) \tag{2.14a}$$
$$a = \operatorname{Diag}(u) K v \tag{2.14b}$$
$$b = \operatorname{Diag}(v) K^\top u. \tag{2.14c}$$

In order to find a solution of (2.14b)–(2.14c), one can apply the Sinkhorn–Knopp algorithm [11] as recently proposed in Cuturi [2]. This amounts to alternating updates in the form of

$$u^{k+1} := \operatorname{Diag}(Kv^k)^{-1}a \tag{2.15a}$$

$$v^{k+1} := \operatorname{Diag}(K^\top u^{k+1})^{-1}b. \tag{2.15b}$$

In (2.15a), u^{k+1} is updated such that u^{k+1} and v^k solve (2.14b), and in the subsequent (2.15b), v^{k+1} is updated such that u^{k+1} and v^{k+1} form a solution of (2.14c).

If we proceed analogously to Section 2.2 and derive a Newton iteration to find a root of the function

$$G(u,v) := \begin{pmatrix} \operatorname{Diag}(u)Kv - a \\ \operatorname{Diag}(v)K^\top u - b \end{pmatrix}, \tag{2.16}$$

then the associated Jacobian matrix is

$$J_G(u,v) = \begin{pmatrix} \operatorname{Diag}(Kv) & \operatorname{Diag}(u)K \\ \operatorname{Diag}(v)K^\top & \operatorname{Diag}(K^\top u) \end{pmatrix}. \tag{2.17}$$

Neglecting the off-diagonal blocks in (2.17) and using the approximation

$$\hat{J}_G(u,v) = \begin{pmatrix} \operatorname{Diag}(Kv) & 0 \\ 0 & \operatorname{Diag}(K^\top u) \end{pmatrix} \tag{2.18}$$

to perform the Newton iteration

$$\begin{pmatrix} u^{k+1} \\ v^{k+1} \end{pmatrix} = \begin{pmatrix} u^k \\ v^k \end{pmatrix} - \hat{J}_G(u^k, v^k)^{-1} G(u^k, v^k) \tag{2.19}$$

leads us to the parallel updates

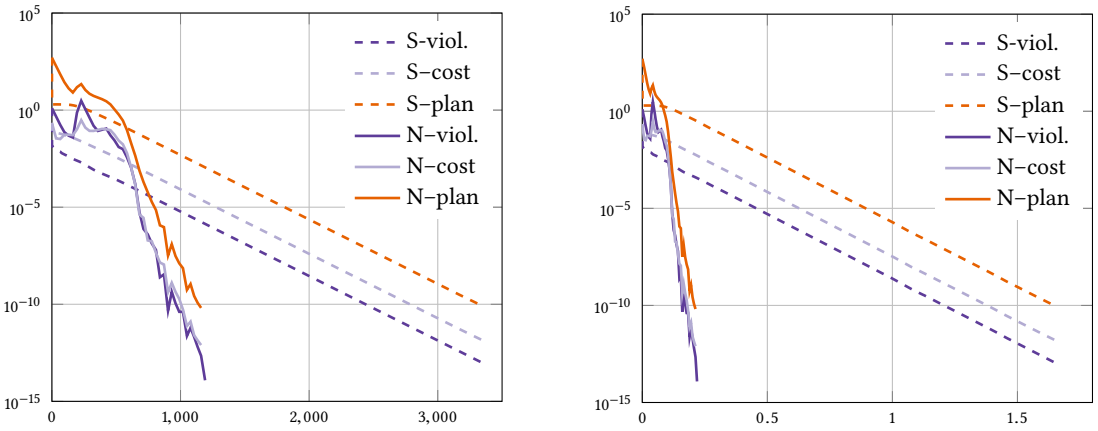
$$u^{k+1} := \operatorname{Diag}(Kv^k)^{-1}a \tag{2.20a}$$

$$v^{k+1} := \operatorname{Diag}(K^\top u^k)^{-1}b. \tag{2.20b}$$

Hence, we see that a Sinkhorn–Knopp step (2.15) simply approximates one Newton step (2.20) by neglecting the off-diagonal blocks and replacing u^k by u^{k+1} in (2.20b). In our experience, neither the Newton iteration for $G(u,v) = 0$ (which seems to work for the less general problem of matrix balancing; see Knight & Ruiz [8]) nor the version of Sinkhorn–Knopp in which v^{k+1} is updated using u^k instead of u^{k+1} converge.

