Department of Complexity Science and Engineering
Graduate School of Frontier Sciences
The University of Tokyo
東京大学大学院新領域創成科学研究科
複雑理工学専攻

2022年
令和3年度

Master's Thesis
修士論文

Robust computation of optimal transport by β-potential regularization
(β-ポテンシャル正則化による最適輸送問題の頑健な計算方法について)

Supervisor: Professor Masashi Sugiyama
指導教員：杉山　将　教授

2022年1月25日提出
Submitted January 25, 2022

Shintaro Nakamura
中村紳太郎
47-206096

Abstract

Machine learning is a data processing technique that automatically extracts essential information from data to allow computers to perform intelligent tasks (e.g., regression analysis, image classification, image generation, and decision making in non-stationary environments). An important approach to this is approximating the probability distribution that generated the data with the model we have in mind. In order to do so, it is necessary to define *closeness* between probability distributions.

In this thesis, we deal with the optimal transport (OT) problem, which has attracted much attention as a measure of *closeness* of probability distributions in the field of machine learning. Intuitively, OT can be formulated as viewing a probability distribution as a pile of sand and transferring one pile of sand to another pile of sand with minimal cost.

Due to the heavy computation of the ordinary OT, one of the common approaches to accelerate the computation of OT is to regularize the ordinary OT problem with an entropic penalty term and use the celebrated Sinkhorn algorithm to approximate the OT. However, since the Sinkhorn algorithm runs a projection associated with the Kullback-Leibler divergence; it is often vulnerable to outliers.

To overcome this problem, we propose regularizing OT with the β-potential term associated with the so-called β-divergence, which was developed in robust statistics. Our theoretical analysis reveals that the β-potential can prevent the probability mass from being transported to outliers. We experimentally demonstrate that the transport matrix computed with our algorithm helps estimate a probability distribution robustly even in the presence of outliers and can detect outliers from a contaminated dataset.

Acknowledgement

First of all, I would like to thank my supervisor, Professor Masashi Sugiyama, who provided us with a great environment to pursue research and always gave me an insightful perspective to work on research. I have also learned a lot of writing techniques from him.

Second, I want to thank all the members of the Sugiyama-Yokoya-Ishida laboratory. Especially, I would like to express my gratitude to my colleague, Mr.Han Bao, for encouraging advice and discussions. I am grateful to him for always being kind and polite in teaching me so much online, even under COVID-19 circumstances. As a prospective Ph.D. student, he is a splendid role model. Dr. Yoshihiro Nagano, Mr. Kento Nozawa, Mr. Takeshi Teshima, and Ms.Yuko Kuroki (an alumna of our laboratory) gave me many important tips on how to proceed with my research.

Finally, I want to thank my family. My parents have always been my most reliable supporters. Let alone with this work, my whole life and my journey for research would not have been started without them. I am also grateful to my brother, who is also a master's student, for giving me a lot of inspiration in pursuing academic studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we will first introduce optimal transport (OT) in the context of machine learning (ML) field. Subsequently, we introduce the celebrated Sinkhorn algorithm, which made OT popular in the ML field and show its vulnerability to outlier data. Finally, we will briefly introduce our approach to approximate OT robustly.

1.1 Optimal transport in machine learning

In the machine learning field, problems are often formulated by defining a discrepancy between probability distributions [2, 3, 4]. As a major discrepancy, the Kullback-Leibler (KL) divergence [5] has been widely used since minimizing the KL divergence of an empirical distribution from a parametric model corresponds to maximum likelihood estimation [6]. However, the KL-divergence suffers from some problems. For instance, the KL-divergence of p from q can not work as a discrepancy when the support of p is not completely included in the support of q because the KL-divergence of p from q will diverge to infinity. Moreover, the KL-divergence does not satisfy the axioms of the metrics in a probability space [7]. On the other hand, *optimal transport* (OT) [8] does not suffer from these problems. OT does not require any conditions on the support of probability distributions and thus is expected to be more stable than the KL-divergence, which means it does not diverge to infinity. In addition, OT of two distributions is a metric in the probability space and therefore OT defines a proper distance between histograms and probability measures [9]. By defining the distance structure between histograms, the barycenter of some histograms can be defined. Applications using the barycenter of histograms have been reported in image processing [10] and color modifications [11].

1.2 Vulnerability of optimal transport theory to outliers

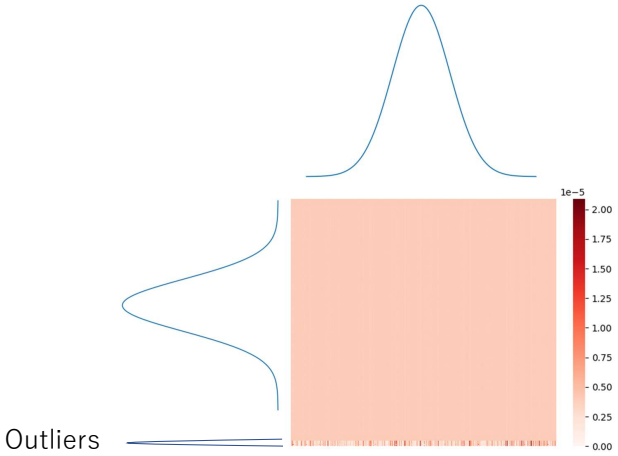
Due to heavy computation of the ordinary OT, one of the common approaches to accelerate computation of OT is to regularize the ordinary OT problem with an entropic penalty term (Boltzmann-Shannon entropy [12]) and use the Sinkhorn algorithm [13] to approximate the OT [14]. The entropic penalty makes the objective strictly convex, ensuring the existence of a unique global optimal solution. The Sinkhorn algorithm projects this global optimal solution onto a set of couplings in terms of the KL-divergence [12].

However, the Sinkhorn algorithm is sensitive to outliers and cannot approximate OT robustly. In our pilot study, we confirmed that the Sinkhorn algorithm cannot compute approximate OT robustly when outliers are included in the dataset. We first computed an approximated OT between $\hat{n}_1$ and $\hat{n}_2$, which are both sets of 500 samples drawn from $\mathcal{N}(0, 1)$. The approximate OT computed with the Sinkhorn algorithm was 0.44. Subsequently, we added 10 outliers which are all 70 to $\hat{n}_2$. Then, an approximated OT computed with the Sinkhorn algorithm drastically changed to 96.37. This is due to the coupling constraint of OT. An OT value changes drastically because probability mass has to be sent even to the outliers. Figure 1.1 shows the heatmap of the computed transport matrix with the Sinkhorn algorithm. We can see that it transports probability mass to the outliers.

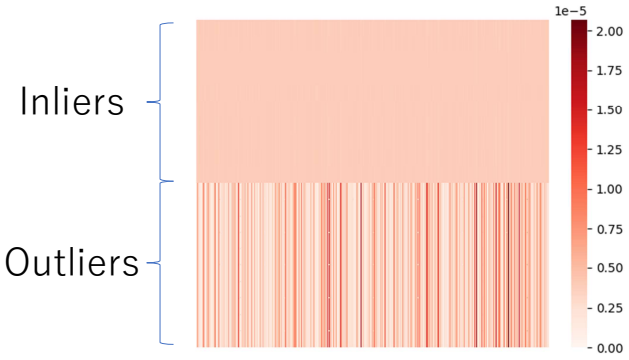
The sensitivity of the Sinkhorn algorithm may lead to undesired solutions in probabilistic modeling when we deal with noisy and adversarial datasets [15, 16]. In this thesis, we propose to mitigate the outlier sensitivity of the Sinkhorn algorithm by regularizing OT with the β-potential term instead of the Boltzmann-Shannon entropy [12]. This formulation can be regarded as a projection based on the β-divergence [17, 18, 19], approximately satisfying the coupling constraint. With some computational tricks, our algorithm is guaranteed not to move any probability mass to outliers (Figure 1.2). In the same pilot study above, the approximated OT computed with our algorithm was 0.008 before adding the outliers and after adding outlliers, it was 0.0078. Through numerical experiments, we demonstrate that our proposed method can be applied to estimating probability distributions ignoring outliers. Moreover, we show several applications of the proposed method such as the estimation of the reward distribution [20] in reinforcement learning (RL) and outlier detection.

1.3 Organization of our thesis

The rest of the thesis is organized as follows. In Chapter 2, we introduce the formulation of continuous and discrete OT and some previous works that have studied OT' robustness. In Chapter 3, we introduce the framework of convex regularization of discrete OT (CROT) [12]. More specifically, we review the theoretical background of the CROT, where we introduce the basics of convex analysis, the Bregman divergences, and the alternate Bregman projections. Subsequently, we formulate the CROT and present some algorithms to compute the solution to it. In Chapter 4, we first define outliers, which helps us approximate OT robustly. After that, we propose our algorithm, which robustly approximates OT, and we show its theoretical properties. In Chapter 5, we first show some toy experiments which demonstrate that our algorithm robustly approximates OT. Next, we show several applications of the proposed method, such as estimating the reward distribution [20] in reinforcement learning (RL) and outlier detection. Finally, we conclude our thesis in Chapter 6.

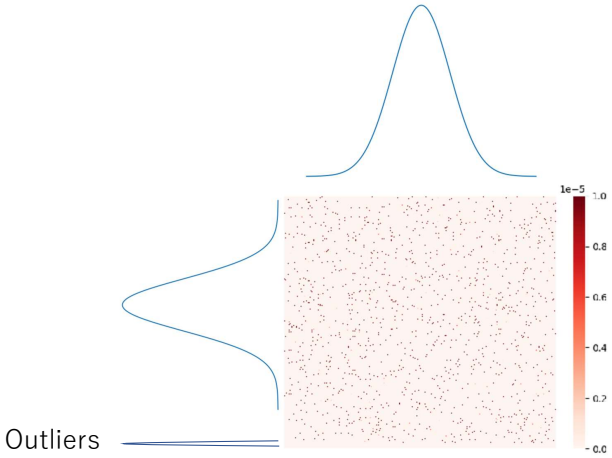


(a) The heatmap of the transport matrix computed with the Sinkhorn algorithm.

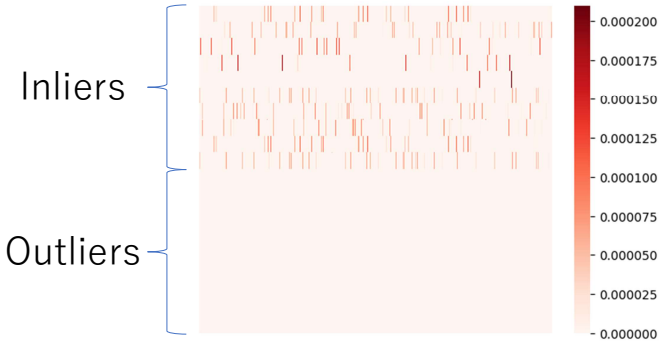


(b) The magnified version of the above figure of the outlier area. We can see that probability mass are moved to the outliers.

Figure 1.1: The heatmap of the transport matrix computed with the Sinkhorn algorithm. We computed two sets of samples $\hat{n}_1$ and $\hat{n}_2$ which are both 500 samples drawn from $\mathcal{N}(0, 1)$. One set has 10 oultliers which are all 70.



(a) The heatmap of the transport matrix computed with our algorithm.



(b) The magnified version of the above figure of the outlier area. We can see that our algorithm does not send any probability mass to outliers.

Figure 1.2: The heatmap of the transport matrix computed with our algorithm. We computed two sets of samples $\hat{n}_1$ and $\hat{n}_2$ which are both 500 samples drawn from $\mathcal{N}(0, 1)$. One set has 10 oultliers which are all 70.

Chapter 2

Formulation of Optimal Transport and Previous Works

In this chapter, we first introduce the formulation of OT. Then, we present two groups of previous studies investigating the robustness of OT. One is an approach that exploits the duality of OT. The other is to regularize OT with a total variation norm and derive an equivalent formulation that can be computed efficiently. Finally, we introduce a study that presents a unified framework for convex regularization of discrete OT, on which our work is based. We explain why this framework overcomes existing methods' shortcomings and helps compute OT robustly.

2.1 Formulation of optimal trasport problem

In this section, we show the continuous formulation of OT and the discrete formulation of OT. In general, if we say continuous optimal transport, we mean the classical Kantorovich formulation [21], a relaxed formulation of the original form introduced by Monge [22]. Since Monge's formulation has proved hard to study [9], we will discuss the Kantorovich formulation in this thesis as many studies in the machine learning field do [23, 1, 24, 25, 26, 27].

2.1.1 Formulation of continuous OT

Continuous OT is a problem to obtain the minimum-cost way to transport a mass from a probability distribution u on $\mathcal{X}$ to another distribution v on $\mathcal{X}$. Formally, the Kantorovich formulation of OT is written as follows:

$$\begin{aligned} \mathcal{W}(u,v) &:= \min_{\Pi \in \mathcal{F}(u,v)} \mathbb{E}_{X_1,X_2\sim\Pi}[c(X_1,X_2)] \\ &= \min_{\Pi \in \mathcal{F}(u,v)} \int\int c(x,y)\Pi(x,y)\mathrm{d}x\mathrm{d}y, \end{aligned} \tag{2.1}$$

where $\mathcal{F}(u,v)$ is the set of couplings between u and v (probibility distributions on $\mathcal{X}\times\mathcal{X}$ whose marginals are u and v) and c is a cost function. In this thesis, we assume $c(x,y)$ to

satisfy the axiom of distance and non-negativity as follows:

$$\begin{aligned} c(x,y) &= 0 \iff x = y, && (2.2)\\ c(x,y) &= c(y,x), && (2.3)\\ c(x,y) &\leq c(x,z) + c(z,y), && (2.4)\\ c(x,y) &\geq 0. && (2.5) \end{aligned}$$

Especially, in the machine learning field, the Wasserstein distance is often used [23, 28, 29]. The p-Wasserstein distance is defined as follows:

$$\mathcal{W}_p(u,v) = \min_{\Pi \in \mathcal{F}(u,v)} \Big(\int \int ||x-y||^p \Pi(x,y) \mathrm{d}x \mathrm{d}y \Big)^{\frac{1}{p}}, \tag{2.6}$$

where $p \in [1, \infty)$.

2.1.2 Formulation of discrete OT

Intuitively, discrete optimal tranport is about delivering items from $m \in \mathbb{N}$ suppliers to $n \in \mathbb{N}$ consumers. Each supplier and consumer has supply $\frac{1}{m}$ and demand $\frac{1}{n}$ respectively. Suppose we have two sets of independent samples drawn from two distributions P_x and P_y which are defined on the same domain and let them be $\{\boldsymbol{x}_i\}_{i=1}^m$ and $\{\boldsymbol{y}_j\}_{j=1}^n$. We write the empirical probability measures from these two samples by $\hat{P}_x \coloneqq \frac{1}{m}\sum_{i=1}^m \boldsymbol{x}_i \delta_{\boldsymbol{x}_i}$ and $\hat{P}_y \coloneqq \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}_i \delta_{\boldsymbol{y}_i}$, where $\delta_{\boldsymbol{x}}$ is the delta function at position $\boldsymbol{x}$. We denote a distance between samples by a non-negative function $h(\boldsymbol{x}_i - \boldsymbol{y}_j)$ and let $\boldsymbol{\gamma} \in \mathbb{R}_+^{m \times n}$ ($\gamma_{ij} = h(\boldsymbol{x}_i - \boldsymbol{y}_j)$ be the distance matrix. We also define a transport matrix by $\boldsymbol{\pi} \in \{\boldsymbol{T} | \boldsymbol{T} \in \mathbb{R}_+^{m \times n}, \boldsymbol{T}\mathbf{1} = \frac{\mathbf{1}}{m}, \boldsymbol{T}^\top \mathbf{1} = \frac{\mathbf{1}}{n}\} =: \mathcal{G}(\frac{\mathbf{1}}{m}, \frac{\mathbf{1}}{n})$, where $\mathbb{R}_+^{m \times n}$ is a set of real numbers larger than 0 and $\mathbf{1}$ is a vector whose elements are all 1, and $\boldsymbol{\pi}^\top$ is the transpose of $\boldsymbol{\pi}$. Then, OT between the two empirical distributions $\hat{P}_x$ and $\hat{P}_y$ is defined as follows [9]:

$$\mathrm{OT}(\hat{P}_x | \hat{P}_y) \coloneqq \min_{\boldsymbol{\pi} \in \mathcal{G}(\frac{\mathbf{1}}{m}, \frac{\mathbf{1}}{n})} \sum_{i,j} \pi_{ij} \gamma_{ij}. \tag{2.7}$$

In order to keep the notation concise, we denote the Frobenius inner product between two matrices $\boldsymbol{\pi}, \boldsymbol{\gamma} \in \mathbb{R}^{m \times n}$ by

$$\langle \boldsymbol{\pi}, \boldsymbol{\gamma} \rangle \coloneqq \sum_{i,j} \pi_{i,j} \gamma_{i,j}. \tag{2.8}$$

We can say that OT is the Frobenius inner product between two matrices.

