Efficient Discrete Optimal Transport Algorithm by Accelerated Gradient Descent

Dongsheng An[1] , **Na Lei**[2*], **Xiaoyin Xu**[3], **Xianfeng Gu**[1]

[1]Stony Brook University, NY, USA
[2] Dalian University of Technology, Liaoning, China
[3]Harvard Medical School, MA, USA
{doan, gu}@cs.stonybrook.edu, nalei@dlut.edu.cn, xxu@bwh.harvard.edu

Abstract

Optimal transport (OT) plays an essential role in various areas like machine learning and deep learning. However, computing discrete OT for large scale problems with adequate accuracy and efficiency is highly challenging. Recently, methods based on the Sinkhorn algorithm add an entropy regularizer to the prime problem and obtain a trade off between efficiency and accuracy. In this paper, we propose a novel algorithm based on Nesterov's smoothing technique to further improve the efficiency and accuracy in computing OT. Basically, the non-smooth c-transform of the Kantorovich potential is approximated by the smooth Log-Sum-Exp function, which smooths the original non-smooth Kantorovich dual functional. The smooth Kantorovich functional can be efficiently optimized by a fast proximal gradient method, the fast iterative shrinkage thresholding algorithm (FISTA). Theoretically, the computational complexity of the proposed method is lower than current estimation of the Sinkhorn algorithm in terms of the precision. Experimentally, compared with the Sinkhorn algorithm, our results demonstrate that the proposed method achieves faster convergence and better accuracy with the same parameter.

Introduction

Optimal transport (OT) is a powerful tool to compute the Wasserstein distance between probability measures and widely used to model various natural and social phenomena, including economics (Galichon 2016), optics (Glimm and Oliker 2003), biology (Schiebinger et al. 2019), physics (Jordan, Kinderlehrer, and Otto 1998) and in other scientific fields. Recently, OT has been successfully applied in machine learning and statistics, such as parameter estimation in Bayesian non-parametric models (Nguyen 2013), computer vision (Arjovsky, Chintala, and Bottou 2017; Courty et al. 2017; Tolstikhin et al. 2018; An et al. 2020a; Lei et al. 2020; An et al. 2020b), and natural language processing (Kusner et al. 2015; Yurochkin et al. 2019). In these areas, the complex probability measures are approximated by summations of Dirac measures supported on the samples. To obtain the Wasserstein distance between the empirical distributions, we then solve the discrete OT problems.

*Corresponding author

Discrete Optimal Transport In discrete OT problem, where both the source and target measures are discrete, the Kantorovich functional becomes a convex function defined on a convex domain. Due to the lack of smoothness, conventional gradient descend method can not be applied directly. Instead, it can be optimized with the sub-differential method (Nesterov 2005), in which the gradient is replaced by the sub-differential. To achieve an approximation error less than ε, the sub-differential method requires $O(1/\varepsilon^2)$ iterations. Recently, several approximation methods have been proposed to improve the computational efficiency. In these methods (Cuturi 2013; Benamou et al. 2015; Altschuler, Niles-Weed, and Rigollet 2017), a strongly convex entropy function is added to the prime Kantorovich problem and thus the regularized problem can be efficiently solved by the Sinkhorn algorithm. More detailed analysis shows that the computational complexity of the Sinkhorn algorithm is $\tilde{O}(n^2/\varepsilon^2)$ (Dvurechensky, Gasnikov, and Kroshnin 2018) by setting $\lambda = \epsilon/4\log n$. Also, a series of primal-dual algorithms are proposed, including the APDAGD (adaptive primal-dual accelerated gradient descent) algorithm (Dvurechensky, Gasnikov, and Kroshnin 2018) with computational complexity $\tilde{O}(n^{2.5}/\epsilon)$, the APDAMD (adaptive primal-dual accelerated mirror descent) algorithm (Lin, Ho, and Jordan 2019) with $\tilde{O}(n^2\sqrt{r}/\epsilon)$ where r is a complex constant of the Bregman divergence, and the APDRCD (accelerated primal-dual randomized coordinate descent) algorithm (Guo, Ho, and Jordan 2020) with $\tilde{O}(n^{2.5}/\epsilon)$. But all of the three methods need to build a matrix with space complexity $O(n^3)$, making them difficult to compute when n is large.

Our Method In this work, instead of starting from the prime Kantorovich problem like the Sinkhorn based methods, we directly deal with the dual Kantorovich problem. The key idea is to approximate the original non-smooth c-transform of the Kantorovich potential by Nesterov's smoothing technique. Specifically, we approximate the $\max$ function by the Log-Sum-Exp function, which has also been used in (Schmitzer 2019; Peyré and Cuturi 2018), such that the original non-smooth Kantorovich functional is converted to an unconstrained $(n-1)$-dimensional smooth convex energy. By using the Fast Proximal Gradient Method named FISTA (Beck and Teboulle 2009), we can quickly optimize

the smoothed energy to get a precise estimate of the OT cost. In theory, the method can achieve the approximate error ε with the space complexity $O(n^2)$ and computational complexity $O(n^{2.5}\sqrt{\log n}/\varepsilon)$. Additionally, we show that the induced approximate OT plan by our algorithm is equivalent to that of the Sinkhorn algorithm. The contributions of our work are as follows.

- We convert the dual Kantorovich problem to an *unconstrained smooth convex optimization problem* by approximating the non-smooth c-transform of the Kantorovich potential with Nesterov's smoothing idea.
- The *smoothed Kantorovich functional* can be efficiently solved by the FISTA algorithm with computational complexity $\tilde{O}(n^{2.5}/\sqrt{\varepsilon})$. At the same time, the computational complexity of the Kantorovich functional itself is given by $\tilde{O}(n^{2.5}/\varepsilon)$.
- The experiments demonstrate that compared with the Sinkhorn algorithm, the proposed method achieves faster convergence and better accuracy with the same parameter λ.

Notation In this work, $\mathbb{R}_{\geq 0}$ represents the non negative real numbers, $\mathbf{0}$ and $\mathbf{1}$ represents the all-zeros and all-ones vectors of appropriate dimension. The set of integers $\{1, 2, \ldots, n\}$ is denoted as $[n]$. And $|\cdot|_1$ and $\|\cdot\|$ are the ℓ_1 and ℓ_2 norms, $|v|_1 = \sum_i |v_i|$ and $\|v\| = \sqrt{\sum_i v_i^2}$, respectively. $R(C)$ is the range of the cost matrix $C = (c_{ij})$, namely $C_{\max} - C_{\min}$, where $C_{\max}$ and $C_{\min}$ represent the maximum and minimum of the elements of C with $c_{ij} > 0$. We use $\nu_{\min}$ to denote the minimal element of ν and $\oslash$ to denote element wise division.

Related Work

Optimal transport plays an important role in various kinds of fields, and there is a huge literature in this area. Here we mainly focus on the most related works. For detailed overview, we refer readers to (Peyré and Cuturi 2018).

When both the source and target measures are discrete, the OT problem can be treated as a standard linear programming (LP) task and solved by interior-point method with computational complexity $\tilde{O}(n^{5/2})$ (Lee and Sidford 2014). But this method requires a practical solver of the Laplacian linear system, which is not currently available for large dataset. Another interior-point based method to solve the OT problem is proposed by Pele and Werman (Pele and Werman 2009) with complexity $\tilde{O}(n^3)$. Generally speaking, it is unrealistic to solve the large scale OT problem with the traditional LP solvers.