3 NUMERICAL EXAMPLES

We illustrate the performance of the Sinkhorn–Newton method and its behavior by several examples. We note that a numerical comparison is not straightforward as there a several possibilities to tune the method, depending on the structure at hand and on the specific goals. As illustrated in Section 2.2, one could take advantage of fast applications of the matrix K or use less memory if one does not want to store P during the iteration. Here we focus on the



(a) Errors over iterations (CG for Newton).

(b) Errors over run time in seconds.

Figure 1: Performance of Sinkhorn (S) and Newton (N) iterations measured by constraint violation (viol.), distance to optimal transport cost (cost) and distance to optimal transport plan (plan).

comparison with the usual (linearly convergent) Sinkhorn iteration. Thus, we do not aim for greatest overall speed but for a fair comparison between the Sinkhorn-Newton method and the Sinkhorn iteration. To that end, we observe that one application of the Newton matrix (2.4) amounts to one multiplication with P and $P^\top$ each, two coordinate-wise products and sums of vectors. For one Sinkhorn iteration we need one multiplication with K and $K^\top$ and two additional coordinate-wise operations. Although Algorithm 2 looks a little closer to the Sinkhorn iteration, we still compare Algorithm 1, as we did not exploit any of the special structure in K or P.

All timings are reported using MATLAB (R2017b) implementations of the methods on an Intel Xeon E3-1270v3 (four cores at 3.5 GHz) with 16 GB RAM. The code used to generate the results below can be downloaded from https://github.com/dirloren/sinkhornnewton.

In all our experiments, we address the case $m = n$ and the considered histograms are defined on equidistant grids $\{x_i\}_{i=1}^n \subset [0,1]^d$ with $d = 2$ (in Sections 3.1 and 3.2) and $d = 1$ (in Section 3.3), respectively. Throughout, the cost is chosen as quadratic, i.e., $C_{ij} := \|x_i - x_j\|_2^2$.

Our Sinkhorn–Newton method is implemented according to Algorithm 1 and using a preconditioned CG method. The iteration is terminated as soon as the maximal violation of the constraints $\|a^k - a\|_\infty$ and $\|b^k - b\|_\infty$ drops below some threshold. If applicable, the same termination criterion is chosen for the Sinkhorn method, which is initialized with $u^0 = v^0 = \mathbb{1}_n$.

3.1 COMPARISON WITH SINKHORN–KNOPP

We first address the comparison of Sinkhorn–Newton with the classical Sinkhorn iteration. For this purpose, we discretize the unit square $[0,1]^2$ using a 20×20 equidistant grid $\{x_i =$

$(x_{i1}, x_{i2})\}_{i=1}^{400} \subset [0,1]^2$ and take

$$\tilde{a}_i := \mathrm{e}^{-36([x_{i1}-\frac{1}{3}]^2-[x_{i2}-\frac{1}{3}]^2)} + 10^{-1}, \tag{3.1a}$$

$$\tilde{b}_j := \mathrm{e}^{-9([x_{j1}-\frac{2}{3}]^2-[x_{j2}-\frac{2}{3}]^2)} + 10^{-1}, \tag{3.1b}$$

which are then normalized to unit mass by setting

$$a := \frac{\tilde{a}}{\sum_i \tilde{a}_i} \quad \text{and} \quad b := \frac{\tilde{b}}{\sum_j \tilde{b}_j}. \tag{3.2}$$