2.2 Previous works to compute approximate OT robustly

In this section, we show some previous works which have studied the robustness of OT [30, 31, 1, 24, 25]. Some of them focused on making robust methods to compute approximate OT that works well in machine learning applications [1, 24, 25]. Since the vulnerability is due to the coupling constraint of the transport matrix, they all took an approach to relax it. Balaji et al. [1] and Staerman et al. [24] have introduced methods that leverage dual formulations of their subspecies of OT. Mukerjee et al. [25] have introduced a formulation of OT called ROBOT (ROBust Optimal Transport), which regularizes the OT with a total-variation norm.

2.2.1 Methods using the dual formulation

Chizat et al. [30] have introduced the so-called unbalanced OT, which relaxes the marginal constraint of OT. For instance, they have proposed one such relaxation using the f-divergence on marginal distributions, which are defined as follows:

$$\mathcal{W}^{\text{ub}}(u,v) \coloneqq \min_{\Pi \in \mathcal{F}_f(\tilde{u},\tilde{v})} \int c(x,y)\Pi(x,y)\mathrm{d}x\mathrm{d}y + \mathcal{D}_f(\tilde{u}||u) + \mathcal{D}_f(\tilde{v}||v), \tag{2.9}$$

where $\mathcal{D}_f$ is the f-divergence between distributions, defined as $\mathcal{D}_f(P||Q) = \int f(\frac{dP}{dQ})dQ$. Furthermore, Liero et al. [31] derived a dual form for the problem. Let f be a convex lower semi-continuous function. Define $r^*(x) \coloneqq \sup_{a>0} \frac{x-f(a)}{a}$. Then,

$$\begin{aligned} \mathcal{W}^{\text{ub}}(u,v) = \max_{f,g} \int f(x)u(x)\mathrm{d}x + \int g(y)v(y)\mathrm{d}y \\ \text{s.t. } r^*(f(x)) + r^*(g(y)) \le c(x,y), \end{aligned} \tag{2.10}$$

where f and g are 1-Lipschitz fuctions. In practice, f and g are implemented using a neural network and the 1-Lipschitz constraint is enforced using weight clipping [32] or a penalty on the gradients [33]. Hence, one benefit to consider the dual problem is that one can easily implement an end-to-end scheme in machine learning applications.

Balajit et al. [1] have shown that using the dual optimization (2.10) in large-scale deep learning applications such as the Wasserstein GAN [23] results in poor convergence and unstable behavior. They introduced a slightly different form of (2.9) as follows:

$$\begin{aligned} \mathcal{W}^{rob}_{\rho_1,\rho_2}(u,v) \coloneqq \min_{\tilde{u},\tilde{v}\in \text{Prob}(\mathcal{X})} \min_{\pi \in \Pi(u,v)} \int\int c(x,y)\pi(x,y)\mathrm{d}x\mathrm{d}y \\ s.t. \mathcal{D}_f(\tilde{u}||u) \le \rho_1,\ \mathcal{D}_f(\tilde{v}||v) \le \rho_2. \end{aligned} \tag{2.11}$$

Here, Prob($\mathcal{X}$) denote the space of probability measures defined on $\mathcal{X}$. In this formulation, one optimizes over the couplings whose marginal constraints are the relaxed distributions $\tilde{u}$ and $\tilde{v}$. To prevent over-relaxation of the marginals, they imposed a constraint that the f-divergence between the relaxed and the true marginals are bounded by constraints ρ_1 and ρ_2. Furthermore, they derived the dual form of this formulation as follows:

$$\begin{aligned} \mathcal{W}^{\text{rob}}_{\rho_1,\rho_2}(u,v) = \min_{\tilde{u},\tilde{v}} \max_{D\in 1-\text{Lip}} \int D(x)\tilde{u}(x)\mathrm{d}x - \int D(y)\tilde{v}(y)\mathrm{d}y \\ s.t. \mathcal{D}_f(\tilde{u}||u) \le \rho_1,\ \mathcal{D}_f(\tilde{v}||v) \le \rho_2. \end{aligned} \tag{2.12}$$

Here, 1-Lip denotes the set of all 1-Lipshitz functions. However, there are still shortcomings in this method. One has to choose hyper-parameters ρ_1 and ρ_2 properly to compute the OT robustly. It can only be estimated when the proportion of the contamination data is known [1], which is a rare case. Moreover, their method is based on the optimization package CVXPY [34] and does not scale to large samples, which yields a serious computational bottleneck.

Staerman et al. [24] introduced an idea to use Median-of-Means (MoM) [35, 36, 37] in OT. Specifically, they have leveraged the Kantorovich-Rubinstein formulation [38], a dual formulation of the 1-Wasserstein distance as follows:

$$\mathcal{W}_1(u,v) = \sup_{f\in\mathcal{B}_L} \mathbb{E}_{X\sim u}[f(X)] - \mathbb{E}_{X\sim v}[f(X)], \tag{2.13}$$

where $\mathcal{B}_L$ is the unit ball of the Lipschitz functions space. Their main contribution was to estimate $\mathbb{E}_{X\sim u}[f(X)]$ and $\mathbb{E}_{X\sim v}[f(X)]$ robustly. Given an i.i.d. sample $\mathbf{X} = \{X_1, \ldots, X_n\}$ drawn from u, the MoM estimator of $\mathbb{E}_{X\sim u}[f(X)]$ is built as follows. First, choose $\mathrm{K}_{\mathbf{X}} \leq n$, and partition $\mathbf{X}$ into $\mathrm{K}_{\mathbf{X}}$ disjoint blocks $\mathcal{B}_1^{\mathbf{X}}, \ldots, \mathcal{B}_{\mathrm{K}_{\mathbf{X}}}^{\mathbf{X}}$ of size $\mathrm{B}_{\mathbf{X}} = n/\mathrm{K}_{\mathbf{X}}$. Then, emprical means are computed on each of the $\mathrm{K}_{\mathbf{X}}$ blocks and the estimator returned is finally the median of the empirical means. Formally, the MoM estimator of $\mathbb{E}_{X\sim u}[f(X)]$ is given as follows:

$$\mathrm{MoM}_{\mathbf{X}}[\Phi] = \underset{1\leq k\leq \mathrm{K}_{\mathbf{X}}}{\mathrm{med}} \Big\{ \frac{1}{\mathrm{B}_{\mathbf{X}}} \sum_{i\in\mathcal{B}_k^{\mathbf{X}}} f(X_i) \Big\}. \tag{2.14}$$

The same is true for estimating $\mathbb{E}_{X\sim v}[f(X)]$.

However, the primal optimization problem is unclear in this method and therefore hard to interpret as OT. Moreover, it can not compute the transport matrix explicitly.

2.2.2 Regularization of OT with a total-variation norm

Mukherjee et al. [25] have proposed a formulation to allow for modifications of u in (2.1), while penalizing their magnitude and ensuring that the modified u is still a probability measure. Their optimization problem titled ROBOT (ROBust Optimal Transport) is formally written as follows:

$$\begin{aligned} & \min_{\Pi,s} \int\int c(x,y)\Pi(x,y)\mathrm{d}x\mathrm{d}y + \lambda||s||_{\mathrm{TV}} \\ \text{s.t.} \quad & \int \pi(x,y)dy = u(x) + s(x) \geq 0, \\ & \int \pi(x,y)dx = v(y), \\ & \int s(x)dx = 0, \end{aligned} \tag{2.15}$$

where $||\cdot||_{\mathrm{TV}}$ is the total-variation norm defined as $||u||_{\mathrm{TV}} = \frac{1}{2}\int |u(x)|dx$. The first and the last constraints ensure that $u+s$ is a valid probability measure, while $\lambda||s||_{\mathrm{TV}}$ penallizes the amount of modifications in u. One can identify exact locations in u by inspecting $u+s$, i.e.. if $u(x)+s(x)=0$, then x is an outlier.

Since (2.15) has constraints, it is hard to compute efficiently. They introduced an equivalent formulation of (2.15) to remedy this issue. The equivalent formulation is as follows:

$$\min_{\Pi\in\mathcal{F}(u,v)} \mathbb{E}_{(X,Y)\sim\Pi}[C_\lambda(X,Y)], \tag{2.16}$$

where C_λ is the truncated cost function defined as $C_\lambda(x,y) = \min\{c(x,y), 2\lambda\}$. This formulation enables one to use existing stochastic optimization algorithms to compute large-scale OT [39, 40].

One of their main results for applications to machine learning tasks was outlier detection. They used a small threshold to compare this to the modified mass and identify outliers. However, this threshold can be chosen arbitrarily. We have to choose this threshold properly, but there is no theoretical guarantee to estimate the proper value in advance.

2.3 Previous works of convex regularization of discrete OT

No matter what algorithms one uses — the network simplex [41] or interior point methods [42] — the cost of computing discrete optimal transport (2.7) scales at least in $O(d^3 \log d)$

when comparing two empirical distributions of dimension d. In order to remedy this issue, Cuturi [14] has proposed to regularize (2.7) with an entropic penalty term as follows:

$$\min_{\boldsymbol{\pi} \in \mathcal{G}(\frac{\mathbf{1}}{m}, \frac{\mathbf{1}}{n})} \langle \boldsymbol{\pi}, \boldsymbol{\gamma} \rangle - \frac{1}{\lambda} h(\boldsymbol{\pi}), \tag{2.17}$$

where $\lambda > 0$ is a hyper-parameter and $h(\boldsymbol{\pi}) \coloneqq -\sum_{i,j} \pi_{i,j} \log \pi_{i,j}$ is an entropic penalty term. Cuturi also [14] showed that a matrix scaling algorithm named the Sinkhorn algorithm computes the optimal solution for (2.17) with a much cheaper cost than existing methods to compute (2.7). They showed that the Sinkhorn algorithm works well in MNIST [43] classification tasks.

After the appearance of the Sinkhorn algorithm, Dessein et al. [12] have presented a unified framework for smooth convex regularization of discrete OT. They showed that the their convex regularized OT (CROT) turns out to be matrix nearness problem with respect to Bregman divergences[1] [12]. For instance, the Sinkhorn algorithm is running a projection in a Kullback-Leibler (KL) divergence [5] space.

In this thesis, we leverage this framework. Using this framework, we can consider running a robust projection in a certain space and obtain a transport matrix explicitly for large-scale data. Specifically, we regularize OT with a β-potential term which enables us to run a *pseudo* projection in a β-divergence [17] space. We call it *pseudo* because our pseudo projection does not satisfy the coupling constraint. With some computational tricks and leverage of the domain of the Fenchel conjugate of the β-potential, our algorithm is guaranteed not to move any probability mass to outliers which leads to robustness in computing OT.

[1] In this thesis, the matrix nearness problem is defined as finding the nearest member of some given class of matrices for any arbitrary matrix, where distance is measured with Bregman divergence. Since Bregman divergence does not satisfy the trigonometric inequality, it cannot strictly be called a distance, but since we only care about the relationship between two matrices, we will call it a distance here.

Chapter 3

Convex Regularization of Discrete OT

In this chapter, we introduce discrete OT regularized by a type of convex function called the Legendre type [44, 12]. We first study the required theoretical backgrounds for the convex regularized discrete OT (CROT) framework [12]. We then show the framework of CROT, which contains the formulation of CROT and efficient algorithms to compute it.

3.1 Notations

We denote by $\mathbb{R}$ and $\mathbb{N}$ the set of real numbers and natural numbers, respectively. We denote an arbitrary size of two-dimensional zero matrices by $\mathbf{0}$ and matrices full of ones by $\mathbf{1}$. When the intended meaning is clear from the context, we also use $\mathbf{0}$ for a zero vector of any dimension and $\mathbf{1}$ for a vector of any dimension full of ones. The notation $\cdot^\top$ represents the transposition operator for matrices and vectors. Functions of a real variable, such as exponential or power functions, are considered element-wise when applied to matrices. The max operator between matrices is also applied in an element-wise manner. Matrix divisions are similarly considered element-wise, whereas element-wise matrix multiplications, also known as the Hadamard or Schur product, are denoted by $\odot$ to remove any ambiguity with standard matrix multiplications. Lastly, the addition or subtraction of a scalar and a matrix should be understood element-wise by replicating the scalar.

3.2 Theortical background

In this section, we review the theoretical background of the CROT framework [12].

3.2.1 Convex analysis

Let $\mathcal{E}$ be a Euclidean space with inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \|$. We denote the boundary, interior and relalative interior of a subset $\mathcal{S} \subseteq \mathcal{E}$ by $\mathrm{bd}(\mathcal{S})$, $\mathrm{int}(\mathcal{S})$ and $\mathrm{ri}(\mathcal{S})$ respectively. Recall that for a convex set $\mathcal{C}$, we have

$$\mathrm{ri}(\mathcal{C}) = \{\boldsymbol{x} \in \mathcal{E} \mid \forall \boldsymbol{y} \in \mathcal{C},\ \exists \lambda > 1,\ \lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in \mathcal{C}\}. \tag{3.1}$$

In convex analysis, scalar functions are defined over the whole space $\mathcal{E}$ and take values in $\mathbb{R} \cup \{-\infty, \infty\}$. The effective domain, or simply domain, of a function f is defined as the set:

$$\operatorname{dom} f = \{\boldsymbol{x} \in \mathcal{E} \mid f(\boldsymbol{x}) < +\infty\}. \tag{3.2}$$

Definition 1 (Closed functions)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be closed if for each $\alpha \in \mathbb{R}$, the sublevel set $\{\boldsymbol{x} \in \operatorname{dom} f \mid f(\boldsymbol{x}) \leq \alpha\}$ is a closed set.*

If $\operatorname{dom} f$ is closed, then f is closed.

Definition 2 (Proper functions)**.** *Suppose a convex function $f : \mathcal{E} \to \mathbb{R} \cup \{\pm\infty\}$ satisfies $f(\boldsymbol{x}) > -\infty$ for every $\boldsymbol{x} \in \operatorname{dom} f$ and there exists some point $\boldsymbol{x}_0$ in its domain such that $f(\boldsymbol{x}_0) < +\infty$. Then f is called a proper function.*

A proper convex function is closed if and only if it is lower semi-continuous[1]. Moreover, a closed function f is continuous relative to any simplex, polytope of a polyhedral subset in $\operatorname{dom} f$ and a convex function f is always continuous in the relative interior $\operatorname{ri}(\operatorname{dom} f)$ of its domain.

Definition 3 (Essential smoothness [45])**.** *Suppose f is a closed convex proper function on $\mathcal{E}$ with $\operatorname{int}(\operatorname{dom} f) \neq \emptyset$. Then f is essentially smooth, if f is differentiable on $\operatorname{int}(\operatorname{dom} f)$ and*

$$\left.\begin{array}{c} \forall n \in \mathbb{N},\ x_n \in \operatorname{int}(\operatorname{dom} f), \\ x_n \to x \in \operatorname{bd}(\operatorname{dom} f) \end{array}\right\} \Rightarrow ||\nabla f(\boldsymbol{x}_n)|| \to \infty.$$

Definition 4 (Essential Strict Convexity [45])**.** *Here, we denote the subgradient of f by ∂f. Suppose f is closed convex proper on $\mathcal{E}$. Then, f is essentially strictly convex, if f is strictly convex on every convex subset of $\operatorname{dom} \partial f$.*

We define a set of functions called the Legendre type and Fenchel conjugate functions.

Definition 5 (Legendre type [45])**.** *Suppose f is a closed convex proper function on $\mathcal{E}$. Then, f is said to be of the Legendre type if f is both essentially smooth and essentially strictly convex.*

Definition 6 (Fenchel conjugate [12])**.** *The Fenchel conjugate f^* of a function f is defined for all $\boldsymbol{y} \in \mathcal{E}$ as follows:*

$$f^*(\boldsymbol{y}) = \sup_{\boldsymbol{x} \in \operatorname{int}(\operatorname{dom} f)} \langle \boldsymbol{x}, \boldsymbol{y} \rangle - f(\boldsymbol{x}). \tag{3.3}$$

The Fenchel conjugate f^* is always a closed convex function and if f is a closed convex function, then $(f^*)^* = f$, and f is of the Legendre type if and only if f^* is of the Legendre type. If f^* is of the Legendre type, the gradient mapping ∇f is a homeomorphism[2] between $\operatorname{int}(\operatorname{dom} f)$ and $\operatorname{int}(\operatorname{dom} f^*)$, with inverse mapping $(\nabla f)^{-1} = \nabla f^*$. This guarantees the existence of dual coordinate systems $\boldsymbol{x}(\boldsymbol{y}) = \nabla f^*(\boldsymbol{y})$ and $\boldsymbol{y}(\boldsymbol{x}) = \nabla f(\boldsymbol{x})$ on $\operatorname{int}(\operatorname{dom} f)$ and $\operatorname{int}(\operatorname{dom} f^*)$.