The prevalent way to compute the OT cost between two discrete measures involves adding a strongly convex entropy function to the prime Kantorovich problem (Cuturi 2013; Benamou et al. 2015). Most of the current solutions for the discrete OT problem follow this strategy. Genevay et al. (Genevay et al. 2016) extend the algorithm in its dual form and solve it by stochastic average gradient method. The Greenkhorn algorithm (Altschuler, Niles-Weed, and Rigollet 2017; Abid and Gower 2018; Chakrabarty and Khanna 2021) is a greedy version of the Sinkhorn algorithm. Specifically, Altschuler et al. (Altschuler, Niles-Weed, and Rigollet 2017) show that the complexity of their algorithm is $\tilde{O}(\frac{n^2}{\epsilon^3})$. Later, Dvurechensky et al. (Dvurechensky, Gasnikov, and Kroshnin 2018) improve the complexity bound of the Sinkhorn algorithm to $\tilde{O}(\frac{n^2}{\epsilon^2})$, and propose an APDAGD method with complexity $\tilde{O}(\min\{\frac{n^{9/4}}{\epsilon}, \frac{n^2}{\epsilon^2}\})$. Jambulapati et al. (Jambulapati, Sidford, and Tian 2019) introduce a parallelelizable algorithm to compute the OT problem with complexity $\tilde{O}(\frac{n^2\|C\|_{\max}}{\epsilon})$. Through screening the negligible components by directly setting them at that value before entering the Sinkhorn problem, the screenkhorn (Alaya et al. 2019) method solves a smaller Sinkhorn problem and improves the computation efficiency. Based on a primal-dual formulation and a tight upper bound for the dual solution, Lin et al. (Lin, Ho, and Jordan 2019) improve the complexity bound of the Greenkhorn algorithm to $\tilde{O}(\frac{n^2}{\epsilon^2})$, and propose the APDAMD algorithm, whose complexity bound is proven to be $\tilde{O}(\frac{n^2\sqrt{r}}{\epsilon})$, where $r \in (0, n]$ refers to some constants in the Bregman divergence. Recently, a practically more efficient method called APDRCD (Guo, Ho, and Jordan 2020) is proposed with complexity $\tilde{O}(n^{2.5}/\epsilon)$. But all these three primal-dual based methods need to build a matrix with space complexity $O(n^3)$, which makes them impractical when n is large. By utilizing Newton-type information, Blanchet et al. (Blanchet et al. 2018) and Quanrud (Quanrud 2018) propose algorithms with complexity $\tilde{O}(\frac{n^2}{\epsilon})$. However, the Newton-based methods only give the theoretical upper bound and provide no practical algorithms.

Besides the entropy regularizer based methods, Blondel et al. (Blondel, Seguy, and Rolet 2018) use the squared 2-norm and group LASSO (least absolute shrinkage and selection operator) to regularize the prime Kantorovich problem and then use the quasi-Newton method to accelerate the algorithm. Xie et al. (Xie et al. 2019b) develop an Inexact Proximal point method for exact optimal transport. By utilizing the structure of the cost function, Gerber and Maggioni (Gerber and Maggioni 2017) optimize the transport plan from coarse to fine. Meng et al. (Meng et al. 2019) propose the projection pursuit Monge map, which accelerates the computation of the original sliced OT problem. Xie et al. (Xie et al. 2019a) also use the generative learning based method to model the optimal transport. But the theoretical analysis of these algorithms is still nascent.

In this work, we introduce a method based on Nesterov's smoothing technique, which is applied to the dual Kantorovich problem with computational complexity $O(n^{2.5}\sqrt{\log\ n}/\varepsilon)$ (or equivalently $\tilde{O}(n^{2.5}/\varepsilon)$) and approximation error bound $2\lambda \log n$.

Optimal Transport Theory

In this section, we introduce some basic concepts and theorems in the classical optimal transport theory, focusing on Kantorovich's approach and its generalization to the discrete settings via c-transform. The details can be found in Villani's book (Villani 2008).

Optimal Transport Problem Suppose $X \subset \mathbb{R}^d, Y \subset \mathbb{R}^d$ are two subsets of the Euclidean space $\mathbb{R}^d$, μ, ν are two probability measures defined on X and Y with equal total measure, $\mu(X) = \nu(Y)$.

Kantorovich's Approach Depending on the cost functions and the measures, the OT map between (X, μ) and (Y, ν) may not exist. Thus, Kantorovich relaxed the transport maps to transport plans, and defined joint probability measure $\pi : X \times Y \to \mathbb{R}_{\geq 0}$, such that the marginal probability of π equals to μ and ν, respectively. Formally, let the projection maps be $\rho_x(x, y) = x$, $\rho_y(x, y) = y$, then we define

$$\pi(\mu, \nu) := \{P : X \times Y \to \mathbb{R}_{\geq 0} : (\rho_x)_{\#} P = \mu, (\rho_y)_{\#} P = \nu\} \tag{1}$$

Problem 1 (Kantorovich Problem)**.** *Given the transport cost function $c : X \times Y \to \mathbb{R}$, find the joint probability measure $P : X \times Y \to \mathbb{R}$ that minimizes the total transport cost*

$$M_c(\mu, \nu) = \min_{P \in \pi(\mu,\nu)} \int_{X \times Y} c(x, y) dP(x, y) \tag{2}$$

Problem 2 (Dual Kantorovich Problem)**.** *Given two probability measures μ and ν supported on X and Y, respectively, and the transport cost function $c : X \times Y \to \mathbb{R}$, the Kantorovich problem is equivalent to maximizing the following* **Kantorovich functional***:*

$$M_c(\mu, \nu) = \max \left\{ - \int_X \phi d\mu + \int_Y \psi d\nu \right\} \tag{3}$$

where $\phi \in L^1(X, \mu)$ and $\psi \in L^1(Y, \nu)$ are called **Kantorovich potentials** *and $-\phi(x)+\psi(y) \leq c(x, y)$. The above problem can be reformulated as the following minimization form with the same constraints:*

$$M_c(\mu, \nu) = - \min \left\{ \int_X \phi d\mu - \int_Y \psi d\nu \right\} \tag{4}$$

Definition 3 (c-transform)**.** *Let $\phi \in L^1(X, \mu)$ and $\psi \in L^1(Y, \nu)$, we define*

$$\phi(x) = \psi^c(x) = \sup_{y \in Y} \psi(y) - c(x, y).$$

With c-transform, Eqn. (4) is equivalent to solving the following optimization problem:

$$M_c(\mu, \nu) = - \min \left\{ \int_X \psi^c(x) d\mu(x) - \int_Y \psi(y) d\nu(y) \right\} \tag{5}$$

where $\psi \in L^1(Y, \nu)$. When $\mu = \sum_{i=1}^m \mu_i \delta(x - x_i)$ and $\nu = \sum_{j=1}^n \nu_j \delta(y - y_j)$, $\psi = (\psi_1, \psi_2, \ldots, \psi_n)^T$, Eqn. (5) gives the unconstrained convex optimization problem:

$$M_c(\mu, \nu) = - \min_{\psi} E(\psi) = - \min_{\psi} \{ \sum_{i=1}^m \mu_i \psi^c(x_i) - \sum_{j=1}^n \nu_j \psi_j \} \tag{6}$$

where the c-transform of ψ is given by:

$$\psi^c(x_i) = \max_j \{\psi_j - c_{ij}\} \tag{7}$$

where $c_{ij} = c(x_i, y_j)$. Suppose ψ^* is the solution to Eqn. (6), then it has the following properties:

1. If the cost function is $\hat{c}(x, y) = c(x, y) - k$, where k is a constant, the corresponding optimal solution is $\hat{\psi}^*$, then $\hat{\psi}^* = \psi^*$. At the same time, we have $M_c(\mu, \nu) = M_{\hat{c}}(\mu, \nu) + k$.
2. $\psi^* + k\mathbf{1}$ is also an optimal solution for Eqn. (6).

In order to make the solution unique, we add a constraint $\psi \in H$ using the indicator function I_H, where $H = \{\psi | \sum_{j=1}^n \psi_j = 0\}$, and modify the **Kantorovich functional** $E(\psi)$ in Eqn. (6) as:

$$\tilde{E}(\psi) = E(\psi) + I_H(\psi), \quad I_H(\psi) = \begin{cases} 0 & \psi \in H \\ \infty & \psi \notin H \end{cases} \tag{8}$$

Then solving Eqn. (6) is equivalent to finding the solution to:

$$\begin{aligned} M_c(\mu, \nu) &= - \min_{\psi} \tilde{E}(\psi) \\ &= - \min_{\psi} \left\{ \sum_{i=1}^m \mu_i \psi^c(x_i) - \sum_{j=1}^n \nu_j \psi_j + I_H(\psi) \right\} \end{aligned} \tag{9}$$

which is essentially an $(n-1)$-dimensional unconstrained convex problem. According to the definition of c-transform in Eqn. (7), ψ^c is non-smooth with respect to ψ.