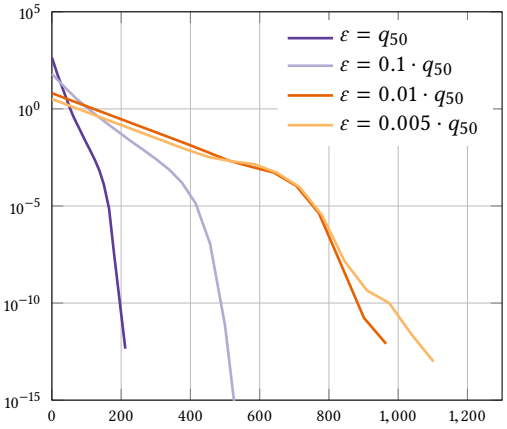
The entropic regularization parameter is set to $\varepsilon := 10^{-3}$ and in case of Sinkhorn-Newton, the CG method is implemented with a tolerance of 10^{-13} and a maximum number of 34 iterations. Moreover, the threshold for the termination criterion is chosen as 10^{-13}.

Figure 1 shows the convergence history of the constraint violation for both iterations together with the error in the unregularized transport cost $|\langle C, P^k - P^*\rangle|$, where P^* denotes the final transport plan, and the error in the transport plan $\|P^k - P^*\|_1$. In Figure 1a, we compare the error as a function of the iterations, where we take the total number of CG iterations for Sinkhorn–Newton to allow for a fair comparison (since both a Sinkhorn and a CG step have comparable costs, dominated by the two dense matrix–vector products Kv, $K^\top u$ and $P^\top \delta f$, $P\delta g$, respectively). It can be seen clearly that with respect to all error measures, Sinkhorn converges linearly while Sinkhorn–Newton converges roughly quadratically, as expected, with Sinkhorn–Newton significantly outperforming classical Sinkhorn for this choice of parameters. The same behavior holds if the error is measured as a function of runtime; see Figure 1b.

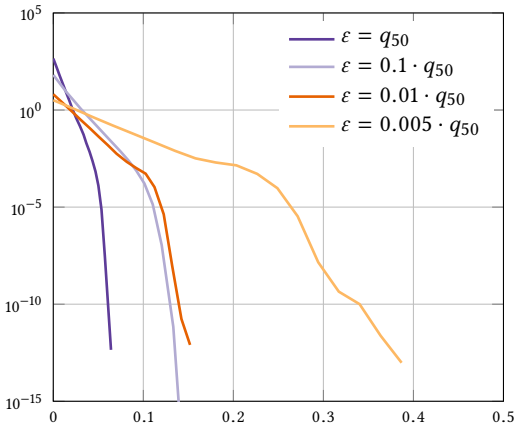
3.2 Dependence on the regularization strength

The second example addresses the dependence of the Sinkhorn–Newton method on the problem parameters. In particular, we consider the dependence on ε and on the minimal value of a and b (via the corresponding transport plans P), since these enter into the convergence rate estimate (2.13). Here, we take an example which is also used in Cuturi [2]: computing the transport distances between different images from the MNIST database, which contains 28×28 images of handwritten digits. We consider these as discrete distributions of dimension $28^2 = 784$ on $[0,1]^2$, to which we add a small offset γ before normalizing to unit mass as before. Here, the tolerance for both the Newton and the CG iteration is set to 10^{-12}, and the maximum number of CG iterations is fixed at 66. The entropic regularization parameter ε is chosen as multiples of the median of the cost (which is $q_{50} = 0.2821$ in this case).