Finally, we say that a function f is a cofinite if it verifies

$$\lim_{\lambda \to +\infty} f(\lambda \boldsymbol{x})/\lambda = +\infty, \tag{3.4}$$

for all nonzero $\boldsymbol{x} \in \mathcal{E}$. Intuitively, it means that f grows super-linearly in every direction. In particular, a closed convex proper function is cofinite if and only if $\operatorname{dom} f^* = \mathcal{E}$.

[1] Let X be a topological space. A funtion $f : X \to \mathbb{R} \cup \{-\infty, \infty\}$ is called lower semicontinuous at a point $x_0 \in X$ if for every $y < f(x_0)$ there exists a neighborhood U of x_0 such that $f(x) > y$ for all $x \in U$.

[2] A function $f : X \to Y$ between two topological spaces is a homeomorphism if it has the following three properties: (a) f is a bijection. (b) f is continuous. (c) The inverse function f^{-1} is continous.

3.2.2 Bregman divergence

Let ϕ be a convex function on $\mathcal{E}$ that is differentiable on $\text{int}(\text{dom}\phi) \neq \emptyset$. The Bregman divergence generated by ϕ is defined as follows:

$$B_\phi(\boldsymbol{x}|\boldsymbol{y}) := \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) - \langle \boldsymbol{x} - \boldsymbol{y}, \nabla\phi(\boldsymbol{y})\rangle, \tag{3.5}$$

for all $\boldsymbol{x} \in \text{dom}\,\phi$ and $\boldsymbol{y} \in \text{dom}\,\phi$. In this thesis, we consider so-called a separable Bregman divergences [12]. It can be seen as an aggregation of element-wise Bregman divergences between scalars on $\mathbb{R}$:

$$B_\phi(\boldsymbol{\pi}|\boldsymbol{\xi}) = \sum_{i=1}^{m}\sum_{j=1}^{n} B_{\phi_{ij}}(\pi_{ij}||\xi_{ij}), \tag{3.6}$$

$$\phi(\boldsymbol{\pi}) = \sum_{i=1}^{m}\sum_{j=1}^{n} \phi_{ij}(\pi_{ij}), \tag{3.7}$$

for all $\boldsymbol{\pi}, \boldsymbol{\xi} \in \mathbb{R}^{m\times n}$. Below we choose the element-wise generator ϕ_{ij} to be equal for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

We have $B_\phi(\boldsymbol{x}||\boldsymbol{y}) \geq 0$ for any $\boldsymbol{x} \in \text{dom}\,\phi$ and $\boldsymbol{y} \in \text{dom}\,\phi$. If in addition ϕ is strictly convex on $\text{int}(\text{dom}\phi)$, then $B_\phi(\boldsymbol{x}||\boldsymbol{y}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{y}$. Bregman divergences are also always convex in the first argument and are invariant under adding an arbitrary affine term to their generator.

Suppose now that ϕ is of the Legendre type, and let $\mathcal{C} \subseteq \mathcal{E}$ be a closed convex set such that $\mathcal{C} \cap \text{int}(\text{dom}\phi) \neq \emptyset$. Then, for any point $\boldsymbol{y} \in \text{int}(\text{dom}\phi)$, the following problem,

$$P_\mathcal{C}(\boldsymbol{y}) = \underset{\boldsymbol{x}\in\mathcal{C}}{\text{argmin}}\, B_\phi(\boldsymbol{x}||\boldsymbol{y}), \tag{3.8}$$

has a unique solution which actually belongs to $\mathcal{C} \cap \text{int}(\text{dom}\phi)$. $P_\mathcal{C}(\boldsymbol{y})$ is called the Bregman projection of $\boldsymbol{y}$ onto $\mathcal{C}$ [12].

3.2.3 Alternate Bregman projections

Let ϕ be a function of the Legendre type with Fenchel conjugate $\phi^* = \psi$. In general, computing Bregman projections onto an arbitrary closed convex set $\mathcal{C} \subseteq \mathcal{E}$ such that $\mathcal{C} \cap \text{int}(\text{dom}\phi) \neq \emptyset$ is nontrivial [12]. Sometimes, it is possible to decompose $\mathcal{C}$ into the intersection of finitely many closed convex sets:

$$\mathcal{C} = \cap_{l=1}^{s}\mathcal{C}_l, \tag{3.9}$$

where the individual Bregman projections onto the respective sets $\mathcal{C}_1, \ldots, \mathcal{C}_s$ are easier to compute. It is then possible to obtain the Bregman projections onto $\mathcal{C}$ by alternate projections onto $\mathcal{C}_1, \ldots, \mathcal{C}_s$ according to Dykstra's algorithm [46].

In more detail, let $\sigma : \mathbb{N} \to \{1, \ldots, s\}$ be a control mapping that determines the sequence of subsets onto which we project. For a given point $\boldsymbol{x}_0 \in \text{int}(\text{dom}\,\phi)$, the Bregman projection $P_\mathcal{C}(\boldsymbol{x}_0)$ of $\boldsymbol{x}_0$ onto $\mathcal{C}$ can be approximated with Dykstra's algorithm by iterating the following updates:

$$\boldsymbol{x}_{k+1} \leftarrow P_{\mathcal{C}_{\sigma(k)}}(\nabla\psi(\nabla\phi(\boldsymbol{x}_k + \boldsymbol{y}^{\sigma(k)})), \tag{3.10}$$

where the correction term $\boldsymbol{y}^1, \ldots, \boldsymbol{y}^s$ for the respective subsets are initialized with the null element of $\mathcal{E}$, and are updated after projection as follows:

$$\boldsymbol{y}^{\sigma(k)} \leftarrow \boldsymbol{y}^{\sigma(k)} + \nabla\phi(\boldsymbol{x}_k) - \nabla\phi(\boldsymbol{x}_{k+1}). \tag{3.11}$$

Under some technical assumptions, the sequence of updates $(\boldsymbol{x}_k)_{k\in\mathbb{N}}$ converges in norm to $P_{\mathcal{C}}(\boldsymbol{x_0})$ with a linear rate. Several sets of such conditions have been studied [47, 48, 49]. As in the study of Dessein et al. [12], we use the conditions proposed by Dhillon et al. [47] for the CROT framework. The convergence of Dykstra's algorithm is guaranteed as soon as the function ϕ is cofinite, the constraint qualification $\mathrm{ri}(\mathcal{C}_1) \cap \cdots \cap \mathrm{ri}(\mathcal{C}_s) \cap \mathrm{int}(\mathrm{dom}\phi) \neq \emptyset$ holds, and the control mapping σ is essentially cyclic. Here, being essentially cyclic means that there exists a number $t \in \mathbb{N}$ such that σ takes each output value at least once during any t consecutive input values. If a given $\mathcal{C}_l$ is a polyhedral set, then the relative interior can be dropped from the constraint qualification. Hence, when all subsets $\mathcal{C}_l$ are polyhedral, the constraint qualification simply reduces to $\mathcal{C} \cap \mathrm{int}(\mathrm{dom}\,\phi) \neq \emptyset$, which is already enforced for the definition of Bregman projections.

Finally, if all subsets $\mathcal{C}_l$ are further affine, then we can relax other assumptions. Notably, we do not require ϕ to be cofinite, or equivalently $\mathrm{dom}\,\psi = \mathcal{E}$, but only $\mathrm{dom}\psi$ to be open. The mapping need not be essentially cyclic anymore, as long as it takes each output value an infinite number of times. Moreover, we can completely drop the correction terms from the updates, leading to a simpler technique known as projections onto convex sets (POCS) [50, 51]:

$$\boldsymbol{x}_{k+1} \leftarrow P_{\mathcal{C}_{\sigma(k)}}(\boldsymbol{x}_k). \tag{3.12}$$

3.3 Framework of CROT

In this section, we introduce the framework of CROT [12]. We begin with the formulation of CROT. Subsequently, we set some assumptions on the regularizers for the CROT framework to work. Finally, we introduce algorithms to compute CROT.

3.3.1 Formulation of CROT

Here, we show the formulation of CROT and show that obtaining the optimal solution of CROT corresponds to minimizing the Bregman divergence between two certain matrices.

Suppose we want to define a distance between two empirical distributions $\hat{P}_x$ and $\hat{P}_y$ with CROT. We regularize the disrete OT (2.7) with a function ϕ which is of the Legendre type:

$$L(\boldsymbol{\pi}) \coloneqq \min_{\boldsymbol{\pi}\in\mathcal{G}(\frac{\mathbf{1}_m}{m},\frac{\mathbf{1}_n}{n})} \langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle + \lambda\phi(\boldsymbol{\pi}), \tag{3.13}$$

where $\lambda > 0$ is a regularization parameter. Here, recall that $\boldsymbol{\gamma}$ is a distance matrix and $\boldsymbol{\pi}$ has to satisfy the coupling constraint $\boldsymbol{\pi} \in \mathcal{G}(\frac{\mathbf{1}_m}{m}, \frac{\mathbf{1}_n}{n}) = \{\boldsymbol{T} | \boldsymbol{T} \in \mathbb{R}_+^{m\times n}, \boldsymbol{T}\mathbf{1}_n = \frac{\mathbf{1}_m}{m}, \boldsymbol{T}^\top \mathbf{1}_m = \frac{\mathbf{1}_n}{n}\}$. Since ϕ is of the Legendre type, there exists a dual coordinate system, ϕ and ψ, on $\mathrm{int}(\mathrm{dom}\,\phi)$ and $\mathrm{int}(\mathrm{dom}\,\phi)$ via the homeomorphism $\nabla\phi = \nabla\psi^{-1}$:

$$\boldsymbol{\pi}(\boldsymbol{\theta}) = \nabla\psi(\boldsymbol{\theta}), \tag{3.14}$$

$$\boldsymbol{\theta}(\boldsymbol{\pi}) = \nabla\phi(\boldsymbol{\pi}). \tag{3.15}$$

If we remove the coupling constraint of (3.13), the optimization problem becomes as follows:

$$\min_{\boldsymbol{\pi}\in\mathbb{R}^{m\times n}} \langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle + \lambda\phi(\boldsymbol{\pi}). \tag{3.16}$$

Since $\langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle$ is linear and ϕ is strictly convex with respect to $\boldsymbol{\pi}$, there is a global optimal solution $\boldsymbol{\xi}$ for (3.16):

$$\boldsymbol{\xi} = \nabla\psi(-\boldsymbol{\gamma}/\lambda), \tag{3.17}$$

Table 3.1: Set of assumptions for the considered regularizers ϕ.

(A) Affine constraints	(B) Polyhedral constraints
(A1) ϕ is of Legendre type.	(B1) ϕ is of Legendre type.
(A2) $(0,1)^{m\times n} \subseteq \mathrm{dom}\,\phi$	(B2) $(0,1)^{m\times n} \subseteq \mathrm{dom}\,\phi$
(A3) $\mathrm{dom}\,\phi \subseteq \mathbb{R}_+^{m\times n}$	(B3) $\mathrm{dom}\,\phi \nsubseteq \mathbb{R}_+^{m\times n}$
(A4) dom ψ is open	(B4) $\mathrm{dom}\,\psi = \mathbb{R}^{m\times n}$
(A5) $\mathbb{R}_-^{m\times n} \subset \mathrm{dom}\,\psi$.	

Table 3.2: Domain of each regularizer and its Fenchel conjugate.

Regularization term	dom ϕ	dom ψ
β-potential $(0<\beta<1)$	$\mathbb{R}_+$	$(-\infty, \frac{1}{1-\beta})$
β-potential ($\beta>1$)	$\mathbb{R}_+$	$(\frac{1}{1-\beta}, \infty)$
Boltzman-Shannon entropy	$\mathbb{R}_+$	$\mathbb{R}$
Euclidean norm	$\mathbb{R}$	$\mathbb{R}$

which can be obtained by solving the first-order optimality condition:

$$\boldsymbol{\gamma} + \lambda \nabla\phi(\boldsymbol{\xi}) = \mathbf{0}_{m\times n}. \tag{3.18}$$

We can easily confirm the following equation about $\boldsymbol{\xi}$.

$$\langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle + \lambda\phi(\boldsymbol{\pi}) - \lambda\phi(\boldsymbol{\xi}) - \langle \boldsymbol{\xi}, \boldsymbol{\gamma}\rangle = \lambda B_\phi(\boldsymbol{\pi}||\boldsymbol{\xi}). \tag{3.19}$$

Then, the following equations hold.

$$\begin{aligned}
\boldsymbol{\pi}_\lambda^* &\coloneqq \underset{\boldsymbol{\pi}\in\mathcal{G}(\frac{\mathbf{1}_m}{m}, \frac{\mathbf{1}_n}{n})}{\mathrm{argmin}} \; L(\boldsymbol{\pi}) && (3.20)\\
&= \underset{\boldsymbol{\pi}\in\mathcal{G}(\frac{\mathbf{1}_m}{m}, \frac{\mathbf{1}_n}{n})}{\mathrm{argmin}} \; \langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle + \lambda\phi(\boldsymbol{\pi}) && (3.21)\\
&= \underset{\boldsymbol{\pi}\in\mathcal{G}(\frac{\mathbf{1}_m}{m}, \frac{\mathbf{1}_n}{n})}{\mathrm{argmin}} \; B_\phi(\boldsymbol{\pi}||\boldsymbol{\xi}). && (3.22)
\end{aligned}$$

Therefore, by computing the Bregman projection of $\boldsymbol{\xi}$ onto $\mathcal{G}(\frac{\mathbf{1}_m}{m}, \frac{\mathbf{1}_n}{n})$, we can obtain the transport matrix which satisfies the coupling constraint and minimizes (3.13). Moreover, if λ tends to 0, then $\boldsymbol{\pi}_\lambda^*$ converges to the optimal solution of (2.7) [12].

3.3.2 Theoretical assumptions of the regularizers

Some technical assumptions are required on the convex regularizer ϕ and its Fenchel conjugate $\psi = \phi^*$ for the CROT framework. In the CROT framework, we need to distinguish between situations where the underlying closed convex set can be described as the intersection of either affine subspaces or polyhedral subsets. The two sets of assumptions (A) and (B) are summarized in Table 3.1.

First, we force the regularizer to be of the Legendre type (assumptions (A1) and (B1)). This is required for the definition of Bregman projections (Subsection 3.2.2). In addition, it guarantees the existence of dual coordinate systems on int(domϕ) and int(domψ) via the homeomorphism $\nabla\phi = \nabla\psi^{-1}$:

$$\begin{aligned}
\boldsymbol{\pi}(\boldsymbol{\theta}) &= \nabla\psi(\boldsymbol{\theta}), && (3.23)\\
\boldsymbol{\theta}(\boldsymbol{\pi}) &= \nabla\phi(\boldsymbol{\pi}). && (3.24)
\end{aligned}$$

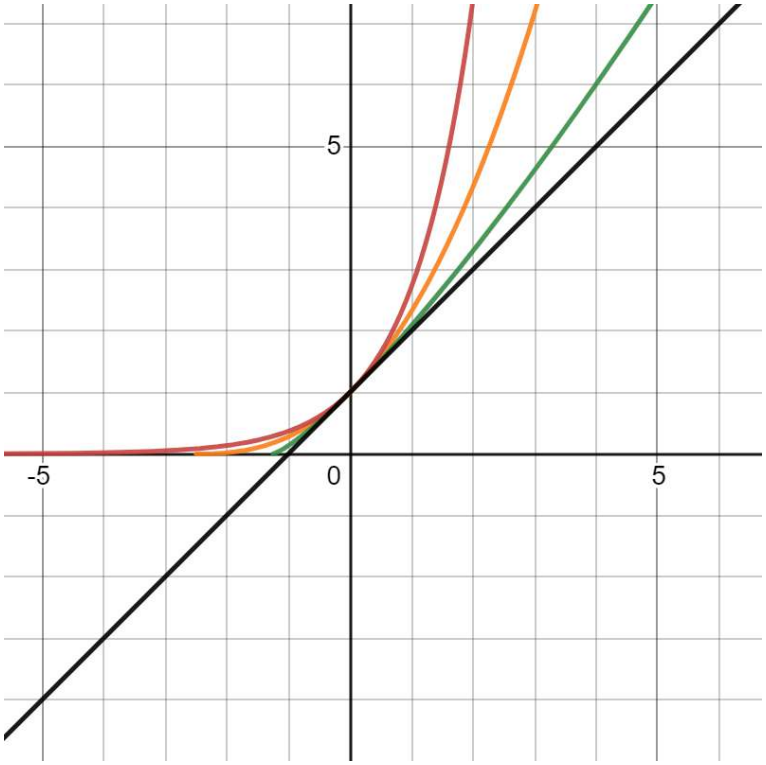


Figure 3.1: $y = \psi'(\theta)$ of the β-potential (yellow: $\beta = 1.4$, green: $\beta = 1.8$), the Boltzmann-Shannon entropy (red) and the Euclidean norm (black).