Nesterov's Smoothing

Following Nesterov's original strategy (Nesterov 2005), which has also been applied in the OT field (Peyré and Cuturi 2018; Schmitzer 2019), we smooth the non-smooth discrete Kantorovich functional $E(\psi)$. We approximate $\psi^c(x)$ with the Log-Sum-Exp function to get the smooth Kantorovich functional $E_\lambda(\psi)$. Then through the FISTA algorithm (Beck and Teboulle 2009), we can easily induce that the computation complexity of our algorithm is $O(n^{2.5}\sqrt{\log n}/\varepsilon)$, with $\tilde{E}(\psi^t) - \tilde{E}(\psi^*) \leq \epsilon$. By abuse of notation, in the following we call both $E(\psi)$ and $\tilde{E}(\psi)$ the Kantorovich functional and both $E_\lambda(\psi)$ and $\tilde{E}_\lambda(\psi)$ the smooth Kantorovich functional.

Definition 4 ((α, β)-smoothable)**.** *A convex function f is called (α, β)-smoothable if, for any $\lambda > 0$, $\exists$ a convex function f_λ such that*

$$f_\lambda(x) \leq f(x) \leq f_\lambda(x) + \beta\lambda$$
$$f_\lambda(y) \leq f_\lambda(x) + \langle \nabla f_\lambda(x), y - x \rangle + \frac{\alpha}{2\lambda}(y - x)^T H_\lambda (y - x)$$

Here $H_\lambda = \nabla^2 f_\lambda(x)$ and f_λ is called a $\frac{1}{\lambda}$-smooth approximation of f with parameters (α, β).

The parameter λ defines a trade-off between the approximation accuracy and the smoothness, where the smaller the λ, the better approximation and the less smoothness.

Lemma 5 (Nesterov's Smoothing)**.** *Given $f : \mathbb{R}^n \to \mathbb{R}$, $f(x) = \max\{x_j : j = 1, \ldots, n\}$, for any $\lambda > 0$, we have its $\frac{1}{\lambda}$-smooth approximation with parameters $(1, \log n)$*

$$f_\lambda(x) = \lambda \log(\sum_{j=1}^n e^{x_j/\lambda}) - \lambda \log n, \tag{10}$$

Proof. We have $\forall x \in \mathbb{R}^n$,

$$f_\lambda(x) \le \lambda \log(n \max_j e^{x_j/\lambda}) - \lambda \log(n) = f(x)$$

$$f(x) = \lambda \log \max_j e^{x_j/\lambda} < \lambda \log \sum_{j=1}^n e^{x_j/\lambda} = f_\lambda(x) + \lambda \log n$$

Furthermore, it is easy to prove that $f_\lambda(x)$ is $\frac{1}{\lambda}$-smooth. Therefore, $f_\lambda(x)$ is an approximation of $f(x)$ with parameters $(1, \log n)$. □

Recalling the definition of c-transform of the Kantorovich potential in Eqn. (7), we obtain the Nesterov's smoothing of ψ^c by applying Eqn. (10)

$$\psi^c_\lambda = \lambda \log \left(\sum_{j=1}^n e^{(\psi_j - c_{ij})/\lambda} \right) - \lambda \log n. \tag{11}$$

We use ψ^c_λ to replace ψ^c in Eqn. (9) to approximate the Kantorovich functional. Then the Nesterov's smoothing of the Kantorovich functional becomes

$$E_\lambda(\psi) = \lambda \sum_{i=1}^m \mu_i \log \left(\sum_{j=1}^n e^{(\psi_j - c_{ij})/\lambda} \right) - \sum_{j=1}^n \nu_j \psi_j - \lambda \log n \tag{12}$$

and its gradient is given by

$$\frac{\partial E_\lambda(\psi)}{\partial \psi_j} = \sum_{i=1}^m \mu_i \frac{e^{(\psi_j - c_{ij})/\lambda}}{\sum_{k=1}^n e^{(\psi_k - c_{ik})/\lambda}} - \nu_j, \ \forall\, j \in [n] \tag{13}$$

Furthermore, we can directly compute the Hessian matrix of $E_\lambda(\psi)$. Let $K_{ij} = e^{-c_{ij}/\lambda}$ and $v_j = e^{\psi_j/\lambda}$, and set $E^i_\lambda := \lambda \log \sum_{j=1}^n K_{ij} v_j, \ \forall\, i \in [m]$. Direct computation gives the following gradient and Hessian matrix:

$$\begin{aligned} \nabla E_\lambda &= diag(v) K^T (\mu \oslash Kv) - \nu \\ \nabla^2 E_\lambda &= \sum_{i=1}^m \mu_i \nabla^2 E^i_\lambda, \\ \nabla^2 E^i_\lambda &= \frac{1}{\lambda} \left(\frac{1}{\mathbf{1}^T V_i} \Lambda_i - \frac{1}{(\mathbf{1}^T V_i)^2} V_i V_i^T \right) \end{aligned} \tag{14}$$

where $V_i = (K_{i1}v_1, K_{i2}v_2, \ldots, K_{in}v_n)^T$, and $\Lambda_i = \mathrm{diag}(K_{i1}v_1, K_{i2}v_2, \ldots, K_{in}v_n)$. By the Hessian matrix, we can show that E_λ is a smooth approximation of E.

Lemma 6. *$E_\lambda(\psi)$ is a $\frac{1}{\lambda}$-smooth approximation of $E(\psi)$ with parameters $(1, \log n)$.*

Proof. In Eqn. (14), $\nabla^2 E^i_\lambda$ has $K = \{k\mathbf{1} : k \in \mathbb{R}\}$ as its null space. In the orthogonal complementary space of K, $\nabla^2 E^i_\lambda$ is diagonal dominant, thus strictly positive definite.

Weyl's inequality (Horn and Johnson 1991) states that the eigen value of $A = B + C$ is no greater than the maximal eigenvalue of B minus the minimal eigenvalue of C, where B is an exact matrix and C is a perturbation matrix. Hence the maximal eigenvalue of $\nabla^2 E^i_\lambda$, denoted as σ_i, has an upper bound,

$$0 \le \sigma_i \le \frac{1}{\lambda} \frac{1}{\mathbf{1}^T V_i} \max_j \{K_{ij} v_j\} \le \frac{1}{\lambda}.$$

Thus the maximal eigenvalue of $\nabla^2 E_\lambda(\psi)$ is no greater than $\sum_{i=1}^n \mu_i \sigma_i \le \frac{1}{\lambda}$. It is easy to find that $E_\lambda(\psi) \le E(\psi) \le E_\lambda(\psi) + \lambda \log n$. Thus, $E_\lambda(\psi)$ is a $\frac{1}{\lambda}$-smooth approximation of $E(\psi)$ with parameters $(1, \log n)$. □

Lemma 7. *Suppose $E_\lambda(\psi)$ is the $\frac{1}{\lambda}$-smooth approximation of $E(\psi)$ with parameters $(1, \log n)$, ψ^*_λ is the optimizer of $E_\lambda(\psi)$, then the approximate OT plan is unique and given by*

$$(P^*_\lambda)_{ij} = \mu_i \frac{e^{((\psi^*_\lambda)_j - c_{ij})/\lambda}}{\sum_{k=1}^n e^{((\psi^*_\lambda)_k - c_{ik})/\lambda}} = \frac{\mu_i K_{ij} v^*_j}{K_i v^*} \tag{15}$$

where K_i is the ith row of K and $v^* = e^{\psi^*_\lambda/\lambda}$.