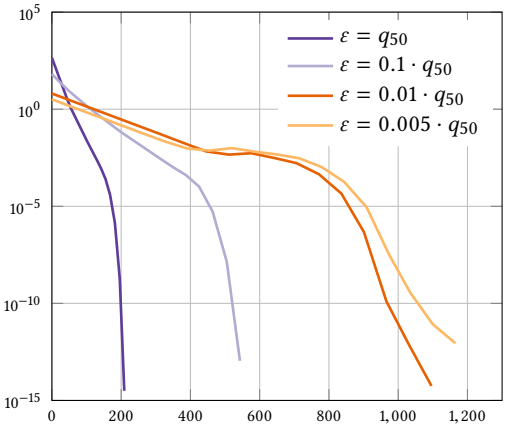
Figure 2 shows the convergence history for different offsets $\gamma \in \{0.5, 0.1, 0.01\}$ and $\varepsilon \in q_{50} \cdot \{1, 0.1, 0.01, 0.005\}$, where we again report the constraint violation both as a function of CG iterations and of the run time in seconds. Comparing Figures 2a to 2f, we see that as ε decreases, an increasing number of CG iterations is required to achieve the prescribed tolerance. However, the convergence seems to be robust in ε at least for larger values of ε and only moderately deteriorate for $\varepsilon \leq 0.01 q_{50}$.



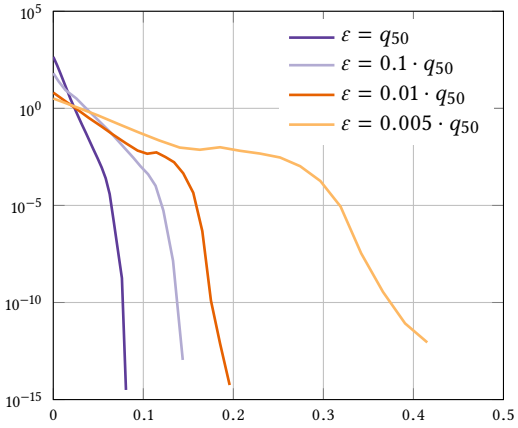
(a) $\gamma = 0.5$ (CG iterations)



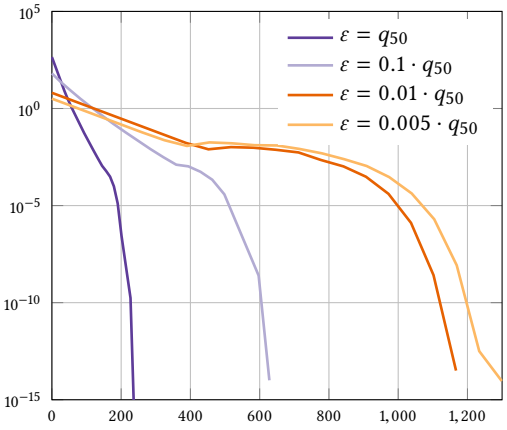
(b) $\gamma = 0.5$ (run time in seconds)



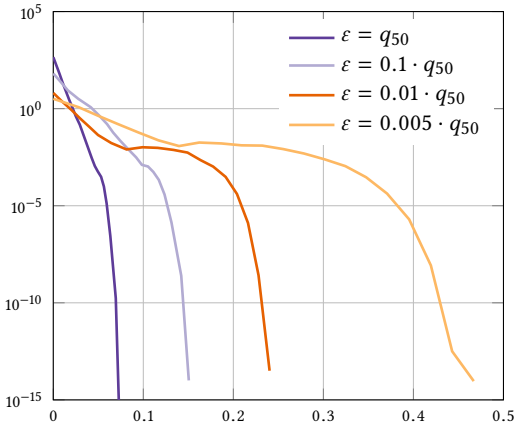
(c) $\gamma = 0.1$ (CG iterations)



(d) $\gamma = 0.1$ (run time in seconds)