Table 3.3: The three divergences we are going to compare.

	entropy term $\phi(\pi)$	$B(\pi\|\|\xi)$
β-divergence ($\beta > 0$)	β -potentials $\frac{1}{\beta(\beta-1)}(\pi^\beta - \beta\pi + \beta - 1)$	$\frac{1}{\beta(\beta-1)}(\pi^\beta + (\beta-1)\xi^\beta - \beta\pi\xi^{\beta-1})$
KL divergence	Boltzman-Shannon entropy $\pi \log \pi - \pi + 1$	$\pi \log \frac{\pi}{\xi} - \pi + \xi$
Euclidean distance	Euclidean norm $\frac{1}{2}(\pi - 1)^2$	$\frac{1}{2}(\pi - \xi)^2$

Table 3.4: The derivative of each regularization term, the derivative of its dual function, and its second derivative.

	$\phi'(\pi)$	$\psi'(\pi)$	$\psi''(\theta)$
β-potential $\frac{1}{\beta(\beta-1)}(\pi\beta - \beta\pi + \beta - 1)$	$\frac{1}{\beta-1}(\pi^{\beta-1} - 1)$	$((\beta-1)\theta + 1)^{\frac{1}{\beta-1}}$	$((\beta-1)\theta + 1)^{\frac{1}{\beta-1}-1}$
Boltzman-Shannon entropy $\pi \log \pi - \pi + 1$	$\log \pi$	$\exp \theta$	$\exp \theta$
Euclidean norm $\frac{1}{2}(\pi - 1)^2$	$\pi - 1$	$\theta + 1$	1

With a slight abuse of notation, we omit the reparameterization to simply denote corresponding primal and dual parameters by $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$.

The second assumptions (A2) and (B2) imply that $\mathrm{ri}(\mathcal{G}(\frac{\mathbf{1}_m}{m}, \frac{\mathbf{1}_n}{n})) \subset \mathrm{dom}\,\phi$ and ensure that the constraint qualification $\mathcal{G}(\frac{\mathbf{1}_m}{m}, \frac{\mathbf{1}_m}{n}) \cap \mathrm{int}(\mathrm{dom}\,\phi) \neq \emptyset$ for the Bregman projection onto the transport polytope.

The third assumptions (A3) and (B3) separate between two cases depending on whether $\mathrm{dom}\phi$ lies within the non-negative orthant or not for the alternate Bregman projections (Subsection 3.2.3). In the former case, non-negativity is already ensured by the domain of the regularizer, so the underlying closed convex set is made of two affine subspaces for the row and column sum constraints, and the POCS method can be considered.

The fourth assumption (A4) requires that $\mathrm{dom}\psi$ be open for convergence of this algorithm. On the other hand, in the latter case, there is one additional polyhedral subset for the non-negative constraints and Dykstra's algorithm should be used. The fourth assumption (B4) hence further requires that $\mathrm{dom}\,\psi = \mathbb{R}^{m\times n}$, or equivalently that ϕ be cofinte, for convergence. In both cases, we remark that we necessarily have $\mathrm{dom}\,\psi = \mathrm{dom}\,\nabla\psi$.

The fifth assumption (A5) in the affine constraints ensures that $-\boldsymbol{\gamma}/\lambda$ belongs to $\mathrm{dom}\nabla\psi$ for definition of CROT problems, independently of the non-negative cost matrix $\boldsymbol{\gamma}$ and positive regularization term λ. This is already guaranteed by the fourth assumption in the polyhedral constraints.

In this thesis, we consider four types of regularizers (see Table 3.2). The Boltzmann-Shannon entropy [12] associated to the Kullback-Leibler divergence and the β-potential ($0 < \beta < 1$) [12] associated to the β-divergence are under assumptions (A). When they are chosen as a regularizer, we employ a method called ASA (alternate scaling algorithm) [12] based on the POCS technique, where alternate Bregman projections onto the two affine subspaces for the row and column sum constraints are considered. On the other hand, the Euclidean norm is under assumption (B). When this is chosen as a regularizer, we use the second

method called NASA (non-negative scaling algorithm) [12] based on Dykstra's algorithm, where correction terms and a further Bregman projection onto the polyhedral non-negative orthant are needed. Finally, when the regularizer is the β-potential ($1 < \beta$), since it is neither under assumptions (A) nor assumptions (B), we can no longer use ASA and NASA. Therefore, we propose another algorithm to compute CROT when the regularizer is the β-potential ($1 < \beta$) in Chapter 4. By leveraging the domain of the Fenchel conjugate of the β-potential ($1 < \beta$), our proposed algorithm successfully prevents any probability mass from moving to points farther than a given distance. We show this property enables us to compute OT robustly.

Interestingly, the β-potential tends to be the Boltzmann-Shannon entropy and the Euclidean norm in the limit of $\beta = 1$ and $\beta = 2$, respectively. Therefore, the β-divergence interpolates between the KL-divergence and Euclidean distance. As we will see in the next section, the algorithms are running the Newton-Raphson method [52] on $y = \nabla\psi(\theta)$. In Figure 3.1, we can see an interpolation between the KL-divergence and Euclidean distance in $y = \psi'(\theta)$ as we move β between $0 < \beta < 2$.

The corresponding divergences for each regularization term are shown in Table 3.3. The derivative of each regularization term, the derivative of its dual function, and its second derivative are shown in Table 3.4.

3.3.3 Algorithms

In this subsection, we introduce algorithms to obtain $\boldsymbol{\pi}^*_\lambda$ in (3.20). To simplify the notations, we omit the regularization parameter λ in the subscript and simply write $\boldsymbol{\pi}^*$ instead of $\boldsymbol{\pi}^*_\lambda$.

We first study the underlying Bregman projections in their generic form. The closed convex transport polytope $\mathcal{G}(\frac{\mathbf{1}_m}{m}, \frac{\mathbf{1}_n}{n})$ is the intersection of the non-negative orthant:

$$\mathcal{C}_0 = \mathbb{R}_+^{m\times n}, \tag{3.25}$$

which is a polyhedral subset, with two affine subspaces:

$$\mathcal{C}_1 = \{\boldsymbol{\pi} \in \mathbb{R}^{m\times n} | \boldsymbol{\pi}\mathbf{1}_n = \frac{\mathbf{1}_m}{m}\}, \tag{3.26}$$

$$\mathcal{C}_2 = \{\boldsymbol{\pi} \in \mathbb{R}^{m\times n} | \boldsymbol{\pi}^\top\mathbf{1}_m = \frac{\mathbf{1}_n}{n}\}. \tag{3.27}$$

We first consider Bregman projections of a given matrix $\overline{\boldsymbol{\pi}} \in \operatorname{int}(\operatorname{dom}\phi)$ onto $\mathcal{C}_1$ and $\mathcal{C}_2$. For the projection onto $\mathcal{C}_1$ and $\mathcal{C}_2$, we can employ the method of Lagrange multipliers. The Lagrangians with Lagrange multipliers $\boldsymbol{\mu} \in \mathbb{R}^m$ and $\boldsymbol{\nu} \in \mathbb{R}^n$ for the Bregman projections $\boldsymbol{\pi}^*_1$ and $\boldsymbol{\pi}^*_2$ of a given matrix $\overline{\boldsymbol{\pi}} \in \operatorname{int}(\operatorname{dom}\phi)$ onto $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively write as follows:

$$\mathcal{L}_1(\boldsymbol{\pi}, \boldsymbol{\mu}) = \phi(\boldsymbol{\pi}) - \langle \boldsymbol{\pi}, \nabla\phi(\overline{\boldsymbol{\pi}})\rangle + \boldsymbol{\mu}^\top(\boldsymbol{\pi}\mathbf{1}_n - \frac{\mathbf{1}_m}{m}), \tag{3.28}$$

$$\mathcal{L}_2(\boldsymbol{\pi}, \boldsymbol{\nu}) = \phi(\boldsymbol{\pi}) - \langle \boldsymbol{\pi}, \nabla\phi(\overline{\boldsymbol{\pi}})\rangle + \boldsymbol{\nu}^\top(\boldsymbol{\pi}^\top\mathbf{1}_m - \frac{\mathbf{1}_n}{n}). \tag{3.29}$$

Their gradients are given on $\operatorname{int}(\operatorname{dom}\phi)$ by

$$\nabla\mathcal{L}_1(\boldsymbol{\pi}, \boldsymbol{\mu}) = \nabla\phi(\boldsymbol{\pi}) - \nabla\phi(\overline{\boldsymbol{\pi}}) + \boldsymbol{\mu}\mathbf{1}_n^\top, \tag{3.30}$$

$$\nabla\mathcal{L}_2(\boldsymbol{\pi}, \boldsymbol{\nu}) = \nabla\phi(\boldsymbol{\pi}) - \nabla\phi(\overline{\boldsymbol{\pi}}) + \mathbf{1}_m\boldsymbol{\nu}^\top, \tag{3.31}$$

and vanish at $\boldsymbol{\pi}^*_1, \boldsymbol{\pi}^*_2 \in \operatorname{int}(\operatorname{dom}\phi)$ if and only if

$$\boldsymbol{\pi}^*_1 = \nabla\psi(\nabla\phi(\overline{\boldsymbol{\pi}}) - \boldsymbol{\mu}\mathbf{1}_n^\top), \tag{3.32}$$

$$\boldsymbol{\pi}^*_2 = \nabla\psi(\nabla\phi(\overline{\boldsymbol{\pi}}) - \mathbf{1}_m\boldsymbol{\nu}^\top). \tag{3.33}$$

Algorithm 1 Alternate scaling algorithm (ASA)

$\boldsymbol{\theta}^* \leftarrow -\boldsymbol{\gamma}/\lambda$
repeat
  $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^* - \boldsymbol{\mu}\mathbf{1}_n{}^\top$, where $\boldsymbol{\mu}$ uniquely solves $\nabla\psi(\boldsymbol{\theta}^* - \boldsymbol{\mu}\mathbf{1}_n{}^\top)\mathbf{1}_n = \frac{\mathbf{1}_m}{m}$
  $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^* - \mathbf{1}_m\boldsymbol{\nu}^\top$, where $\boldsymbol{\nu}$ uniquely solves $\nabla\psi(\boldsymbol{\theta}^* - \mathbf{1}_m\boldsymbol{\nu}^\top)^\top\mathbf{1}_m = \frac{\mathbf{1}_n}{n}$
until convergence
$\boldsymbol{\pi}^* \leftarrow \nabla\boldsymbol{\psi}(\boldsymbol{\theta}^*)$

By duality, the Bregman projections onto $\mathcal{C}_1$, $\mathcal{C}_2$ are thus equivalent to finding the unique vectors $\boldsymbol{\mu}$, $\boldsymbol{\nu}$, such that the rows of $\boldsymbol{\pi}_1^*$ sum up to $\frac{\mathbf{1}_m}{m}$, respectively the columns of $\boldsymbol{\pi}_2^*$ sum up to $\frac{\mathbf{1}_m}{n}$:

$$\nabla\psi(\nabla\phi(\overline{\boldsymbol{\pi}}) - \boldsymbol{\mu}\mathbf{1}_n^\top)\mathbf{1}_n = \frac{\mathbf{1}_m}{m}, \tag{3.34}$$

$$\nabla\psi(\nabla\phi(\overline{\boldsymbol{\pi}}) - \mathbf{1}_m\boldsymbol{\nu}^\top)^\top\mathbf{1}_m = \frac{\mathbf{1}_n}{n}. \tag{3.35}$$

Alternate scaling algorithm (ASA) [12]

Under assumptions (A) in Table 3.1, since $\operatorname{dom}\phi \in \mathbb{R}_+^{m\times n}$ is guaranteed, we can obtain $P_{\mathcal{C}}(\boldsymbol{\xi})$ by simply repeating projections on $\mathcal{C}_1 \to \mathcal{C}_2 \to \mathcal{C}_1 \to \mathcal{C}_2 \to \cdots$ starting from $\boldsymbol{\xi} = \nabla\psi(-\boldsymbol{\gamma}/\lambda)$.

Starting from $\boldsymbol{\xi}$ and writing the successive vectors $\boldsymbol{\mu}^{(k)}$, $\boldsymbol{\nu}^{(k)}$ along iterations, we have the following sequence:

$$\begin{aligned}
\nabla\psi(\boldsymbol{\gamma}/\lambda) &\to \nabla\psi(\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}_n^\top)\\
&\to \nabla\psi(\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\nu}^{(1)\top})\\
&\to \cdots\\
&\to \nabla\psi(\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\nu}^{(1)\top} - \cdots - \boldsymbol{\mu}^{(k)}\mathbf{1}_n^\top)\\
&\to \nabla\psi(\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\nu}^{(1)\top} - \cdots - \boldsymbol{\mu}^{(k)}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\nu}^{(\boldsymbol{k})\top})\\
&\to \cdots\\
&\to \boldsymbol{\pi}^*.
\end{aligned}$$

In other words, we obtain $\boldsymbol{\pi}^*$ by scaling iteratively the rows and columns of the successive estimates through $\nabla\psi$. An efficient algorithm, called ASA, is to store a unique $m \times n$ matrix in dual parameter space and update it by alternating the projections in primal parameter space (Algorithm 1).

Alternate scaling algorithm (ASA) in the separable case

Since we are restricting ourselves to a separable Bregman divergence, we can compute the projection step more efficiently. Due to the separability, the projections onto $\mathcal{C}_1$ and $\mathcal{C}_2$ can be divided into m and n parallel subproblems in the search space of 1-dimension as follows:

$$\sum_{j=1}^{n}\psi'(\overline{\theta}_{ij} - \mu_i) = \frac{1}{m}, \tag{3.36}$$

$$\sum_{i=1}^{m}\psi'(\overline{\theta}_{ij} - \nu_j) = \frac{1}{n}. \tag{3.37}$$

Here, we denote the dual coordinate of $\overline{\boldsymbol{\pi}}$ by $\overline{\boldsymbol{\theta}}$.

In order to obtain the Lagrange multipliers μ_i and ν_j, we use the Newton Raphson method. More specifically, we eploit the following functions:

$$f(\mu_i) = -\sum_{j=1}^{n} \psi'(\overline{\theta}_{ij} - \mu_i), \tag{3.38}$$

$$g(\nu_j) = -\sum_{i=1}^{m} \psi'(\overline{\theta}_{ij} - \nu_j). \tag{3.39}$$

These functions are defined on the open intervals $(\hat{\theta}_i - \theta_{\text{limit}}, +\infty)$ and $(\check{\theta}_j - \theta_{\text{limit}}, +\infty)$, where $0 < \theta_{\text{limit}} < +\infty$ is such that $\operatorname{dom}\psi = (-\infty, \theta_{\text{limit}})$, and $\hat{\theta}_i = \max\{\overline{\theta}_{ij}\}_{1\leq j\leq n}$, $\check{\theta}_j = \max\{\overline{\theta}_{ij}\}_{1\leq i\leq m}$. We can now obtain the unique solution to $f(\mu_i) = -\frac{1}{m}$ and $g(\nu_j) = -\frac{1}{n}$. Starting with $\mu_i = 0$ and $\nu_j = 0$, the Newton Raphson updates:

$$\mu_i \leftarrow \mu_i + \frac{\sum_{j=1}^{n} \psi'(\overline{\theta}_{ij} - \mu_i) - \frac{1}{m}}{\sum_{j=1}^{n} \psi''(\overline{\theta}_{ij} - \mu_i)}, \tag{3.40}$$

$$\nu_j \leftarrow \nu_j + \frac{\sum_{i=1}^{m} \psi'(\overline{\theta}_{ij} - \nu_i) - \frac{1}{n}}{\sum_{i=1}^{m} \psi''(\overline{\theta}_{ij} - \nu_i)}, \tag{3.41}$$

converge to the optimal solution with a quadraitic rate. To avoid storing the intermediate Lagrange multipliers, the updates can be directly written in terms of the dual parameters:

$$\theta^*_{1,ij} \leftarrow \theta^*_{1,ij} - \frac{\sum_{j=1}^{n} \psi'(\theta^*_{1,ij}) - \frac{1}{m}}{\sum_{j=1}^{n} \psi''(\theta^*_{1,ij})}, \tag{3.42}$$

$$\theta^*_{2,ij} \leftarrow \theta^*_{2,ij} - \frac{\sum_{i=1}^{m} \psi'(\theta^*_{2,ij}) - \frac{1}{n}}{\sum_{i=1}^{m} \psi''(\theta^*_{2,ij})}, \tag{3.43}$$

after initilalization by $\theta^*_{1,ij} \leftarrow \overline{\theta}_{ij}$, $\theta^*_{2,ij} \leftarrow \overline{\theta}_{ij}$. Here, $\theta^*_{1,ij}$ and $\theta^*_{2,ij}$ are the ith row and jth column of $\boldsymbol{\theta}^*_1$ and $\boldsymbol{\theta}^*_2$ respectively. $\boldsymbol{\theta}^*_1$ and $\boldsymbol{\theta}^*_2$ are the dual coordinates of $\boldsymbol{\pi}^*_1$ and $\boldsymbol{\pi}^*_2$ respectively.