Proof. By the gradient formula Eqn. (13) and the optimizer ψ^*_λ, we have

$$\frac{\partial E_\lambda(\psi^*)}{\partial \psi_j} = \sum_{i=1}^m (P^*_\lambda)_{ij} - \nu_j = 0 \quad \forall\, j = 1, \cdots, n.$$

On the other hand, by the definition of P^*_λ, we have

$$\sum_{j=1}^n (P^*_\lambda)_{ij} = \mu_i, \quad \forall\, i = 1, \ldots, m,$$

Combing the above two equations, we obtain that $P^*_\lambda \in \pi(\mu, \nu)$ and it is the approximate OT plan. □

Similar to the discrete Kantorovich functional in Eqn. (6), the optimizer of the smooth Kantorovich functional in Eqn. (12) is also not unique: given an optimizer ψ^*_λ, then $\psi^*_\lambda + k\mathbf{1}$, $k \in \mathbb{R}$ is also an optimizer. We can eliminate the ambiguity by adding the indicator function as Eqn. (8), $\tilde{E}_\lambda(\psi) = E_\lambda(\psi) + I_H(\psi)$,

$$\begin{aligned} \tilde{E}_\lambda(\psi) = \lambda \sum_{i=1}^m \mu_i \log \left(\sum_{j=1}^n e^{(\psi_j - c_{ij})/\lambda} \right) \\ - \sum_{j=1}^n \nu_j \psi_j - \lambda \log n + I_H(\psi) \end{aligned} \tag{16}$$

This energy can be optimized effectively through the following FISTA iterations (Beck and Teboulle 2009).

$$\begin{aligned} z^{t+1} &= \Pi_{\eta_t I_H} \left(\psi^t - \eta_t \nabla E_\lambda(\psi^t) \right) \\ \psi^{t+1} &= z^{t+1} + \frac{\theta_t - 1}{\theta_{t+1}} (z^{t+1} - z^t) \end{aligned} \tag{17}$$

with initial conditions $\psi^0 = v^0 = \mathbf{0}$, $\theta_0 = 1$, $\eta_t = \lambda$ and $\theta_{t+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4\theta_t^2}\right)$. Here $\Pi_{\eta_t I_H}(z) = z - \frac{1}{n}\sum_{j=1}^n z_j$ is the projection of z to H (the proximal function of $I_H(x)$) (Parikh and Boyd 2014). Similar to the Sinkhorn's algorithm, this algorithm can be parallelized, since all the operations are row based.

Theorem 8. *Given the cost matrix $C = (c_{ij})$, the source measure $\mu \in R^m_+$ and target measure $\nu \in R^n_+$ with $\sum_{i=1}^m \mu_i = \sum_{j=1}^n \nu_j = 1$, ψ^* is the optimizer of the discrete dual Kantorovich functional $\tilde{E}(\psi)$, and ψ^*_λ is the optimizer of the smooth Kantorovich functional $\tilde{E}_\lambda(\psi)$. Then the approximation error is*

$$|\tilde{E}(\psi^*) - \tilde{E}_\lambda(\psi^*_\lambda)| \le 2\lambda \log n$$

Proof. Assume ψ^* and ψ_λ^* are the minimizers of $E(\psi)$ and $E_\lambda(\psi)$ respectively. Then by the inequality in Eqn. (10)

$$\begin{array}{llll} E_\lambda(\psi^*) & \le E(\psi^*) & \le E(\psi_\lambda^*) & \le E_\lambda(\psi_\lambda^*) + \lambda \log n \\ E_\lambda(\psi_\lambda^*) & \le E_\lambda(\psi^*) & \le E(\psi^*) & \le E_\lambda(\psi^*) + \lambda \log n \end{array}$$

This shows $|E_\lambda(\psi^*) - E_\lambda(\psi_\lambda^*)| \le \lambda \log n$. Removing the indicator functions, we can get

$$\begin{aligned} &|\tilde{E}(\psi^*) - \tilde{E}_\lambda(\psi_\lambda^*)| \\ =&|E(\psi^*) - E_\lambda(\psi_\lambda^*)| \\ \le&|E(\psi^*) - E_\lambda(\psi^*)| + |E_\lambda(\psi^*) - E_\lambda(\psi_\lambda^*)| \\ \le&2\lambda \log n \end{aligned}$$ □

This also shows that $E(\psi_\lambda^*)$ converges quickly to $E(\psi^*)$ as the decrease of λ. The convergence analysis of FISTA is given as follows:

Theorem 9 (Thm 4.4 of (Beck and Teboulle 2009))**.** *Assume **(1)** $g(x)$ is convex and differentiable with $dom(g) = \mathbb{R}^n$, ∇g is Lipschitz continuous with Lipschitz constant $L > 0$; and **(2)** $h(x)$ is convex and its proximal function can be evaluated. Then from the minimization of $f(x) = g(x) + h(x)$ by FISTA with fixed step size $\eta_t = \frac{1}{L}$, we can get*

$$f(x^t) - f(x^*) \le \frac{2L}{(t+1)^2}\|x^0 - x^*\|^2 \tag{18}$$

Corollary 10. *Suppose λ is fixed and $\psi^0 = \mathbf{0}$, then for any $t \ge \sqrt{\frac{2\|\psi_\lambda^*\|^2}{\lambda\varepsilon}}$, we have*

$$\tilde{E}_\lambda(\psi^t) - \tilde{E}_\lambda(\psi_\lambda^*) \le \varepsilon, \tag{19}$$

where ψ_λ^* is the optimizer of $\tilde{E}_\lambda(\psi)$.

Proof. Under the settings of the smoothed Kantorovich problem Eqn. (12), $E_\lambda(\psi)$ is convex and differentiable with $\nabla^2 E_\lambda(\psi) \preceq \frac{1}{\lambda} I$, $I_H(\psi)$ is convex and its proximal function is given by $\Pi_H(v)$. Thus, directly applying Thm. 9 and setting $L = \frac{1}{\lambda}$, we can get $\tilde{E}_\lambda(\psi^t) - \tilde{E}_\lambda(\psi_\lambda^*) \le \frac{2}{\lambda(t+1)^2}\|\psi_\lambda^* - \psi^0\|^2$. Set $\psi^0 = \mathbf{0}$ and $\frac{2}{\lambda(t+1)^2}\|\psi_\lambda^*\|^2 \le \varepsilon$, then we get that, when $t \ge \sqrt{\frac{2\|\psi_\lambda^*\|^2}{\lambda\varepsilon}}$, we have $\tilde{E}_\lambda(\psi^t) - \tilde{E}_\lambda(\psi_\lambda^*) \le \varepsilon$. □

With the above analysis of the convergence of the *smooth Kantorovich functional* $\tilde{E}_\lambda(\psi^t)$, in the following we give the convergence analysis of the *original Kantorovich functional* $\tilde{E}(\psi^t)$ in Eqn. (9), where ψ^t is obtained by FISTA.

Theorem 11. *If $\lambda = \frac{\varepsilon}{2\log n}$, then for any $t \ge \frac{\sqrt{8\|\bar{C}\|^2 n \log n}}{\epsilon}$ with $\bar{C} = C_{\max} - \lambda \log \nu_{min}$, we have*

$$\tilde{E}(\psi^t) - \tilde{E}(\psi^*) < \varepsilon, \tag{20}$$

where ψ^t is the solver of $\tilde{E}_\lambda(\psi)$ after t steps in the iterations in Alg. 1, and ψ^* is the optimizer of $\tilde{E}(\psi)$. Then the total computational complexity is $O(\frac{n^{2.5}\sqrt{\log n}}{\varepsilon})$.