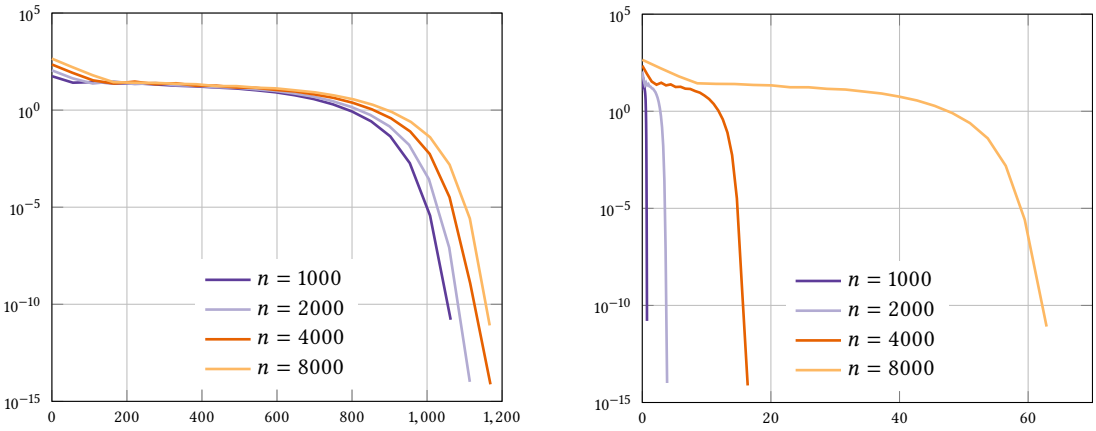


(e) $\gamma = 0.01$ (CG iterations)



(f) $\gamma = 0.01$ (run time in seconds)

Figure 2: Constraint violation for different offsets γ and different regularization parameters ε.



(a) Errors over CG iterations.

(b) Errors over run time in seconds.

Figure 3: Constraint violation for different mesh sizes n

3.3 DEPENDENCE ON THE PROBLEM DIMENSION

We finally address the dependence on the dimension of the problem. For this purpose, we discretize the unit interval $[0,1]$ using n equidistant points $x_i \in [0,1]$ and take

$$\tilde{a}_i = \mathrm{e}^{-100(x_i-0.2)^2} + \mathrm{e}^{-20|x_i-0.4|} + 10^{-2}, \tag{3.3a}$$

$$\tilde{b}_j = \mathrm{e}^{-100(x_j-0.6)^2} + 10^{-2}, \tag{3.3b}$$

which are again normalized to unit mass to obtain a and b. The regularization parameter is fixed at $\varepsilon = 10^{-3}$. Moreover, the inner and outer tolerances are here set to 10^{-10}, and the maximum number of CG iterations is coupled to the mesh size via $\lceil n/12 \rceil$.

Figure 3 shows the convergence behavior of Sinkhorn–Newton for $n \in \{1000, 2000, 4000, 8000\}$. As can be seen from Figure 3a, the behavior is nearly independent of n; in particular, the number of CG iterations required to reach the prescribed tolerance stays almost the same. (This is also true for the Newton method itself with 21, 22, 23 and 23 iterations.) Since each CG iteration involves two dense matrix–vector products with complexity $O(n^2)$, the total run time scales quadratically; see Figure 3b.

4 PROOF OF THEOREM 2.2

For the sake of presentation, we restrict ourselves to the case $m = n$ here. However, in the end, we suggest how the proof can be generalized to the case $m \neq n$.

To estimate (2.12) and in particular $J_F(y^k) - J_F(\eta)$ for $y^k = (f^k, g^k)$ and $\eta = (\alpha, \beta) \in \mathbb{R}^n \times \mathbb{R}^n$ we observe that

$$J_F(y^k) - J_F(\eta) = \frac{1}{\varepsilon}\left[\mathrm{e}^{\frac{-c_{ij}-f_i^k-g_j^k}{\varepsilon}} - \mathrm{e}^{\frac{-c_{ij}-\alpha_i-\beta_j}{\varepsilon}}\right]_{ij} = \left[P_{ij}^k\left(1 - \mathrm{e}^{\frac{f_i^k-\alpha_i+g_j^k-\beta_j}{\varepsilon}}\right)\right]_{ij}.$$