Therefore, the projections can be obtained by iterating the respective Newton-Raphson update steps, which can be written compactly with matrix and vector operations shown in Algorithm 2.

Algorithm 2 can be applied to CROT when the regularizer is the Boltzmann-Shannon entropy or the β-potential ($0 < \beta < 1$). When the regularizer is the Boltzmann-Shannon entropy, the updates in the POCS techniques can be written analytically, leading to the Sinkhorn algorithm [14]. Specifically, the two projections amount to normalizing, in turn, the rows and columns of $\boldsymbol{\pi}^*$ so that they sum up to $\frac{\mathbf{1}_m}{m}$ and $\frac{\mathbf{1}_m}{n}$, respectively:

$$\boldsymbol{\pi}^* \leftarrow \operatorname{diag}\Big(\frac{\frac{\mathbf{1}_m}{m}}{\boldsymbol{\pi}^* \mathbf{1}_n}\Big)\boldsymbol{\pi}^*, \tag{3.44}$$

$$\boldsymbol{\pi}^* \leftarrow \boldsymbol{\pi}^* \operatorname{diag}\Big(\frac{\frac{\mathbf{1}_n}{n}}{\boldsymbol{\pi}^{*\top} \mathbf{1}_m}\Big). \tag{3.45}$$

Here, $\operatorname{diag}(\boldsymbol{v})$ is an operator which transforms a vector $\boldsymbol{v} \in \mathbb{R}^d$ into a diagonal matrix $\boldsymbol{\pi} \in \mathbb{R}^{m\times n}$ such that $\pi_{ii} = v_i$, for all $1 \leq i \leq d$. This can be optimized by remarking that $\boldsymbol{\pi}^{*(k)}$ after each couple of projections verifies

$$\boldsymbol{\pi}^{*(k)} = \operatorname{diag}(\boldsymbol{u}^{(k)})\boldsymbol{\xi}\operatorname{diag}(\boldsymbol{v}^{(k)}), \tag{3.46}$$

Algorithm 2 Alternate scaling algorithm in the separable case

$\boldsymbol{\theta}^* \leftarrow -\boldsymbol{\gamma}/\lambda$
repeat
  repeat
    $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^* - \frac{\nabla\psi(\boldsymbol{\theta}^*)\mathbf{1}_m - \frac{\mathbf{1}_m}{m}}{\nabla^2\psi(\boldsymbol{\theta}^*)\mathbf{1}_m}\mathbf{1}_n^\top$
  until convergence
  repeat
    $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^* - \mathbf{1}_m \frac{\mathbf{1}_m^\top \nabla\psi(\boldsymbol{\theta}^*) - (\frac{\mathbf{1}_n}{n})^\top}{\mathbf{1}_n^\top \nabla^2\psi(\boldsymbol{\theta}^*)}$
  until convergence
until convergence
$\boldsymbol{\pi}^* \leftarrow \nabla\boldsymbol{\psi}(\boldsymbol{\theta}^*)$

where $\boldsymbol{\Lambda} = \exp(-\boldsymbol{\gamma}/\lambda)$ and vectors $\boldsymbol{u}^{(k)}$, $\boldsymbol{v}^{(k)}$ satisfy the following recursion:

$$\boldsymbol{u}^{(k)} = \frac{\frac{\mathbf{1}_m}{m}}{\boldsymbol{\Lambda}\boldsymbol{v}^{(k-1)}}, \tag{3.47}$$

$$\boldsymbol{v}^{(k)} = \frac{\frac{\mathbf{1}_m}{n}}{\boldsymbol{\Lambda}^\top\boldsymbol{u}^{(k)}}, \tag{3.48}$$

with $\boldsymbol{v}^0 = 1$. This allows a fast implementation by performing only matrix-vector multiplications using a fixed matrix $\boldsymbol{\Lambda} = \exp(-\boldsymbol{\gamma}/\lambda)$. We can further save one element-wise vector multiplication per update:

$$\boldsymbol{u} \leftarrow \frac{\mathbf{1}_m}{\mathrm{diag}(\boldsymbol{m})\boldsymbol{\Lambda}\boldsymbol{v}}, \tag{3.49}$$

$$\boldsymbol{v} \leftarrow \frac{\mathbf{1}_m}{\mathrm{diag}(\boldsymbol{n})\boldsymbol{\Lambda}^\top\boldsymbol{u}}, \tag{3.50}$$

where the matrices $\mathrm{diag}(\boldsymbol{m})\boldsymbol{\Lambda}$ and $\mathrm{diag}(\boldsymbol{n})\boldsymbol{\Lambda}^\top$ are computed and stored. Here, $\mathrm{diag}(\boldsymbol{m})$ and $\mathrm{diag}(\boldsymbol{n})$ denote a diagonal matrix whose diagonal elements are all m and n, respectively.

Non-negative alternate scaling algorithm (NASA) [12]

Under assumptions (B) in Table 3.1, we now have to consider a non-negative constraint since $\mathrm{dom}\phi \nsubseteq \mathbb{R}_+^{m\times n}$. We project $\boldsymbol{\xi} = \nabla\psi(\boldsymbol{\gamma}/\boldsymbol{\lambda})$ on $\mathcal{C}_0 \to \mathcal{C}_1 \to \mathcal{C}_0 \to \mathcal{C}_2 \to \mathcal{C}_0 \to \mathcal{C}_1 \to \mathcal{C}_0 \to \cdots$ with the non-negativity of each update guaranteed.

Let us consider the projection of given matrix $\overline{\boldsymbol{\pi}}$ onto $\mathcal{C}_0$. We denote this projection $P_{\mathcal{C}_0}(\overline{\boldsymbol{\pi}})$ by $\boldsymbol{\pi}_0^*$. Then, the Karush-Kuhn-Tucker conditions [53, 54] for $\boldsymbol{\pi}_0^*$ are as follows:

$$\boldsymbol{\pi}_0^* \geq \mathbf{0}_{m\times n}, \tag{3.51}$$

$$\nabla\phi(\boldsymbol{\pi}_0^*) - \nabla\phi(\overline{\boldsymbol{\pi}}) \geq \mathbf{0}_{m\times n}, \tag{3.52}$$

$$(\nabla\phi(\boldsymbol{\pi}_0^*) - \nabla\phi(\overline{\boldsymbol{\pi}})) \odot \boldsymbol{\pi}_0^* = \mathbf{0}_{m\times n}, \tag{3.53}$$

where (3.51) is the primal feasibility, (3.52) is the dual feasibility, and (3.53) is the complementary slackness.

Since the non-negative orthant is polyhedral, but not affine, we also need to incorporate correction terms $\boldsymbol{\vartheta}$, $\boldsymbol{\varrho}$, $\boldsymbol{\varsigma}$ for all three projections. In more detail, the projections are computed after correction so that we do not directly project the obtained updates $\boldsymbol{\theta}^*$ but the corrected

Algorithm 3 Non-negative alternate scaling algorithm

$\boldsymbol{\theta}^* \leftarrow -\boldsymbol{\gamma}/\lambda$
$\boldsymbol{\vartheta} \leftarrow \mathbf{0}_{m\times n}$
$\boldsymbol{\varrho} \leftarrow \mathbf{0}_{m\times n}$
$\boldsymbol{\varsigma} \leftarrow \mathbf{0}_{m\times n}$
$\overline{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^* + \boldsymbol{\vartheta}$
$\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$, where $\boldsymbol{\theta}$ uniquely solves $\nabla\psi(\boldsymbol{\theta}) \geq \mathbf{0}_{m\times n},\ \theta \geq \overline{\theta},\ (\boldsymbol{\theta} - \overline{\boldsymbol{\theta}}) \odot \nabla\psi(\boldsymbol{\theta}) = \mathbf{0}_{m\times n}$
$\boldsymbol{\vartheta} \leftarrow \overline{\boldsymbol{\theta}} - \boldsymbol{\theta}^*$
repeat
  $\overline{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^* + \boldsymbol{\varrho}$
  $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta} - \boldsymbol{\mu}\mathbf{1}_n^\top,$ where $\boldsymbol{\mu}$ uniquely solves$\nabla\psi(\overline{\boldsymbol{\theta}} - \boldsymbol{\mu}\mathbf{1}_n^\top)\mathbf{1}_m = \frac{\mathbf{1}_m}{m}$
  $\boldsymbol{\varrho} \leftarrow \overline{\boldsymbol{\theta}} - \boldsymbol{\theta}^*$
  $\overline{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^* + \boldsymbol{\vartheta}$
  $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$, where $\boldsymbol{\theta}$ uniquely solves $\nabla\psi(\boldsymbol{\theta}) \geq \mathbf{0}_{m\times n},\ \theta \geq \overline{\theta},\ (\boldsymbol{\theta} - \overline{\boldsymbol{\theta}}) \odot \nabla\psi(\boldsymbol{\theta}) = \mathbf{0}_{m\times n}$
  $\boldsymbol{\vartheta} \leftarrow \overline{\boldsymbol{\theta}} - \boldsymbol{\theta}^*$
  $\overline{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^* + \boldsymbol{\varsigma}$
  $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta} - \mathbf{1}_m\boldsymbol{\nu}^\top,$ where $\boldsymbol{\mu}$ uniquely solves $\nabla\psi(\overline{\boldsymbol{\theta}} - \mathbf{1}_m\boldsymbol{\nu}^\top)^\top\mathbf{1}_m = \frac{\mathbf{1}_m}{n}$
  $\boldsymbol{\varsigma} \leftarrow \overline{\boldsymbol{\theta}} - \boldsymbol{\theta}^*$
  $\overline{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^* - \vartheta$
  $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}$, where $\boldsymbol{\theta}$ uniquely solves $\nabla\psi(\boldsymbol{\theta}) \geq \mathbf{0}_{m\times n},\ \theta \geq \overline{\theta},\ (\boldsymbol{\theta} - \overline{\boldsymbol{\theta}}) \odot \nabla\psi(\boldsymbol{\theta}) = \mathbf{0}_{m\times n}$
  $\boldsymbol{\vartheta} \leftarrow \overline{\boldsymbol{\theta}} - \boldsymbol{\theta}^*$
until convergence
$\boldsymbol{\pi}^* \leftarrow \nabla\psi(\boldsymbol{\theta}^*)$

updates $\overline{\boldsymbol{\theta}} = \boldsymbol{\theta}^* + \boldsymbol{\vartheta}$, $\overline{\boldsymbol{\theta}} = \boldsymbol{\theta}^* + \boldsymbol{\varrho}$, $\overline{\boldsymbol{\theta}} = \boldsymbol{\theta}^* + \boldsymbol{\varsigma}$ for the respective subsets. The correction terms are also updated as the difference $\overline{\boldsymbol{\theta}} - \boldsymbol{\theta}^*$ between the projected point and its projection. Dykstra's algorithm (3.10) for Bregman divergences with corrections (3.11) then guarantees that the projection of $\boldsymbol{\xi}$ onto $\mathcal{G}(\frac{\mathbf{1}_m}{m}, \frac{\mathbf{1}_n}{n})$ is obtained with linear convergence. The algorithm is shown in Algorithm 3.

Non-negative alternate scaling algorithm (NASA) in the separable case

In case of separability, the Karush-Kuhn-Tucker conditions for projection onto $\mathcal{C}_0$ simplify to provide a closed-form solution on primal parameters:

$$\pi^*_{0,ij} = \max\{0, \overline{\pi}_{ij}\}, \tag{3.54}$$

where, $\pi^*_{0,ij}$ is the element in row i and column j of matrix $\boldsymbol{\pi}^*_0$. Since ϕ' is increasing, this is equivalent on dual parameters to

$$\theta^*_{0,ij} = \max\{\phi'(0), \overline{\theta}_{ij}\}. \tag{3.55}$$

Therefore, in the separable case, the non-negativity constraint can be obtained analytically (3.54) and the sequence of updates greatly simplifies. Starting from $\boldsymbol{\xi}$ and writing the

Algorithm 4 Non-negative alternate scaling algorithm in the separable case

$\tilde{\boldsymbol{\theta}} \leftarrow -\boldsymbol{\gamma}/\lambda$
$\boldsymbol{\theta}^* \leftarrow \max\{\nabla\phi(\mathbf{0}_{m\times n}), \tilde{\boldsymbol{\theta}}\}$
repeat
  $\boldsymbol{\tau} \leftarrow \mathbf{0}_{m\times n}$
  repeat
    $\boldsymbol{\tau} \leftarrow \boldsymbol{\tau} + \frac{\nabla\psi(\boldsymbol{\theta}^* - \boldsymbol{\tau}\mathbf{1}_n^\top)\mathbf{1}_n - \frac{\mathbf{1}_m}{m}}{\nabla^2\psi(\boldsymbol{\theta}^* - \boldsymbol{\tau}\mathbf{1}_n^\top)\mathbf{1}_n}$
  until convergence
  $\tilde{\boldsymbol{\theta}} \leftarrow \tilde{\boldsymbol{\theta}} - \boldsymbol{\tau}\mathbf{1}_n^\top$
  $\boldsymbol{\theta}^* \leftarrow \max\{\nabla\phi(\mathbf{0}_{m\times n}), \tilde{\boldsymbol{\theta}}\}$
  $\boldsymbol{\sigma} \leftarrow \mathbf{0}_{m\times n}$
  repeat
    $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} + \frac{\mathbf{1}_n^\top \nabla\psi(\boldsymbol{\theta}^* - \mathbf{1}_m\boldsymbol{\sigma}^\top) - (\frac{\mathbf{1}_n}{n})^\top}{\mathbf{1}_m^\top \nabla^2\psi(\boldsymbol{\theta}^* - \mathbf{1}_m\boldsymbol{\sigma}^\top)}$
  until convergence
  $\tilde{\boldsymbol{\theta}} \leftarrow \tilde{\boldsymbol{\theta}} - \mathbf{1}_m\boldsymbol{\sigma}^\top$
  $\boldsymbol{\theta}^* \leftarrow \max\{\nabla\phi(\mathbf{0}_{m\times n}), \tilde{\boldsymbol{\theta}}\}$
until convergence
$\boldsymbol{\pi}^* \leftarrow \boldsymbol{\nabla\psi}(\boldsymbol{\theta}^*)$

successive vectors $\boldsymbol{\mu}^{(k)}$, $\boldsymbol{\nu}^{(k)}$ along iterations, we have:

$$
\begin{aligned}
\nabla\psi(-\boldsymbol{\gamma}/\lambda) &\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}),\ -\boldsymbol{\gamma}/\lambda\}\Big)\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}),\ -\boldsymbol{\gamma}/\lambda\} - \boldsymbol{\mu}^{(1)}\mathbf{1}_n^\top\Big)\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}),\ -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}_n^\top\ )\}\Big)\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}_n^\top)\} - \mathbf{1}_m\boldsymbol{\nu}^{(1)\top}\Big)\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\nu}^{(1)\top}\}\Big)\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}_n^\top) - \mathbf{1}_m\boldsymbol{\nu}^{(1)\top}\} + \boldsymbol{\mu}^{(1)}\mathbf{1}_n^\top - \boldsymbol{\mu}^{(2)}\mathbf{1}_n^\top\Big)\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(2)}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\nu}^{(1)\top}\}\Big)\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(2)}\mathbf{1}_n^\top) - \mathbf{1}_m\boldsymbol{\nu}^{(1)\top}\} + \mathbf{1}_m\boldsymbol{\nu}^{(1)\top} - \mathbf{1}_m\boldsymbol{\nu}^{(2)\top}\Big)\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(2)}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\nu}^{(2)\top}\}\Big)\\
&\rightarrow \cdots\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(k)}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\nu}^{(k)\top}\}\Big)\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(k)}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\nu}^{(k)\top}\} + \boldsymbol{\mu}^{(k)}\mathbf{1}_n^\top - \boldsymbol{\mu}^{(k+1)}\mathbf{1}_n^\top\Big)\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(k+1)}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\nu}^{(k)\top}\}\Big)\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(k+1)}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\nu}^{(k)\top}\} + \mathbf{1}_m\boldsymbol{\nu}^{(k)\top} - \mathbf{1}_m\boldsymbol{\nu}^{(k+1)\top}\Big)\\
&\rightarrow \nabla\psi\Big(\max\{\nabla\phi(\mathbf{0}_{m\times n}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(k+1)}\mathbf{1}_n^\top - \mathbf{1}_m\boldsymbol{\nu}^{(k+1)\top}\}\Big)\\
&\rightarrow \cdots\\
&\rightarrow \boldsymbol{\pi}^*.
\end{aligned}
$$

An efficent algorithm then exploits the differences $\boldsymbol{\tau}^{(k)} = \boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}^{(k-1)}$ and $\boldsymbol{\sigma}^{(k)} = \boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)}$ to scale the rows and columns (Algorithm 4). Algorithm 4 can be applied to CROT when the regularizer is the Euclidean norm.