Proof. We set the initial condition $\psi^0 = \mathbf{0}$. For any given $\varepsilon > 0$, we choose iteration step t, such that $\frac{2}{\lambda(t+1)^2}\|\psi_\lambda^*\|^2 \le \frac{\varepsilon}{2}$, $t \ge \frac{\sqrt{8\|\psi_\lambda^*\|^2 \log n}}{\epsilon}$, where ψ_λ^* is the optimizer of $\tilde{E}_\lambda(\psi)$. By theorem 9, we have

$$\begin{aligned} \tilde{E}_\lambda(\psi^t) - \tilde{E}(\psi^*) &\le \tilde{E}_\lambda(\psi^t) - \tilde{E}_\lambda(\psi^*) \\ &\le \tilde{E}_\lambda(\psi^t) - \tilde{E}_\lambda(\psi_\lambda^*) \\ &\le \frac{2}{\lambda(t+1)^2}\|\psi_\lambda^*\|^2 \\ &\le \frac{\varepsilon}{2} \end{aligned}$$

By Eqn. (16), we have

$$\begin{aligned} \tilde{E}(\psi^t) - \tilde{E}(\psi^*) &= E(\psi^t) + I_H(\psi^t) - E(\psi^*) - I_H(\psi^*) \\ &= (E(\psi^t) - E_\lambda(\psi^t)) + (E_\lambda(\psi^t) + I_H(\psi^t)) \\ &\quad - (E(\psi^*) + I_H(\psi^*)) \\ &\le |E(\psi^t) - E_\lambda(\psi^t)| + (\tilde{E}_\lambda(\psi^t) - \tilde{E}(\psi^*)) \\ &\le \lambda \log n + \frac{\varepsilon}{2} \\ &= \varepsilon \end{aligned}$$

Next we show that $\|\psi_\lambda^*\|^2 \le n\|\bar{C}\|^2$ by proving $|(\psi_\lambda^*)_j| \le \bar{C}\ \forall j \in [n]$. According to Eq. (15),

$$\begin{aligned} \nu_j &= \sum_{i=1}^m \mu_i \frac{e^{((\psi_\lambda^*)_j - c_{ij})/\lambda}}{\sum_{k=1}^n e^{((\psi_\lambda^*)_k - c_{ik})/\lambda}} \\ &= \sum_{i=1}^m \mu_i \frac{e^{(\psi_\lambda^*)_j/\lambda}}{\sum_{k=1}^n e^{(\psi_\lambda^*)_k/\lambda} e^{(c_{ij} - c_{ik})/\lambda}} \end{aligned} \tag{21}$$

Assume $(\psi_\lambda^*)_{\max}$ is the maximal element of ψ_λ^*, we have $\sum_{k=1}^n e^{(\psi_\lambda^*)_k/\lambda} e^{(c_{ij} - c_{ik})/\lambda} \ge e^{(\psi_\lambda^*)_{\max}/\lambda} e^{-C_{\max}/\lambda}$, where $C_{\max}$ is the maximal element of the matrix C. Thus,

$$\begin{aligned} \nu_j &\le \sum_{i=1}^m \mu_i \frac{e^{(\psi_\lambda^*)_j/\lambda}}{e^{(\psi_\lambda^*)_{\max}/\lambda} e^{-C_{\max}/\lambda}} \\ &= \frac{e^{(\psi_\lambda^*)_j/\lambda}}{e^{(\psi_\lambda^*)_{\max}/\lambda} e^{-C_{\max}/\lambda}} \end{aligned}$$

Algorithm 1: Accelerated gradient descent for OT

Input: The cost matrix $C = (c_{ij})$, the corresponding source weights μ and target weights ν, the approximate parameter λ, and the step length η.
Output: The smoothed Kantorovich functional ψ_λ.
Initialize $\psi = (\psi_1, \psi_2, \ldots, \psi_n) \leftarrow (0, 0, \ldots, 0)$.
Initialize $z \leftarrow (0, 0, \ldots, 0)$.
Initialize $K = e^{-C/\lambda}$, $\theta_0 = 1$
repeat
  $v^t = e^{\psi^t/\lambda}$.
  $\nabla E_\lambda(\psi^t) = diag(v)K^T(\mu \oslash Kv) - \nu$.
  $z^{t+1} = \psi^t - \eta \nabla E_\lambda(\psi^t)$
  $z^{t+1} = z^{t+1} - mean(z^{t+1})$.
  $\psi^{t+1} = z^{t+1} + \frac{\theta_t - 1}{\theta_{t+1}}(z^{t+1} - z^t)$.
  $\theta_{t+1} = \frac{1}{2}(1 + \sqrt{1 + 4\theta_t^2})$.
  $t = t + 1$
until Converge
The OT cost $E(\psi^t) = \sum_{i+1}^m \mu_i \psi^c(x_i) - \sum_{j=1}^n \psi_j^t \nu_j$.

Then, $(\psi^*_\lambda)_{\max} \le (\psi^*_\lambda)_j + C_{\max} - \lambda \log \nu_j$ and

$$\begin{aligned}(\psi^*_\lambda)_{\max} &\le \frac{1}{n}\sum_{j=1}^{n}\{(\psi^*_\lambda)_j + C_{\max} - \lambda \log \nu_j\} \\ &\le C_{\max} - \lambda \log \nu_{min}\end{aligned} \tag{22}$$

According to the inequality of arithmetic and geometric means, we have $\sum_{k=1}^{n} e^{(\psi^*_\lambda)_k/\lambda} \ge n e^{\frac{1}{n}(\sum_{k=1}^{n}(\psi^*_\lambda)_k/\lambda)} = n$. Thus, $\nu_j \le \frac{e^{(\psi^*_\lambda)_j/\lambda}}{n e^{-C_{\max}/\lambda}}$.

$$\begin{aligned}(\psi^*_\lambda)_j &\ge \lambda \log n - C_{\max} + \lambda \log \nu_j \\ &\ge \lambda \log \nu_{min} - C_{\max}\end{aligned} \tag{23}$$

Combine Eqn. (22) and (23), we have $|(\psi^*_\lambda)_j| \le C_{\max} - \lambda \log \nu_{min} = \bar{C}$. Hence, we obtain that when $t \ge \frac{\sqrt{8\|\bar{C}\|^2 n \log n}}{\epsilon}$, $\tilde{E}(\psi^t) - \tilde{E}(\psi^*) < \varepsilon$.

For each iteration in Eqn. (17), we need $O(n^2)$ times of operations, thus total the computational complexity of the proposed method is $O(\frac{n^{2.5}\sqrt{\log n}}{\varepsilon})$. □

Relationship with Softmax If there exists an OT map from μ to ν, then each sample x_i of the source distribution is classified into the corresponding $y_j = T(x_i)$. If there does not exist an OT map, we can only get the OT plan, which can be treated as a soft classification problem: each weighted sample x_i with weight μ_i will be sent to the corresponding y_js with weight $\mu_i \frac{P_{ij}}{\sum_{k=1}^{n} P_{ik}}$ where $P_{ij} > 0$. Here $P_{ij} = \mu_i \frac{P_{ij}}{\sum_{k=1}^{n} P_{ik}}$ gives the OT plan from the source to the target distribution. The smoothed OT plan given by minimizing the smooth Kantorovich functional can be further treated as a relaxed OT plan. Instead of sending the weights of a specific sample to several target samples, the smooth solver tends to send each source sample to all of the target samples weighted by $\frac{e^{(\psi^*_j - c_{ij})/\lambda}}{\sum_{k=1}^{n} e^{(\psi^*_k - c_{ik})/\lambda}}$. Sample x_i weighted by μ_i will be sent to y_j with weight $\mu_i \frac{e^{(\psi^*_j - c_{ij})/\lambda}}{\sum_{k=1}^{n} e^{(\psi^*_k - c_{ik})/\lambda}}$.

Relationship with entropy regularized OT problem The Sinkhorn algorithm is deduced from minimizing the entropy regularized OT problem (Cuturi 2013): $\langle P, C\rangle + \lambda KL(P|\mu \otimes \nu)$ with $P \in \pi(\mu, \nu)$. Its dual is given by (Genevay et al. 2016):

$$\begin{aligned}W_\lambda(\mu,\nu) = -\min_{\psi} \ \{&\lambda \sum_{i=1}^{m} \mu_i \log\left(\sum_{j=1}^{n} \nu_j e^{(\psi_j - c_{ij})/\lambda}\right) \\ &- \sum_{j=1}^{n} \nu_j \psi_j + \lambda\}\end{aligned} \tag{24}$$

with gradient $\frac{\partial W_\lambda}{\partial \psi_j} = \sum_{i=1}^{m} \mu_i \frac{\nu_j e^{(\psi_j - c_{ij})/\lambda}}{\sum_{k=1}^{n} \nu_k e^{(\psi_k - c_{ik})/\lambda}} - \nu_j$. With the optimal solver ψ^*, the approximate OT plan is given by $P_{ij} = \mu_i \frac{\nu_j e^{(\psi_j - c_{ij})/\lambda}}{\sum_{k=1}^{n} \nu_k e^{(\psi_k - c_{ik})/\lambda}}$. We can compare them with our gradient Eqn. (13) and approximated OT plan Eqn. (15) to see the subtle differences. Actually, by setting $\psi := \psi - \log \nu$, the minimizing problem in Eqn. (24) is equivalent to our smoothed semi-discrete problem of Eqn. (16) with a different constant term.