To keep the notation concise, we abbreviate $\psi = (\varphi, \gamma) \coloneqq y^k - \eta$ and also write y and P for y^k and P^k, respectively. Then, we compute

$$\begin{aligned} J_F(y)^{-1}[J_F(y) - J_F(\eta)](y-\eta) &= J_F(y)^{-1}\begin{bmatrix} \left(\sum_j P_{ij}(\mathrm{e}^{(\varphi_i+\gamma_j)/\varepsilon}-1)(\varphi_i+\gamma_j)/\varepsilon\right)_i \\ \left(\sum_i P_{ij}(\mathrm{e}^{(\varphi_i+\gamma_j)/\varepsilon}-1)(\varphi_i+\gamma_j)/\varepsilon\right)_j \end{bmatrix} \\ &= J_F(y)^{-1}\begin{bmatrix} \left(\sum_j P_{ij}\sum_{k=2}^{\infty}\frac{1}{(k-1)!\varepsilon^k}\sum_{l=0}^{k}\binom{k}{l}\varphi_i^l\gamma_j^{k-l}\right)_i \\ \left(\sum_i P_{ij}\sum_{k=2}^{\infty}\frac{1}{(k-1)!\varepsilon^k}\sum_{l=0}^{k}\binom{k}{l}\varphi_i^l\gamma_j^{k-l}\right)_j \end{bmatrix} \\ &= \sum_{k=2}^{\infty}\sum_{l=0}^{k}\binom{k}{l}\frac{1}{(k-1)!\varepsilon^k}J_F(y)^{-1}\begin{bmatrix} \left(\sum_j P_{ij}\varphi_i^l\gamma_j^{k-l}\right)_i \\ \left(\sum_i P_{ij}\varphi_i^l\gamma_j^{k-l}\right)_j \end{bmatrix} \end{aligned}$$

where all exponents are applied componentwise. Now we first treat only the summands for $l = 0$ and $l = k$. For those terms (2.4) immediately implies

$$\sum_{k=2}^{\infty}\sum_{l=0,k}\binom{k}{l}\frac{1}{(k-1)!\varepsilon^k}J_F(y)^{-1}\begin{bmatrix} \left(\sum_j P_{ij}(\varphi_i^k+\gamma_j^k)\right)_i \\ \left(\sum_i P_{ij}(\varphi_i^k+\gamma_j^k)\right)_j \end{bmatrix} = \sum_{k=2}^{\infty}\sum_{l=0,k}\frac{1}{(k-1)!\varepsilon^{k-1}}\begin{bmatrix}\varphi^k \\ \gamma^k\end{bmatrix} = \begin{bmatrix}(\mathrm{e}^{\varphi/\varepsilon}-1)\varphi \\ (\mathrm{e}^{\gamma/\varepsilon}-1)\gamma\end{bmatrix},$$

which has supremum norm bounded by $(\mathrm{e}^{\frac{1}{\varepsilon}}-1)\|(\varphi,\gamma)\|_\infty^2$ for all $\|(\varphi,\gamma)\|_\infty \le 1$. For all other summands (i.e. $1 \le l \le k-1$), we write

$$\begin{bmatrix}\alpha \\ \beta\end{bmatrix} \coloneqq \frac{\binom{k}{l}}{(k-1)!\varepsilon^k}J_F(y)^{-1}\begin{bmatrix} \left(\sum_j P_{ij}\varphi_i^l\gamma_j^{k-l}\right)_i \\ \left(\sum_i P_{ij}\varphi_i^l\gamma_j^{k-l}\right)_j \end{bmatrix}.$$

Using (2.4) again, it follows that

$$\underbrace{\begin{bmatrix}\mathrm{Diag}(P\mathbb{1}_n) & P \\ P^\top & \mathrm{Diag}(P^\top\mathbb{1}_n)\end{bmatrix}}_{=:A}\begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \frac{\binom{k}{l}}{(k-1)!\varepsilon^{k-1}}\begin{bmatrix}\mathrm{Diag}(P\gamma^{k-l})\varphi^l \\ \mathrm{Diag}(P^\top\varphi^l)\gamma^{k-l}\end{bmatrix},$$

and we aim to estimate $\|\alpha\|_\infty$ and $\|\beta\|_\infty$ by $\|\varphi\|_\infty$ and $\|\gamma\|_\infty$. By Lemma 2.1, the matrix A has a one-dimensional kernel spanned by $q \coloneqq (\mathbb{1}_n^\top, -\mathbb{1}_n^\top)^\top$, and a solution (α, β) in the orthogonal complement is also a solution to