Chapter 4

Outlier robust CROT

In this chapter, we propose an algorithm to compute CROT robustly by using β-potential ($1 < \beta$) as the regularizer.

4.1 Definition of outliers

In general, an outlier is defined as an observation that lies an abnormal distance from other samples in data [55]. In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal. Many definitions of an outlier have been proposed [56, 57, 58].

In this thesis, the definition outliers is defined as follows. Suppose we have two datasets $\{\boldsymbol{x}_i\}_{i=1}^m$ and $\{\boldsymbol{y}_j\}_{j=1}^n$. We assume $\{\boldsymbol{x}_i\}_{i=1}^m$ are samples from a single distribution P_x. Let $\boldsymbol{\gamma}$ ($\gamma_{ij} = h(\boldsymbol{x}_i - \boldsymbol{y}_j)$) be the distance matrix.

Definition 7. *For $z > 0$, the indices of outliers J are defined as follows:*

$$\forall j \in J,\ \forall i \in \{1, \dots, m\}, \gamma_{ij} \geq z. \tag{4.1}$$

This means that any point in $\{\boldsymbol{y}_j\}_{j=1}^n$ that is more than z away from any point in $\{\boldsymbol{x}_i\}_{i=1}^m$ is considered as outliers.

Next, we define *transporting no mass* as follows.

Definition 8. *Suppose $\boldsymbol{\pi} \in \mathbb{R}_+^{m \times n}$ and a set of indices $O \subseteq \{1, \dots, n\}$ satisfy the following condition:*

$$\forall i, \pi_{ij} = 0 \quad \text{if} \quad j \in O. \tag{4.2}$$

Then, we say $\boldsymbol{\pi}$ transports no mass to O.

This means that any point in $\{\boldsymbol{y}_j\}_{j=1}^n$ that is more than z away from any point in $\{\boldsymbol{x}_i\}_{i=1}^m$ is considered as outliers.

4.2 β-potential regularization

We use the β-potential $\phi(\pi) = \frac{1}{\beta(\beta-1)}(\pi^\beta - \beta\pi + \beta - 1)$, associated with the β-divergence (Table 3.3) to robustify the CROT. The domains of primal ϕ and its Fenchel conjugate ψ are shown in Table 3.2.

Algorithm 5 Non-negative alternate scaling algorithm for β-divergence when $\beta > 1$

1: $\tilde{\boldsymbol{\theta}} \leftarrow -\boldsymbol{\gamma}/\lambda$
2: $\boldsymbol{\theta}^* \leftarrow \max\{\nabla\phi(\mathbf{0}_{m\times n}), \tilde{\boldsymbol{\theta}}\}$
3: **repeat**
4: $\quad \boldsymbol{\tau} = \frac{\nabla\psi(\boldsymbol{\theta}^*)\mathbf{1}_n - \frac{\mathbf{1}_m}{m}}{\nabla^2\psi(\boldsymbol{\theta}^*)\mathbf{1}_n}$
5: $\quad \boldsymbol{\tau} \leftarrow \max(\boldsymbol{\tau},\ \hat{\boldsymbol{\theta}}^* - \nabla\phi(\frac{\mathbf{1}_m}{m}))$
6: $\quad \tilde{\boldsymbol{\theta}} \leftarrow \tilde{\boldsymbol{\theta}} - \boldsymbol{\tau}\mathbf{1}_n{}^\top$
7: $\quad \boldsymbol{\theta}^* \leftarrow \max\{\nabla\phi(\mathbf{0}), \tilde{\boldsymbol{\theta}}\}$
8: $\quad \boldsymbol{\sigma} = \frac{\mathbf{1}_m{}^\top\nabla^2\psi(\boldsymbol{\theta}^*) - (\frac{\mathbf{1}_n}{n})^\top}{\mathbf{1}_m{}^\top\nabla^2\psi(\boldsymbol{\theta}^*)}$
9: $\quad \boldsymbol{\sigma} \leftarrow \max(\boldsymbol{\sigma}, \hat{\boldsymbol{\theta}}^* - \nabla\phi(\frac{\mathbf{1}_n}{n}))$
10: $\quad \tilde{\boldsymbol{\theta}} \leftarrow \tilde{\boldsymbol{\theta}} - \mathbf{1}_m\boldsymbol{\sigma}^\top$
11: $\quad \boldsymbol{\theta}^* \leftarrow \max\{\nabla\phi(\mathbf{0}_{m\times n}), \tilde{\boldsymbol{\theta}}\}$
12: **until** convergence
13: $\boldsymbol{\pi}^* \leftarrow \boldsymbol{\nabla\psi(\theta^*)}$

Our proposed algorithm is shown in Algorithm 5. The dual coordinate of the unconstrained CROT solution $\boldsymbol{\xi}$ is denoted by $\tilde{\boldsymbol{\theta}}$. We execute the projections in the cyclic order of $\mathcal{C}_0 \to \mathcal{C}_1 \to \mathcal{C}_0 \to \mathcal{C}_2 \to \mathcal{C}_0 \to \mathcal{C}_1 \to \mathcal{C}_0 \to \mathcal{C}_2 \to \cdots$.

Lines 2, 7, and 11 in Algorithm 5 enforce the dual constraint $\theta^*_{ij} \geq \frac{1}{1-\beta}$ corresponding to $\operatorname{dom}\psi = (\frac{1}{1-\beta}, \infty)$ (Table 3.3). Lines 4–6 correspond to a projection onto $\mathcal{C}_1$ implemented on the dual coordinate. Since the dual variable must satisfy $\theta^*_{ij} \geq \frac{1}{1-\beta}$ due to $\operatorname{dom}\psi = (\frac{1}{1-\beta}, \infty)$, we update the dual variable only once in the Newton-Raphson method (line 4) to meet this constraint. Similarily, the projection onto $\mathcal{C}_2$ is shown in lines 8–10.

The procedure in line 5 is based on Section 4.6 in [12] which accelerates the convergence of Algorithm 5 experimentally. Let $\hat{\boldsymbol{\theta}}^*$ be the m-dimensional vector whose ith element is the largest value in the ith row of $\boldsymbol{\theta}^*$ defined as follows:

$$\hat{\theta}^*_i \quad := \quad \max\{\theta^*_{ij}\}_{1\leq j\leq n}. \tag{4.3}$$

For any i, we force $\pi^*_{1,ij}$ to satisfy the following conditions:

$$\forall j,\ 0 \leq \pi^*_{1,ij} \leq \tfrac{1}{m}. \tag{4.4}$$

Since ϕ is convex,

$$\begin{aligned} & 0 \leq \pi^*_{1,ij} \leq \tfrac{1}{m} \\ \iff \quad & \phi'(0) \leq \phi'(\pi^*_{1,ij}) = \theta^*_{1,ij} \leq \phi'(\tfrac{1}{m}) \end{aligned} \tag{4.5}$$

holds. Hence, for every i, if we lower-bound τ_i as

$$\tau_i \quad = \quad \max\{\tau_i, \hat{\theta}^*_i - \phi'\left(\tfrac{1}{m}\right)\}, \tag{4.6}$$

then, for any j,

$$\begin{aligned} \tilde{\theta}_{ij} - \tau_i \quad &\leq \quad \max\{\phi'(0), \tilde{\theta}_{ij}\} && (4.7)\\ &= \quad \theta^*_{ij} - \tau_i && (4.8)\\ &\leq \quad \hat{\theta}^*_i - \tau_i && (4.9)\\ &\leq \quad \phi'\left(\tfrac{1}{m}\right). && (4.10) \end{aligned}$$

This means that every element in the ith row of $\tilde{\boldsymbol{\theta}}$ computed in line 6 in Algorithm 5 is no larger than $\phi'(\frac{1}{m})$. After line 7, $\boldsymbol{\theta}^*$ satisfies the condition (4.4). Similarly, we force $\pi_{2,ij}$ to satisfy the following conditions:

$$\forall i,\ 0 \leq \pi_{2,ij} \leq \tfrac{1}{n}. \tag{4.11}$$

After line 11, this condition is satisfied.

4.3 Theoretical analysis of the algorithm

Here, we will call the loop from line 3 to line 12 the *outer loop*. Again, suppose we want to compute CROT (3.13) of two empirical distributions $\hat{P}_x := \frac{1}{m}\sum_{i=1}^m \boldsymbol{x}_i\delta_{x_i}$ and $\hat{P}_y := \frac{1}{n}\sum_{j=1}^n \boldsymbol{y}_j\delta_{y_j}$. We have the following proposition.

Proposition 1. *For a given z ($> \frac{\lambda}{\beta-1}$), let $J \subseteq \{1,\dots,n\}$ be a subset of indices which sastisfies the following conditions:*

$$\forall j \in J,\ \forall i \in \{1,\dots,m\}, \gamma_{ij} \geq z. \tag{4.12}$$

Suppose we obtained a transport matrix $\boldsymbol{\pi}^{\text{output}}$ by running the outer loop t times satisfying the following condition:

$$\frac{(\frac{1}{m})^{\beta-1} + (\frac{1}{n})^{\beta-1}}{\beta-1} t < \frac{1}{1-\beta} - \left(-\frac{z}{\lambda}\right). \tag{4.13}$$

Then, we send no mass to J.

Proof. Before the outer loop starts,

$$-\frac{z}{\lambda} < \frac{1}{1-\beta} \tag{4.14}$$

holds. Since every element in $\boldsymbol{\theta}^*$ is greater than or equal to $\phi'(0) = \frac{1}{1-\beta}$, the following inequality holds for every i in Algorithm 5:

$$\begin{aligned} \tau_i &\geq \frac{1}{1-\beta} - \phi'\left(\frac{1}{m}\right) && (4.15)\\ &= \frac{1}{1-\beta} - \left(\frac{1}{\beta-1}\left(\left(\frac{1}{m}\right)^{\beta-1} - 1\right)\right) \\ &= -\frac{1}{\beta-1}\left(\frac{1}{m}\right)^{\beta-1}. && (4.16) \end{aligned}$$

Therefore,

$$-\tau_i \leq \frac{1}{\beta-1}\left(\frac{1}{m}\right)^{\beta-1}. \tag{4.17}$$

Similarly, for every j, the following inequality holds:

$$-\sigma_j \leq \frac{1}{\beta-1}\left(\frac{1}{n}\right)^{\beta-1}. \tag{4.18}$$

Therefore, if the algorithm finished running the outer loop t times and the following inequality holds,

$$-\frac{z}{\lambda}+t\times\frac{1}{\beta-1}\left(\frac{1}{m}\right)^{\beta-1}+t\times\frac{1}{\beta-1}\left(\frac{1}{n}\right)^{\beta-1}<\frac{1}{1-\beta}, \tag{4.19}$$

then,

$$\forall i,\tilde{\theta}_{ij}<\frac{1}{1-\beta}\quad\text{if}\quad j\in J \tag{4.20}$$

$$\tag{4.21}$$

holds. Therefore,

$$\forall i,\boldsymbol{\pi}_{ij}^{\text{output}}=0\quad\text{if}\quad j\in J. \tag{4.22}$$

□

Algorithm 5 sends no mass to points in $\{\boldsymbol{y}_j\}_{j=1}^n$ that are more than z away from any point in $\{\boldsymbol{x}_i\}_{i=1}^m$. Although we do not expect to transport any mass to outliers, the optimal solution of the CROT must satisfy the coupling constraint and the condition (4.2) is never satisfied. To ensure (4.2), we consider solving the CROT with only a finite number of updates subsequently. Then, the obtained solution can satisfy (4.2) under a certain condition, although the coupling constraint is not satisfied. This is in stark contrast to the previous works [30, 1], which cannot avoid transporting some mass to outliers.

Chapter 5

Experimental Results

In this chapter, we show our algorithm computes OT robustly when data are contaminated with outliers. In toy experiment 1, we compute CROT of two sets of samples from Gaussian distributions containing outliers. In toy experiment 2, we show that using the transport matrix computed with our algorithm can restore the discrete distribution ignoring the outliers. As an application of the restoration of the discrete distribution, we apply this to a reinforcement learning task in a noisy environment and estimate the reward distribution [59] robustly. Finally, we use our algorithm to detect outliers in the dataset. We show that our method outperforms the previous methods.

5.1 Experiments with synthetic data

In this section, we first show the robustness of our methods to compute CROT with synthetic data.

5.1.1 Toy experiment 1

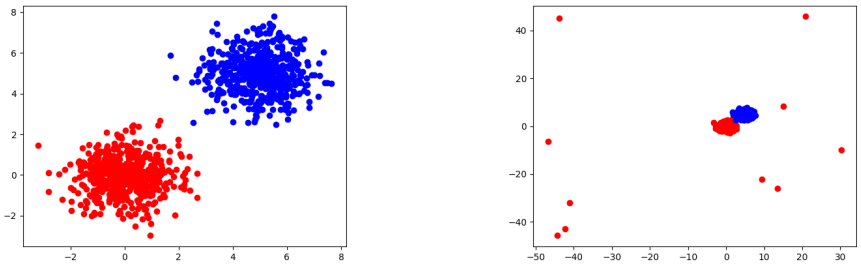
We compute the squared discrete 2-Wasserstein distance ($\gamma_{ij} = ||\boldsymbol{x}_i - \boldsymbol{y}_j||_2^2$) between 2-dimensional empirical probability distributions $\hat{n}_1(\boldsymbol{x})$ and $\hat{n}_2(\boldsymbol{x})$ generated from $N_1(x) := \mathcal{N}(\mathbf{0}, I)$ and $N_2(x) := \mathcal{N}(\mathbf{5}, I)$, respectively. Here, I is the 2-dimensional identity matrix. In this experiment, a few samples from $\mathrm{U}\{(x, y)| -50 \leq x, y \leq 50\}$ will be added to $\hat{n}_2(\mathbf{x})$. We will call this uniform distribution a *contamination distribution*. We look how much the CROT value changes according to the number of samples from the contamination distribution.

First, in order to compute CROT between $\hat{n}_1$ and $\hat{n}_2$, 500 samples were generated from each of $N_1(x)$ and $N_2(x)$. Then, we calculated the squared discrete 2-Wasserstein distance when zero, ten, and twenty-five samples from the contamination distribution were added to $\hat{n}_2$. The figure when ten samples from the contamination distribution were added to $\hat{n}_2$ is shown in Figure 5.1.

The results are shown in Table 5.1. Table 5.1 shows that when the regularizer is the Euclidean norm and β-potential term, we can successfully weaken the influence of outliers.

5.1.2 Toy experiment 2

In toy experiment 1, we confirmed both the Euclidean norm and β-potential enable us to compute OT robustly. Next we will see how *close* the transport matrices computed with



(a) When $\hat{n}_1$ and $\hat{n}_2$ are both clean data. **(b)** When $\hat{n}_2$ is polluted.

Figure 5.1: (a) 500 samples (red) are drawn from $\mathcal{N}([0,0]^\top, I)$ and 500 samples (blue) are from $\mathcal{N}([5,5]^\top, I)$. I is the two-dimensional identity matrix. (b)The figure when $\hat{n}_2$ is polluted with 10 samples from two-dimensional uniform distribution $\mathrm{U}\{(x,y)| -50 \leq x, y \leq 50\}$.