Furthermore, if we set $u = (u_1, u_2, \ldots, u_m)^T$ with $u_i = \frac{\mu_i}{K_i v}$ in Eqn. (15), the computed approximate OT plan can be rewritten as $P_\lambda = diag(u) K diag(v)$, which is the same as the form of the Sinkhorn solution (Cuturi 2013). Since the solution of the Sinkhorn algorithm is unique, we conclude that the induced approximate optimal transport plan Eqn. (15) by our algorithm is equivalent to that of the Sinkhorn.

Experiments

In this section, we investigate the performance of the proposed algorithm under different parameters, and then compare it with the Sinkhorn algorithm (Cuturi 2013). In the following, we first introduce the various settings of the experiments including the parameters, the cost matrix and the evaluation metrics. Then we show the experimental results. All of the codes were written in MATLAB with GPU acceleration, including the proposed method and the Sinkhorn algorithm (Cuturi 2013). The experiments are also conducted on a Windows laptop with Intel Core i7-7700HQ CPU, 16 GB memory and NVIDIA GTX 1060Ti GPU.

Parameters There are two parameters involved in the proposed algorithm, λ and η_t. The former is used to control the approximate accuracy between the Log-Sum-Exp function and the Kantorovich potential ψ^c in Eqn. (11), and the latter controls the step size of the FISTA algorithm in Eqn. (17). Basically, smaller λ gives better approximation.

In our experiments, to get λ as small as possible, based on the Property 1 of the Eqn. (6), we set the median of the cost matrix C equal to zero, so that the full range of the exponential of the floating-point numbers can be used, instead of only the negative part[1]. Thus we set $C = C - \frac{C_{\max}+C_{\min}}{2}$ and call it the *translation trick*. If the range of C is denoted as R, then the accuracy parameter is set to be $\lambda = \frac{R}{T}$, where T is a positive constant. For the FISTA algorithm, the ideal step size should be $\eta_t = \frac{1}{\sigma_{\max}}$, where $\sigma_{\max}$ is the maximal eigenvalue of the Hessian matrix $\nabla^2 \tilde{E}_\lambda(\psi)$ in Eqn. (14). By Nesterov smoothing, we know $\sigma_{\max} \le \frac{1}{\lambda}$, so we set the step length $\eta_t = \eta\lambda$, where η is a constant[2]. In practice we use (T, η) as control parameters instead of (λ, η_t).

Cost Matrix In the following experiments, we test the performance of the algorithm with different parameters under different metrics. Specifically, we set $\mu = \sum_{i=1}^{m} \mu_i \delta(x - x_i)$, $\nu = \sum_{j=1}^{n} \nu_j \delta(y - y_j)$. Note that after the settings of μ_is and ν_js, they are normalized by $\mu_i = \frac{\mu_i}{\sum_{k=1}^{m} \mu_k}$ and $\nu_j = \frac{\nu_j}{\sum_{k=1}^{n} \nu_k}$. To build the cost matrix, we use the Eu-

[1]For example, if double-precision floating-point format is used in 64-bit processors, the range of the number is about $2.2251e^{-308} \sim 1.7977e^{+308}$ when using MATLAB.

[2]For one thing, if λ is relatively large, only with small step size, the algorithm may run out of the precision range of the processor and thus get 'Inf' or 'NAN'. Thus, η may be far less that 1. For the other thing, we have $H \le \frac{1}{\lambda} \max_i(\frac{\max_j K_{ij} v_j}{\mathbf{K_i v}}) \le \frac{1}{\lambda}$, we may also choose $\eta > 1$ when λ itself is small.

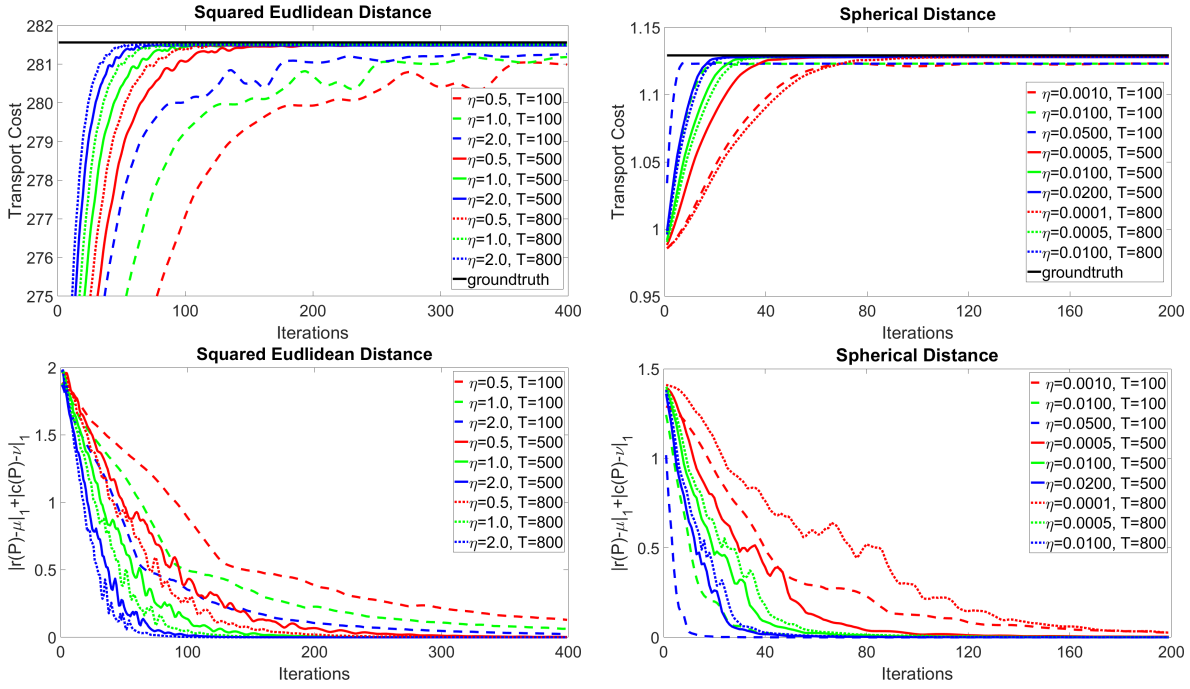

Figure 1: The performance of the proposed algorithm with different parameters.

clidean distance, squared Euclidean distance, spherical distance, and random cost matrix.