$$\underbrace{(A + \Delta qq^\top)}_{B}\begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \frac{\binom{k}{l}}{(k-1)!\varepsilon^{k-1}}\begin{bmatrix}\mathrm{Diag}(P\gamma^{k-l})\varphi^l \\ \mathrm{Diag}(P^\top\varphi^l)\gamma^{k-l}\end{bmatrix}$$

for any $\Delta > 0$. From Varah [12], we know that the ℓ_∞-norm of the inverse matrix of B is estimated by

$$\|B^{-1}\|_\infty \le \left[\min_i\left(|B_{ii}| - \textstyle\sum_{j\neq i}|B_{ij}|\right)\right]^{-1}. \tag{4.1}$$

In this case, we calculate for $i = 1, \dots, n$ that

$$|B_{ii}| - \sum_{\substack{1 \le j \le 2n \\ j \neq i}} |B_{ij}| = \sum_{1 \le j \le n} P_{ij} + \Delta - (n-1)\Delta - \sum_{1 \le j \le n} |P_{ij} - \Delta|.$$

For any $\Delta \le \min_j P_{ij}$, this leads to

$$|B_{ii}| - \sum_{\substack{1 \le j \le 2n \\ j \neq i}} |B_{ij}| \le 2\Delta.$$

Similarly, we get that

$$|B_{n+i,n+i}| - \sum_{\substack{1 \le j \le 2n \\ j \neq n+i}} |B_{n+i,j}| \le 2\Delta.$$

Choosing $\Delta := \min_{ij} P_{ij}$, we thus obtain that

$$\|B^{-1}\|_\infty \le \left[2 \min_{ij} P_{ij}\right]^{-1}.$$

Using that

$$\| \operatorname{Diag}(P^\top \varphi^l)\gamma^{k-l}\| \le \|P^\top \mathbb{1}_n\|_\infty \|\varphi\|_\infty^l \|\gamma\|_\infty^{k-l},$$

and similarly for $\operatorname{Diag}(P\gamma^{k-l})\varphi^l$, finally gives

$$\|(\alpha, \beta)\|_\infty \le \frac{\binom{k}{l}}{(k-1)!\varepsilon^{k-1}} M \|\varphi\|_\infty^l \|\gamma\|_\infty^{k-l}$$

with

$$M = \frac{\max\{\|P\mathbb{1}_n\|_\infty, \|P^\top \mathbb{1}_n\|_\infty\}}{2 \min_{ij} P_{ij}}.$$

Hence we obtain

$$\begin{aligned}
&\left\| \sum_{k=2}^{\infty} \sum_{l=1}^{k-1} \binom{k}{l} \frac{1}{(k-1)!\varepsilon^k} J_F(y)^{-1} \begin{bmatrix} \left(\sum_j P_{ij} \varphi_i^l \gamma_j^{k-l}\right)_i \\ \left(\sum_i P_{ij} \varphi_i^l \gamma_j^{k-l}\right)_j \end{bmatrix} \right\|_\infty \\
&\qquad \le M \sum_{k=2}^{\infty} \sum_{l=1}^{k-1} \binom{k}{l} \frac{1}{(k-1)!\varepsilon^{k-1}} \|\varphi\|_\infty^l \|\gamma\|_\infty^{k-l} \\
&\qquad = M \Bigg[\left(\mathrm{e}^{(\|\varphi\|_\infty + \|\gamma\|_\infty)/\varepsilon} - 1\right) (\|\varphi\|_\infty + \|\gamma\|_\infty) \\
&\qquad\quad - \left(\mathrm{e}^{\|\varphi\|_\infty/\varepsilon} - 1\right) \|\varphi\|_\infty - \left(\mathrm{e}^{\|\gamma\|_\infty/\varepsilon} - 1\right) \|\gamma\|_\infty \Bigg] \\
&\qquad \le 2\mathrm{e}^{\frac{1}{\varepsilon}} M (\mathrm{e}^{\frac{1}{\varepsilon}} - 1) \|(\varphi, \gamma)\|_\infty^2
\end{aligned}$$