Table 5.1: The squared discrete 2-Wasserstein value with each regularizer according to the number of outliers.

	Zero outlier	Ten outliers	Twenty-five outliers
Boltzman Shannon entropy	53.74	92.19	138.24
Euclidean norm	50.27	49.86	49.29
β-potential $(1 < \beta)$	50.10	50.00	50.18

non-contaminated data and transport matrices with contaminated data are.

How to compare the transport matrices [60]

Let $\hat{P}_x$ and $\hat{P}_y$ be two 1-dimensional empirical probability measures on $\mathbb{R}$, defined respectively by their supports $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$ as $\hat{P}_x = \frac{1}{n}\sum_{i=1}^n x_i \delta_{x_i}$ and $\hat{P}_y = \frac{1}{n}\sum_{j=1}^n y_j \delta_{y_j}$. We consider a non-negative function defined as $(x,y) \in \mathbb{R}^2 \mapsto h(y-x)$ where $h : \mathbb{R} \mapsto \mathbb{R}_+$ and we define the distance matrix $\boldsymbol{\gamma}$ as $\gamma_{ij} := (h(y_j - x_i))_{ij}$. Recall that the OT problem is written as follows:

$$\mathrm{OT}(\hat{P}_x | \hat{P}_y) := \min_{\boldsymbol{\pi} \in \mathcal{G}(\frac{1}{n}, \frac{1}{n})} \sum_{i,j} \pi_{ij} \gamma_{ij}. \tag{5.1}$$

We will leverage the folllowing proposition [60] to compare transport matrices.

Proposition 2 ([60])**.** *Let $\boldsymbol{y}$ be an increasing vector*[1] *of size n. For all strictly convex functions h, if $\boldsymbol{\pi}_*$ is an optimal solution to (5.1), then the vector $\boldsymbol{S}(\boldsymbol{x})$ which sorts $\boldsymbol{x}$ in ascending order can be written as follows:*

$$\boldsymbol{S}(\boldsymbol{x}) = n \boldsymbol{\pi}_*^\top \boldsymbol{x}. \tag{5.2}$$

We compare transport matrices by comparing the corresponding $\boldsymbol{S}(\boldsymbol{x})$ to $\boldsymbol{x}$. We will call the sorting procedures *β-sorting* and *Euclidean-sorting* using the β-potential and Euclidean norm, respectively.

[1] An increasing vector $\boldsymbol{y}_n = (y_1, \ldots, y_n)$ is a vector that satisfies $y_1 < \ldots < y_n$.

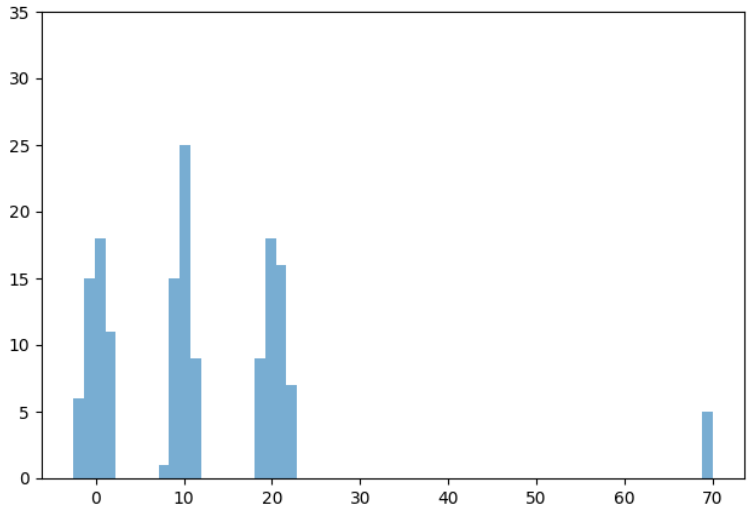


Figure 5.2: 50 samples from $\mathcal{N}(0,1)$, $\mathcal{N}(10,1)$, $\mathcal{N}(20,1)$ each and 5 outliers which are all 70.

We sorted a multimodal distribution shown in Figure 5.2. We sampled 50 samples each from $\mathcal{N}(0,1)$, $\mathcal{N}(10,1)$, $\mathcal{N}(20,1)$ and mixed 5 outliers which are all 70. We denote these 157 samples by $\boldsymbol{x}$ and prepare an increasing vector $\boldsymbol{y} = (\frac{1}{157}, \frac{2}{157}, \ldots, 1)$ of size 157. We then computed P_* when the regularization term is the β-potential term or Euclidean norm. The sorted vectors of β-sorting and Euclidean-sorting are shown in Figure 5.3 and Figure 5.4. They both do not contain any outlier 70 but we can say β-sorting restored the inlier elements (samples from $\mathcal{N}(0,1)$, $\mathcal{N}(10,1)$, $\mathcal{N}(20,1)$) better than Euclidean sorting.

5.2 Application to reinforcement learning

We applied our algorithm to an offline reinforcement learning (RL) [61] task (Figure 5.5). We consider an RL environment with 4×7 blocks RL environment shown in Figure 5.6.

The agent first collects data from running 2000 trials in the environment. In each trial, the agent starts from block 21. The agent can move one block vertically or horizontally in each time step. If the agent reaches block 6, it obtains a reward of 5 points and the trial ends. In addition, if the agent reaches block 0 or block 27, it obtains a reward of 500 points with probability 0.3 or a reward of -1500 points with probability 0.1, and the trial ends; otherwise it does not obtain any rewards and the trial will not end. In each time step, if the trial does not end, the agent obtains a reward of -0.05 points. If the trial does not end after 20 steps, the agent obtains a reward of 0 points, and the trial ends.

We train a policy with a distributional RL [62] approach.

5.2.1 Distributional RL

Here, we model the agent-environment interactions by a Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, R, P, \gamma)$ [63], with $\mathcal{S}$ and $\mathcal{A}$ the state and action spaces, R the reward, a random

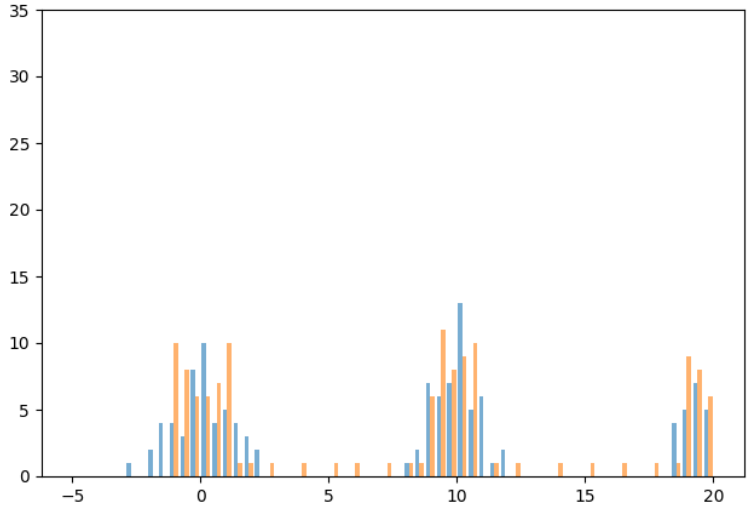


Figure 5.3: The historgram of ordinary samples (blue) and the histogram of sorted elements by Euclidean-sorting (orange).

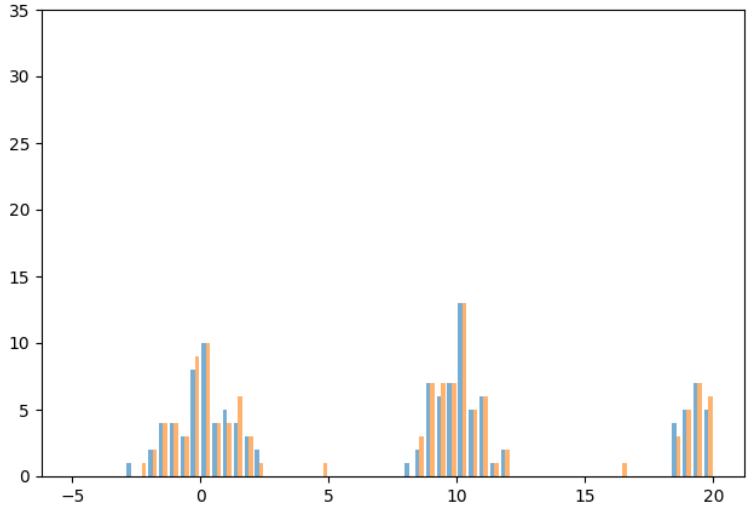


Figure 5.4: The histogram of ordinary samples (blue) and the histogram of sorted elements by β-sorting (orange).

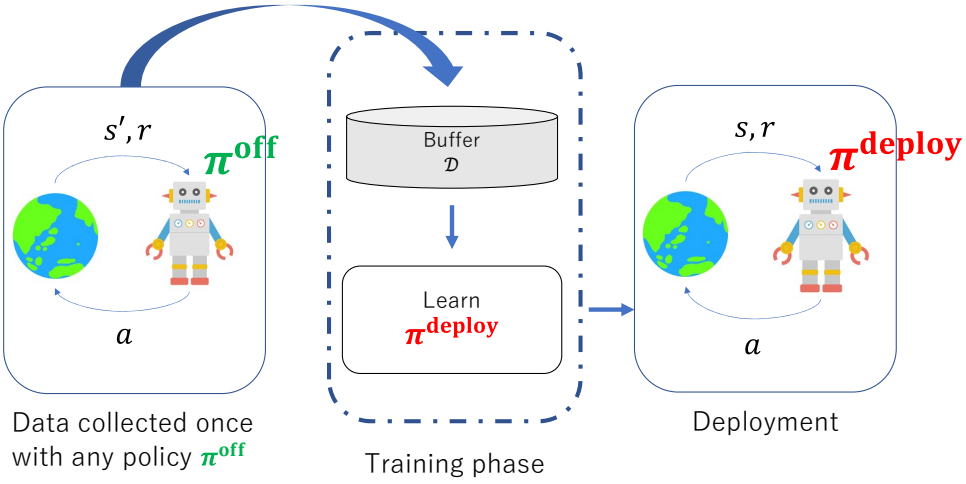


Figure 5.5: An illustration of offline RL. In the data collection phase, an agent in state s interacts with the environment by commiting an action a according to a policy $\boldsymbol{\pi}^{\text{off}}$. Each time it makes an action, it observes the next state s' and obtains reward r. Note that $\boldsymbol{\pi}^{\text{off}}$ is fixed in the data collecting phase. In the training phase, we seek a good policy $\boldsymbol{\pi}^{\text{deploy}}$ using the collected data $\{(s_i, a_i, s_{i+1}, r_{i+1})\}_{i=0,\ldots,N}$. In the end, π^{deploy} will be deployed in the test phase and the quality of the method will be evaluated by the reward it obtained.

Note. Images are cited from https://free-icons.net/life053/ and https://threestardesign.com/earth/.

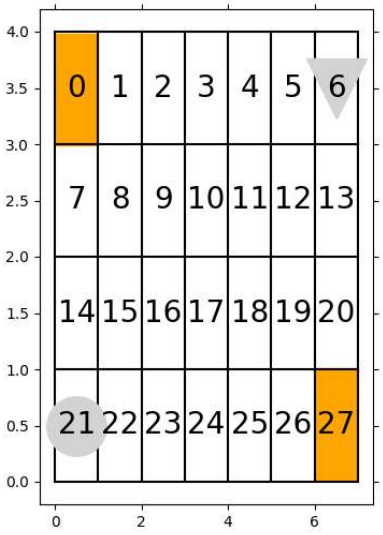


Figure 5.6: The field of RL task. The agent starts from block 21 (circle). If it reaches block 6 (inverted triangle), it obtains a reward of 6 points. If it reaches block 0 or 27, it obtains a reward of 500 points with probability 0.3 or a reward of −1500 points with probability 0.1, and the trial ends; otherwise it will receive no rewards.

variable, $P(s'|s,a)$ the probability of transiting from state s to s' after taking action a, and $\gamma \in [0,1)$ the discount factor. A policy $\pi(\cdot|s)$ maps each state $s \in \mathcal{S}$ to a distribution over $\mathcal{A}$.

For a fixed policy π, *the discounted cumulative reward*, $Z^\pi = \sum_{t=0}^{\infty} \gamma^t R_t$, is a random variable representing the sum of discounted rewards observed along the trajectory of states and actions while following π. Many RL algorithms [64, 65, 66] estimate the action-value function,

$$Q^\pi(s,a) \coloneqq \mathbb{E}[Z^\pi(s,a)] = \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t R(s_t,a_t)\Big] = \mathbb{E}[R(s,a)] + \gamma \mathbb{E}_{a' \sim \pi(\cdot|s), s' \sim P(\cdot|s,a)}[Q(s',a'))], \tag{5.3}$$

$$s_t \sim P(\cdot|s_{t-1},\ a_{t-1}), a_t \sim \pi(\cdot|s_t),\ s_0 = s,\ a_0 = a.$$

A notion called the *Bellman operator* [67] is defined as follows:

$$\mathcal{T}^\pi Q(s,a) \coloneqq \mathbb{E}[R(s,a)] + \gamma \mathbb{E}_\pi[Q(s',a'))]. \tag{5.4}$$

In distributional RL, the distribution over returns (i.e.. the probability law of Z^π) plays a central role and replaces the action-value function. The action-value distribution can be computed through dynamic programming using a *distributional Bellman operator* [20] defined as follows:

$$\begin{aligned} \mathcal{T}^\pi Z(s,a) &\overset{\mathrm{d}}{\coloneqq} R(s,a) + \gamma Z(s',a'), \\ s' &\sim P(\cdot|s,a),\ a' \sim \pi(\cdot|s'), \end{aligned} \tag{5.5}$$

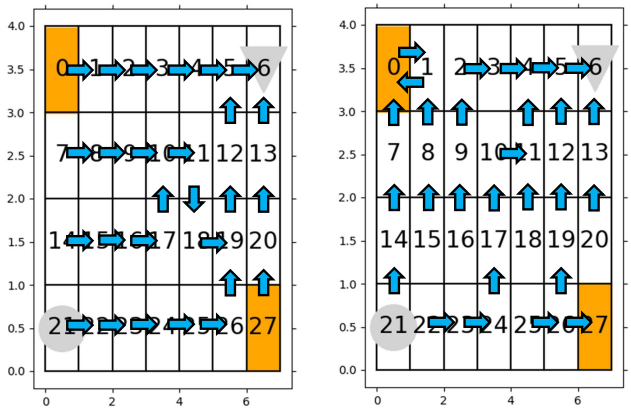


Figure 5.7: The direction which has the largest median of reward for each block (left: β-sorting, right: regular sorting). The β-sorting agent follows path $21 \to 22 \to 23 \to 24 \to 25 \to 26 \to 19 \to 12 \to 5 \to 6$ more often. The regular-sorting agent follows path $21 \to 14 \to 7 \to 0 \to 1 \to 0 \to 1 \to 0 \to 1 \to \cdots$ more often.

where $Y \overset{\mathrm{d}}{:=} U$ denotes equality of probability laws, that is, the random variable Y is distributed according to the same law as U.

To make the problem setting simple, in our problem setting, we made the transition from one state to another deterministic when making the action. For instance, if an agent takes the action of going right in block 10, the agent will move to block 11 with probability 1 and move to block 3, 9, or 17 with probability 0.

5.2.2 Training scheme

An agent collects data by acting a random walk on the data collection phase. Here, a random walk is a uniformly random selection of the next block to go from one block to the next. Suppose a trial followed a trajectory of $\{(s_0, a_0, s_1, r_1), \ldots, (s_{N-1}, a_{N-1}, s_N, r_N)\}$. Then, we used a Monte-Carlo estimation approach to approximate the reward distribution. For all 4×7 pairs of (s, a), we store the discounted reward as shown in Algorithm 6 based on (5.5).