- For Euclidean distance (ED) and the squared Euclidean distance (SED) experiments, in experiment 1, x_i's are randomly generated from the Gaussian distribution $\mathcal{N}(3\mathbf{1_d}, I_d)$ and y_j's are randomly sampled from the uniform distribution $Uni([0,1]^d) - 5$. Both μ_i and ν_j are randomly generated from the uniform distribution $Uni([0,1])$. Experiment 3 also uses a similar sampling strategy to build the discrete source and target measures. In experiment 2, like (Altschuler, Niles-Weed, and Rigollet 2017), we randomly choose one pair of images from the MNIST dataset (LeCun and Cortes 1998), and then add negligible noise 0.01 to each background pixel with intensity 0. μ_i and x_i (ν_j and y_j) are set to be the value and the coordinate of each pixel in the source (target) image. Then the Euclidean distance and squared Euclidean distance between x_i and y_j are given by $c(x_i, y_j) = \|x_i - y_j\|$ and $c(x_i, y_j) = \|x_i - y_j\|^2$, respectively.
- For the spherical distance (SD) experiment, both μ_i and ν_j are randomly generated from the uniform distribution $Uni([0,1])$. x_i's are randomly generated from the Gaussian distribution $\mathcal{N}(3\mathbf{1_d}, I_d)$ and y_j's are randomly generated from the uniform distribution $Uni([0,1]^d)$. Then we normalize x_i and y_j by $x_i = \frac{x_i}{\|x_i\|_2}$ and $y_j = \frac{y_j}{\|y_j\|_2}$. As a result, both x_i's and y_j's are located on the sphere. The spherical distance is given by $c(x_i, y_j) = \arccos(\langle x_i, y_j \rangle)$.
- For the random distance (RD) matrix experiment, both μ_i and ν_j are randomly generated from the uniform distribution $Uni([0,1])$. Also, to build C, we randomly sample c_{ij} from the Gaussian distribution $\mathcal{N}(0,1)$, then C is defined as $C = C - C_{\min} + 1.0$.

Evaluation Metrics We use two metrics to evaluate the proposed method: the first one is the transport cost, which is defined by Eqn. (6) and is given by $-E(\psi)$; and the second is the L_1 distance from the computed transport plan P_λ to the admissible distribution space $\pi(\mu, \nu)$ defined in Eqn. (1), and the distance is defined as $D(P_\lambda) = \|P_\lambda \mathbf{1} - \mu\|_1 + \|P_\lambda^T \mathbf{1} - \nu\|_1$.

Experiment 1: The influence of different parameters We test the performance of the proposed algorithm with different parameters under the SED and SD with $m = n = 100$ and $d = 5$, as shown in Fig. 1. The left column shows the results for SED and the right column is the result for SD. The top row illustrates the transport costs over iterations, and the bottom row is the distance $D(P_\lambda)$.

In the top row of Fig. 1, the black lines give the groundtruth transport costs, which are computed by linear programming. It is obvious that for the same η, by increasing T (decreasing λ, see the different types of the lines with the same color), the approximate accuracy is improved, and the convergence rate is increased; if T (equivalently λ) is fixed, by increasing η (see the different colors of the lines with the same type), we increase the convergence speed.

Experiment 2: Faster Convergence For the experiments with ED and SED, the distributions come from the MNIST dataset (LeCun and Cortes 1998), as illustrated in the *Cost Matrix*. For the the experiments with SD and RD, we set $m = n = 500$ and $d = 5$. Then we compare with the

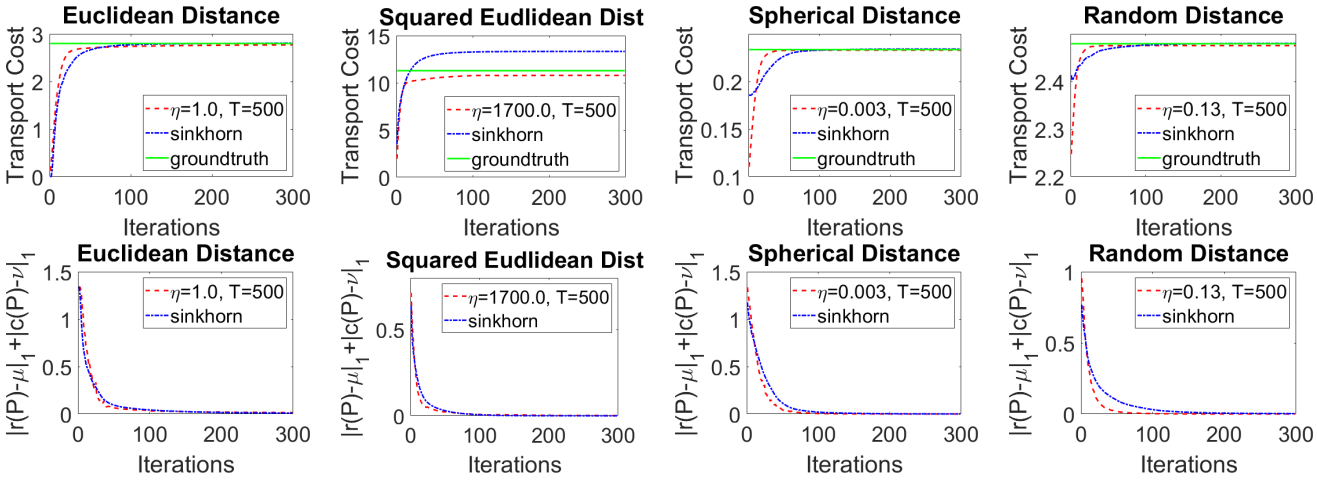

Figure 2: Comparison with the Sinkhorn algorithm (Cuturi 2013) under different cost matrix.

	Sink	Green	Screen	APDAMD	Ours
ED	0.0596	0.0923	0.0541	3.76	**0.0404**
SED	0.0431	0.0870	0.0328	3.21	**0.0197**
SD	0.0564	0.0862	0.0400	2.29	**0.0142**
RD	0.0374	0.0726	0.0313	2.88	**0.0227**

Table 1: Running time (s) of our method, Sinkhorn (Sink) (Cuturi 2013), Greenkhorn (Green) (Altschuler, Niles-Weed, and Rigollet 2017), Screenkhorn (Screen) (Alaya et al. 2019) and APDAMD (Lin, Ho, and Jordan 2019).

p	GT	Sink	Ours	|Sink-GT|	|Ours-GT|
1.5	103.33	103.51	103.27	0.18	**0.06**
2	281.7	282.5	281.6	0.8	**0.1**
3	2189.8	2197.1	2187.5	7.3	**2.3**
4	16951.4	17038.5	16932.0	87.1	**19.4**

Table 2: Comparison among the OT cost (GT) by linear programming, the Sinkhorn results (Cuturi 2013) denoted as 'Sink' and the results of the proposed method denoted as 'Ours' with $T = 500$ and different p.

Sinkhorn algorithm (Cuturi 2013) with respect to both the convergence rate and the approximation accuracy. We manually set $T = 500$ to get a good estimate of the OT cost, and then adjust η to get the best convergence speed of the proposed algorithm. For the purpose of fair comparison, we use the same cost matrix with the same *translation trick* and the same λ for the Sinkhorn algorithm, where we treat each update of v as one step. We summarize the results in Fig. 2, where the green curves represent the groundtruth computed by linear programming, the blue curves are for the Sinkhorn algorithm, and the red curves give the results of our method. It is obvious that in all of the four experiments, our method achieves faster convergence than the Sinkhorn algorithm. Note that the computed approximate transport plan of the Sinkhorn algorithm is intrinsically equivalent to our induced transport plan in Eqn. (15).

In Tab. 1, we report the running time of our method, Sinkhorn (Cuturi 2013), its variants algorithms, including Greenkhorn (Altschuler, Niles-Weed, and Rigollet 2017) and Screenkhorn (Alaya et al. 2019), and APDAMD (Lin, Ho, and Jordan 2019) for the four experiments shown in Fig. 2 with $T = 700$. The stop condition is set to be $|E(\psi^{t+1}) - E(\psi^t)|/|E(\psi^t)| < 10^{-3}$. For all of the experiments, we can see that our proposed method achieves the fastest convergence.

Experiment 3: Better Accuracy From Fig. 2, we can observe that $-E(\psi^*_\lambda)$ gives a comparable or better approximate of the OT cost than $\langle P^*_\lambda, C\rangle$ with the same small λ, especially for the L_p cost function $c(x, y) = \|x - y\|^p$ with $p > 1$, see the second column of Fig. 2 for an example of $p = 2$. To achieve ϵ precision, $\langle P^*_\lambda, C\rangle$ (equivalent to the Sinkhorn result) needs to set $\lambda = \frac{\epsilon}{4 \log n}$ (Dvurechensky, Gasnikov, and Kroshnin 2018), which is smaller than our requirement of $\lambda = \frac{\epsilon}{2 \log n}$ according to Thm. 11. Thus, with the same λ, the results of our algorithm should be more accurate than the Sinkhorn solutions. To verify this point, we give more examples in Tab. 2 with $p = 1.5,\ 2,\ 3$ and 4. Here we use the discrete measures similar to the squared Euclidean distance as stated in the Cost Matrix part, and set $m = n = 500$, $d = 5$. From the table, we can see that our method obtains more accurate results than Sinkhorn.