for all $\|(\varphi, \gamma)\|_\infty \le 1$. In summary we obtain

$$\|J_F(y)^{-1}[J_F(y) - J_F(\eta)](y - \eta)\|_\infty \le (1 + 2\mathrm{e}^{\frac{1}{\varepsilon}} M)(\mathrm{e}^{\frac{1}{\varepsilon}} - 1)\|y - \eta\|_\infty^2,$$

as desired

To generalize this to the case $m \neq n$, one can take $q = (\Delta_1 \mathbb{1}_n, -\Delta_2 \mathbb{1}_m)$ for $\Delta_1 \neq -\Delta_2$ and $\Delta_1 \Delta_2 = \Delta$ and choose Δ_1 to equilibrate the lower bounds for the first m and the last n rows of B.

5 CONCLUSION

We have proposed a Newton iteration to solve the entropically regularized discrete optimal transport problem. Different from related Newton type approaches for matrix balancing, our method iterates on the logarithm of the scalings, which seems to be necessary for robust convergence in the optimal transport setting. Numerical examples show that our algorithm is a robust and efficient alternative to the more commonly used Sinkhorn–Knopp algorithm, at least for small regularization strength.

REFERENCES

[1] Carlier, Duval, Peyré & Schmitzer, Convergence of entropic schemes for optimal transport and gradient flows, *SIAM Journal on Mathematical Analysis* 49 (2017), 1385–1418, DOI: 10.1137/15M1050264.

[2] Cuturi, Sinkhorn distances: Lightspeed computation of optimal transport, in: *Advances in Neural Information Processing Systems (NIPS) 26*, 2013, 2292–2300, URL: http://papers.nips.cc/paper/4927-sinkhorn-distances-lightspeed-computation-of-optimal-transport.

[3] Cuturi & Doucet, Fast computation of Wasserstein barycenters, in: *Proceedings of the 31st International Conference on Machine Learning*, 2014, 685–693, URL: http://proceedings.mlr.press/v32/cuturi14.pdf.

[4] Deuflhard, *Newton Methods for Nonlinear Problems, Affine Invariance and Adaptive Algorithms*, Springer, 2011, DOI: 10.1007/978-3-642-23899-4.

[5] Frogner, Zhang, Mobahi, Araya & Poggio, Learning with a Wasserstein loss, in: *Advances in Neural Information Processing Systems (NIPS) 28*, 2015, 2053–2061, URL: https://papers.nips.cc/paper/5679-learning-with-a-wasserstein-loss.

[6] Kantorovivc, On the translocation of masses, *C. R. (Doklady) Acad. Sci. URSS (N.S.)* 37 (1942), 199–201.

[7] Kantorovivc & Rubinvsteuin, On a functional space and certain extremum problems, *Doklady Akademii Nauk SSSR* 115 (1957), 1058–1061.

[8] Knight & Ruiz, A fast algorithm for matrix balancing, *IMA J. Numer. Anal.* 33 (2013), 1029–1047, DOI: 10.1093/imanum/drs019.

[9] Kolouri, Park, Thorpe, Slepcev & Rohde, Optimal mass transport: signal processing and machine-learning applications, *IEEE Signal Processing Magazine* 34 (2017), 43–59, DOI: 10.1109/MSP.2017.2695801.

[10] Monge, Mémoire sur la théorie des déblais et des remblais, *Histoire de l'Académie Royale des Sciences de Paris* (1781).

[11] Sinkhorn & Knopp, Concerning nonnegative matrices and doubly stochastic matrices, *Pacific Journal of Mathematics* 21 (1967), 343–348, DOI: 10.2140/pjm.1967.21.343.

[12] Varah, A lower bound for the smallest singular value of a matrix, *Linear Algebra and Appl.* 11 (1975), 3–5, DOI: 10.1016/0024-3795(75)90112-3.