Conclusion

In this paper, we propose a novel algorithm based on Nesterov's smoothing technique to improve the accuracy for solving the discrete OT problem. The c-transform of the Kantorovich potential is approximated by the smooth Log-Sum-Exp function, and the smoothed Kantorovich functional can be solved by FISTA efficiently. Theoretically, the computational complexity of the proposed method is given by $O(n^{2.5}\sqrt{\log n}/\varepsilon)$, which is lower than current estimation of the Sinkhorn method. Experimentally, our results demonstrate that the proposed method achieves faster convergence and better accuracy than the Sinkhorn algorithm.

Acknowledgments

Lei has been supported by National Key R&D Program of China 2021YFA1003003, NSFC 61936002, NSFC 61772105 and NSFC 61720106005; An and Gu by NSF CMMI-1762287, NSF DMS-1737812, and NSF FAIN-2115095; Xu by NIH R21EB029733 and NIH R01LM012434.

References

Abid, B. K.; and Gower, R. M. 2018. Greedy stochastic algorithms for entropy-regularized optimal transport problems. In *AISTATS*.

Alaya, M. Z.; Berar, M.; Gasso, G.; and Rakotomamonjy, A. 2019. Screening Sinkhorn Algorithm for Regularized Optimal Transport. In *Advances in Neural Information Processing Systems 32*.

Altschuler, J.; Niles-Weed, J.; and Rigollet, P. 2017. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems 30*.

An, D.; Guo, Y.; Lei, N.; Luo, Z.; Yau, S.-T.; and Gu, X. 2020a. AE-OT: A new Generative Model based on extended semi-discrete optimal transport. In *International Conference on Learning Representations*.

An, D.; Guo, Y.; Zhang, M.; Qi, X.; Lei, N.; and Gu, X. 2020b. AE-OT-GAN: Training GANs from data specific latent distribution. In *European Conference on Computer Vision (ECCV)*, 548–564.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *ICML*, 214–223.

Beck, A.; and Teboulle, M. 2009. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1): 183–202.

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

Blanchet, J.; Jambulapati, A.; Kent, C.; and Sidford, A. 2018. Towards Optimal Running Times for Optimal Transport. In *arxiv:1810.07717*.

Blondel, M.; Seguy, V.; and Rolet, A. 2018. Smooth and Sparse Optimal Transport. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, 880–889.

Chakrabarty, D.; and Khanna, S. 2021. Better and Simpler Error Analysis of the Sinkhorn-Knopp Algorithm for Matrix Scaling. *Mathematical Programming*, 188(1): 395–407.

Courty, N.; Flamary, R.; Tuia, D.; and Rakotomamonjy, A. 2017. Optimal Transport for Domain Adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9): 1853–1865.

Cuturi, M. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transportation Distances. In *International Conference on Neural Information Processing Systems*, volume 26, 2292–2300.

Dvurechensky, P.; Gasnikov, A.; and Kroshnin, A. 2018. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm. In *International Conference on Machine Learning*, 1367–1376.

Galichon, A. 2016. *Optimal Transport Methods in Economics*. Princeton University Press.

Genevay, A.; Cuturi, M.; Peyré, G.; and Bach, F. 2016. Stochastic optimization for large-scale optimal transport. In *Advances in Neural Information Processing Systems*, 3440–3448.

Gerber, S.; and Maggioni, M. 2017. Multiscale Strategies for Computing Optimal Transport. *Journal of Machine Learning Research*.

Glimm, T.; and Oliker, V. 2003. Optical design of single reflector systems and the Monge–Kantorovich mass transfer problem. *Journal of Mathematical Sciences*, 117(3): 4096–4108.

Guo, W.; Ho, N.; and Jordan, M. 2020. Fast Algorithms for Computational Optimal Transport and Wasserstein Barycenter. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2088–2097.

Horn, T. A.; and Johnson, C. R. 1991. *Topics in Matrix Analysis*. Cambridge.

Jambulapati, A.; Sidford, A.; and Tian, K. 2019. A Direct $\tilde{O}(1/\epsilon)$ Iteration Parallel Algorithm for Optimal Transport. In *International Conference on Neural Information Processing System*.

Jordan, R.; Kinderlehrer, D.; and Otto, F. 1998. The variational formulation of the Fokker–Planck equation. *SIAM Journal on Mathematical Analysis*, 29(1): 1–17.

Kusner, M.; Sun, Y.; Kolkin, N.; and Weinberger, K. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, 957–966.

LeCun, Y.; and Cortes, C. 1998. MNIST handwritten digit database. In *Proceedings of the IEEE*, 2278–2324.

Lee, Y. T.; and Sidford, A. 2014. Path finding methods for linear programming: Solving linear programs in $\tilde{O}$(sqrt(rank))iterations and faster algorithms for maximum flow. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, 424–433.

Lei, N.; An, D.; Guo, Y.; Su, K.; Liu, S.; Luo, Z.; Yau, S.-T.; and Gu, X. 2020. A Geometric Understanding of Deep Learning. *Engineering*, 6(3): 361–374.

Lin, T.; Ho, N.; and Jordan, M. 2019. On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms. In *International Conference on Machine Learning*, 3982–3991.

Meng, C.; Ke, Y.; Zhang, J.; Zhang, M.; Zhong, W.; and Ma, P. 2019. Large-scale optimal transport map estimation using projection pursuit. In *Advances in Neural Information Processing Systems 32*.

Nesterov, Y. 2005. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1): 127–152.

Nguyen, X. 2013. Convergence of latent mixing measures in finite and infinite mixture models. *Ann. Statist*, 41: 370–400.

Parikh, N.; and Boyd, S. 2014. *Proximal Algorithms*. Foundations and Trends in Optimization.

Pele, O.; and Werman, M. 2009. Fast and robust earth mover's distances. In *2009 IEEE 12th International Conference on Computer Vision (ICCV)*, 460–467. IEEE.

Peyré, G.; and Cuturi, M. 2018. *Computational Optimal Transport*. https://arxiv.org/abs/1803.00567.

Quanrud, K. 2018. Approximating optimal transport with linear programs. In *arXiv:1810.05957*.

Schiebinger, G.; Shu, J.; Tabaka, M.; Cleary, B.; Subramanian, V.; Solomon, A.; Gould, J.; Liu, S.; Lin, S.; Berube, P.; Lee, L.; Chen, J.; Brumbaugh, J.; Rigollet, P.; Hochedlinger, K.; Jaenisch, R.; Regev, A.; and Lander, E. 2019. Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming. *Cell*, 176(4): 928–943.

Schmitzer, B. 2019. Stabilized Sparse Scaling Algorithms for Entropy Regularized Transport Problems. *SIAM Journal on Scientific Computing*, 41(3): A1443–A1481.

Tolstikhin, I.; Bousquet, O.; Gelly, S.; and Schoelkopf, B. 2018. Wasserstein Auto-Encoders. In *ICLR*.

Villani, C. 2008. *Optimal transport: old and new*, volume 338. Springer Science & Business Media.

Xie, Y.; Chen, M.; Jiang, H.; Zhao, T.; and Zha, H. 2019a. On Scalable and Efficient Computation of Large Scale Optimal Transport. In *Proceedings of the 36th International Conference on Machine Learning*, 6882–6892.

Xie, Y.; Wang, X.; Wang, R.; and Zha, H. 2019b. A Fast Proximal Point Method for Computing Wasserstein Distance. In *Conference on Uncertainty in Artificial Intelligence*, 433–453.

Yurochkin, M.; Claici, S.; Chien, E.; Mirzazadeh, F.; and Solomon, J. M. 2019. Hierarchical Optimal Transport for Document Representation. In *Advances in Neural Information Processing Systems 32*.