After the data collection phase, we now have data of rewards for each pair (s, a). For each pair (s, a), let us denote the vector of rewards and its size by $\boldsymbol{x}^{\mathrm{r}}_{s,a}$ and $n^{\mathrm{r}}_{s,a}$. Then, we compare two ϵ-greedy agents in the test phase to show that our proposed algorithm estimates the inlier reward distribution robustly and leads to obtaining large reward robustly. A *regular sorting* agent will sort $\boldsymbol{x}^{\mathrm{r}}_{s,a}$ for each pair (s, a) and choose the action with the largest median with probability 0.99 and an action randomly with probability 0.01. A β-sorting agent will sort $\boldsymbol{x}^{\mathrm{r}}_{s,a}$ using Propostion 2. Specifically, we computed CROT between $\boldsymbol{x}^{\mathrm{r}}_{s,a}$ and an increasing vector of size $n^{\mathrm{r}}_{s,a}$. We then used the computed transport matrix $\boldsymbol{\pi}^{\beta}_{s,a}$ and

Algorithm 6 Monte-Carlo approach to approximate the reward distribution

Input: Trajectory of a trial $\{(s_0, a_0, s_1, r_1), \ldots, (s_{N-1}, a_{N-1}, s_N, r_N)\}$, discount factor $\gamma \in [0, 1)$

1: $r_{\text{curr}} \leftarrow r_N$
2: **for** $i = 1$ to N **do**
3: store r_{curr} to (s_{N-i}, a_{N-i})
4: $r_{\text{curr}} \leftarrow r_{N-i} + \gamma\, r_{\text{curr}}$
5: **end for**

Table 5.2: Comparison of the regular sorting agent and β-sorting agent.

	The number of times of reaching block 6	Total reward
β sorting	199990	1000950
Regular sorting	960	70330

computed $S^{\text{pseudo}}(\boldsymbol{x}^{\text{r}}_{s,a}) \coloneqq n^{\text{r}}_{s,a} \boldsymbol{\pi}^{\beta}_{s,a} \boldsymbol{x}^{\text{r}}_{s,a}$ to *pseudo* sort $\boldsymbol{x}^{\text{r}}_{s,a}$. Similar to the regular sorting agent, a β-sorting will choose the action with the largest median in each state with probability 0.99 and an action randomly with probability 0.01.

Note that, since the expected reward finishing at block 0 or 27 is 0 ($=500 \times 0.3 + (-1500) \times 0.1$), the best strategy is to reach block 6.

5.2.3 Results

After 2000 trials in the data collection phase, we ran 200,000 trials in the test phase. We compared the regular sorting agent and β-sorting agent (Table 5.2). In Figure 5.8, we can see that the β-sorting agent estimates the inlier distribution ignoring the outlier distribution.

5.3 Applications to outlier detection

Our algorithm enables us to detect outliers. Let μ_m be a clean dataset and ν_n be a dataset which is polluted with some outlier data. If we compute the transport matrix with μ_m and ν_n and all the element in column j is 0, then the j th data in ν_n is an outlier.

In this experiment, we used Fashion-MNIST as a clean dataset and MNIST data as an outlier dataset. ν_n consists of 9500 images from Fashion-MNIST and 500 images from MNIST. μ_m consists of 10000 images from Fashion-MNIST. We computed the transport

Table 5.3: The percentage of true outliers/inliers detected as outliers/inliers over 50 experiments.

	Outliers	Inliers
One Class SVM	49.8 ± 1.8 %	50.0 ± 0.1 %
Local outlier factor	49.7 ± 3.8 %	99.2 ± 0.1 %
Isolation forest	50.7 ± 10.2 %	65.5 ± 4.4 %
Elliptical envelope	79.9 ± 7.0 %	80.0 ± 4.6 %
MoM-based [24]	79.7 ± 12.6 %	98.9 ± 0.7 %
Baseline technique	92.7 ± 0.4 %	92.8 ± 1.6 %
ROBOT [25]	99.5 ± 0.3 %	84.8 ± 0.5 %
Our Method	99.1 ± 0.7 %	87.3 ± 0.5 %

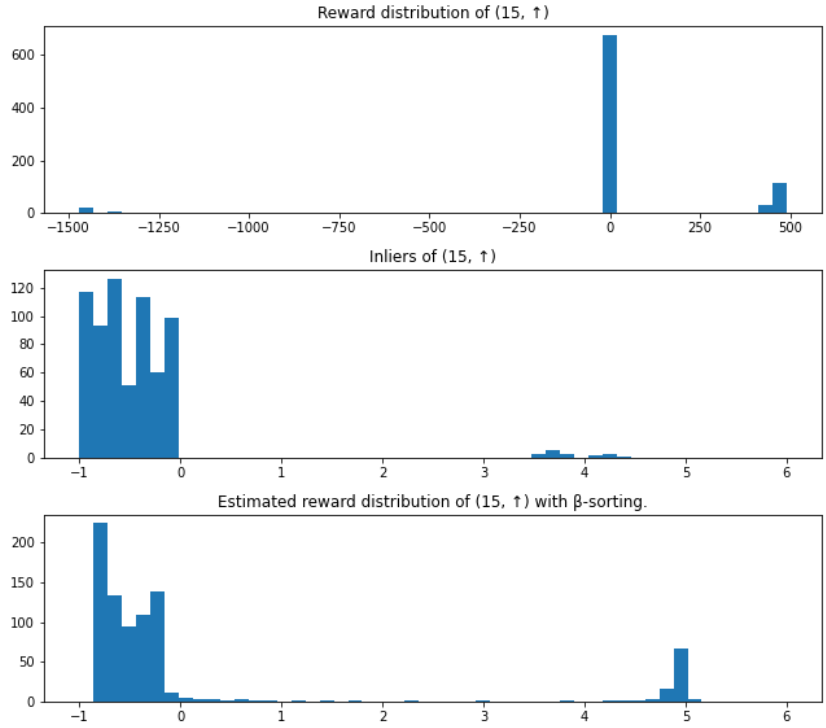


Figure 5.8: Top: The histogram of $\boldsymbol{x}^{\mathrm{r}}_{15,\uparrow}$. Middle: The true inliers of $\boldsymbol{x}^{r}_{15,\uparrow}$. Bottom: The histogram of $S^{\mathrm{pseudo}}(\boldsymbol{x}^{\mathrm{r}}_{15,\uparrow}) = n^{r}_{15,\uparrow}\boldsymbol{\pi}^{\beta}_{15,\uparrow}\boldsymbol{x}^{\mathrm{r}}_{15,\uparrow}$.

Table 5.4: Comparison with Balaji et al. [1] with 1000 datas

	Outliers	Inliers	Run-time
Balaji et al. [1]	89.0 ± 16.9 %	67.0 ± 8.9 %	820 ± 17 seconds
Our Method	96.6 ± 2.0 %	88.0 ± 0.7 %	6 ± 0.2 seconds

matrix with the two datasets and identified the outlying MNIST images. We simply used the raw data to compute the tranport matrix and defined the distance matrix as $C_{ij} = h(\boldsymbol{x}_i - \boldsymbol{y}_j) = ||\boldsymbol{x}_i - \boldsymbol{y}_j||_2^2$, i.e., the Euclidean norm between raw data. We compared the proposed method with the "ROBust Optimal Transport" (ROBOT) method [25] and "MOM-based" method [24] which are methods to compute OT robustly. We also compared our method with a variety of outlier detection algorithms available in Scikit-learn [68]: the one class support vector machine (SVM) [69], local outlier factor [70], isolation forest [71], and elliptical envelope [72]. In the ROBOT method, we set the cost truncation hyperparameter to the 99th percentile of the distance matrix in the subsampling phase [25]. Since the "MoM-based" method does not compute the transport matrix explicitly, we used the Lipschitz function trained as a nueral network in the dual form to detect outliers. We trained the neural network with clean and contaminated data first to approximate the Lipschitz function and determined data as outliers if their Lipschitz function value is less than the 2.5% percentile or greater than 97.5 %.

Hyperparameter selection for our method.

In order to leverage the theoretical analysis, we will estimate the maximum distance between inliers. We assume the distance between any pair of an inlier and an outlier is greater than the distance between inliers. Therefore, if the maximum distance between inliers is z, and run the outer loop t times and to satisfy the following inequality,

$$\frac{(\frac{1}{m})^{\beta-1} + (\frac{1}{n})^{\beta-1}}{\beta - 1} t < \frac{1}{1-\beta} - \left(-\frac{z}{\lambda}\right), \tag{5.6}$$

then Proposition 1 holds. To estimate z, we propose the following heuristic: since we know that μ_m is clean, we subsample two datasets from it and compute the distance matrix. Then, we choose the minimum value for each row and set z with the largest value among them. This procedure is essentially estimating the maximum distance between two samples in the clean dataset. In order to avoid subsampling noise, we used the 95th percentile instead of the maximum.

A natural baseline to compare with will be to identify a data point as an outlier if the minimum distance to the clean dataset is larger than the distance computed in the subsampling phase. We call this methond "the baseline technique".

The results are shown in Table 5.3. One can see that our method has a high performance in detecting not only outliers but also inliers.

Comparison with Balaji et al. [1].

Since Balaji et al. [1] is using CVXPY [34] to compute the transport matrix, their method does not scale to large sample sizes and become extremely time consuming when the dataset gets larger. We conducted the outlier detection experiment with 1000 data. Similar to the previous experiments, the clean dataset μ_m consists of 1000 Fashion-MNIST data and the polluted dataset ν_n consists of 950 Fashion-MNIST data as inliers and 50 MNIST datas as

outliers. Table 5.4 shows the accuracy and the run-time of 10 experiment repetitions. Our method outperforms the method by Balaji et al. not in accuracy of detecting outliers and inliers, but also in the shortness of computation.

Chapter 6

Conclusion

In this thesis, we proposed to robustly approximate OT by regularizing the ordinary OT with the β-potential term. By leveraging the domain of the Fenchel conjugate of the β-potential, our algorithm does not move any probability mass to outliers. Although our algorithm violates the coupling constraint of OT, we showed it robustly approximates OT through several numerical experiments. Specifically, we demonstrated that our proposed method could be used to estimate a probability distribution robustly even in the presence of outliers and detect outliers from a contaminated dataset.

Finally, let us discuss a direction for future work. One possible approach would be to formulate a dual problem that prevents us from moving any probability mass to outliers. Then, we can use neural networks to approximate the Lipschitz functions in the dual problem and should be able to make many deep learning techniques such as GAN robust. Another possible direction would be to consider a situation where both inputs contain outliers. In our method, if both inputs have outliers and the outliers are close to each other, we cannot avoid moving probability mass from outliers to outliers, which may ruin the robust computation of OT.

Bibliography

[1] Yogesh Balaji, Rama Chellappa, and Soheil Feizi. Robust optimal transport with applications in generative modeling and domain adaptation. In *Advances in Neural Information Processing Systems*, 33, 2020.

[2] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *J. Mach. Learn. Res.*, 10:1391–1445, dec 2009.

[3] XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. 56(11):5847–5861, nov 2010.

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[5] S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[6] Introduction to statistical inference— stanford(lecture 16 mle under model misspecification). 2016.

[7] Solomon Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.

[8] Cédric Villani. *Optimal transport – Old and new*, volume 338, pages xxii+973. 01 2008.

[9] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

[10] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision*, pages 435–446, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[11] Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph.*, 34(4), jul 2015.

[12] Arnaud Dessein, Nicolas Papadakis, and Jean-Luc Rouas. Regularized optimal transport and the rot mover's distance. *Journal of Machine Learning*, 25(10):2734–2775, 2018.

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

[14] Marco Cuturi. Sinkhorn distances: lightspeed computation of optimal transport". In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, volume 2, pages 2292–2300, 2013.

[15] Jernej Kos, Ian Fischer, and Dawn Xiaodong Song. Adversarial examples for generative models. *2018 IEEE Security and Privacy Workshops (SPW)*, pages 36–42, 2018.

[16] Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'13, page 1196–1204, Red Hook, NY, USA, 2013. Curran Associates Inc.

[17] A. Basu, I.R. Harris, N.L. Hjort, and M.C. Jones. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 1998.

[18] Takafumi Kanamori and Hironori Fujisawa. Robust estimation under heavy contamination using unnormalized models. *Biometrika*, 102(3):559–572, 05 2015.

[19] Futoshi Futami, Issei Sato, and Masashi Sugiyama. Variational inference based on robust divergences. In *AISTATS*, 2018.

[20] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 449–458. JMLR.org, 2017.

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

[22] G. Monge. *Mémoire sur la théorie des déblais et des remblais*. De l'Imprimerie Royale, 1781.

[23] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017.

[24] Guillaume Staerman, Pierre Laforgue, Pavlo Mozharovskyi, and Florence d ' Alché Buc. When ot meets mom: Robust estimation of wasserstein distance. In *International Conference on Artificial Intelligence and Statistics*, volume 130, pages 136–144. PMLR, 2021.

[25] Debarghya Mukherjee, Aritra Guha, and Justin Solomon. Outlier-robust optimal transport. In *International Conference of Machine Learning*, 2020.

[26] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2017.

[27] Théo Lacombe, Steve Oudot, and Marco Cuturi. Large scale computation of means and clusters for persistence diagrams using optimal transport. In *Advances in Neural Information Processing Systems 2018*, 2018.

[28] Matthew Staib, Sebastian Claici, Justin Solomon, and Stefanie Jegelka. Parallel streaming wasserstein barycenters. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 2644–2655, Red Hook, NY, USA, 2017. Curran Associates Inc.

[29] Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Trans. Graph.*, 34(4), jul 2015.

[30] Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced transport problems. *arXiv preprint arXiv:1607.05816*, 2016.

[31] Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures. *Inventions mathematicae*, 211(3):969–1117, 2018.

[32] Martin Arjovsky, Soumith Chintala, and L ' eon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34nd International Conference on Machine Learning*, 2017.

[33] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, 30, 2017.

[34] Steven Diamond and Stephen Boyd. Cvxpy: A pythonembedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[35] Arkadii Semenovich Nemirovsky and David Borisovich Yudin. Problem complexity and method efficiency in optimization. *John Wiley & Sons Ltd*, 1983.

[36] Mark R Jerrum, Leslie G Valiant, and Vijay V Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, (43):169–188, 1986.

[37] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and system sciences*, 58(1):137–147, 1999.

[38] Leonid Vasilevich Kantorovich and Gennady S Rubinstein. On a space of completely additive functions. *Vestnik Leningrad. Univ*, 13(7):52–59, 1958.

[39] Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Advances in Neural Information Processing Systems*, pages 3440–3448, 2016.

[40] Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-scale optimal transport and mapping estimation. *arXiv:1711.02283 [stat]*, Feburary, 2018.

[41] James B. Orlin. A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78:109–129, 1996.

[42] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[43] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[44] Borwein Jonathan M. Bauschke, Heinz G. Legendre functions and the method of random bregman projections. *Journal of Convex Analysis*, 4(1):27–67, 1997.

[45] H.H. Bauschke and J.M. Borwein. Legendre functions and the method of random bregman projections. *Jounal of Convex Analysis*, 1997.

[46] J.P. Boyle and R.L. Dykstra. A method for finding projections onto the intersection of convex sets in hilbert spaces. *Lecture Notes in Statistics.*, pages 28–47, 1986.

[47] I.S. Dhillon and J.A. Tropp. Matrix nearness problems with bregman divergences. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1120–1146, 2007.

[48] H. H. Bauschke and A. S. Lewis. Dual coordinate ascent methods for non-strictly convex minimization. *Mathematical Programming*, 59(1-3):231–247, 1993.

[49] H. H. Bauschke and A. S. Lewis. Dykstra ' s algorithm with bregman projections: A convergence proof. *Optimization*, 48(4):409–427, 2000.

[50] Heinz H. Bauschke and Jonathan Michael Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Rev.*, 38:367–426, 1996.

[51] John Von Neumann. On rings of operators. reduction theory. *Ann. of Math*, 50(2):401–485, 1949.

[52] V. Ryaben'kii and Semyon Tsynkov. A theoretical introduction to numerical analysis. 2006.

[53] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950*, pages 481–492, Berkeley and Los Angeles, 1951. University of California Press.

[54] William Karush. Minima of functions of several variables with inequalities as side conditions. Master's thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA, 1939.

[55] Nist/sematech e-handbook of statistical methods. https://www.itl.nist.gov/div898/handbook/prc/section1/prc16.htm, 2012.

[56] Benjamin Peirce. Criterion for the rejection of doubtful observations. *Astronomical Journal II*, 45, 1852.

[57] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[58] Frank E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[59] Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Parametric return density estimation for reinforcement learning. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, page 368–375, Arlington, Virginia, USA, 2010. AUAI Press.

[60] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable ranking and sorting using optimal transport. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[61] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020.

[62] Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *AAAI*, 2018.

[63] Martin L. Puterman. Markov decision processes: Discrete stochastic dynamic programming. In *Wiley Series in Probability and Statistics*, 1994.

[64] G. Rummery and Mahesan Niranjan. On-line q-learning using connectionist systems. *Technical Report CUED/F-INFENG/TR 166*, 11 1994.

[65] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. 2013. cite arxiv:1312.5602 Comment: NIPS Deep Learning Workshop 2013.

[66] Christopher J. C. H. Watkins and Peter Dayan. Q-learning. In *Machine Learning*, pages 279–292, 1992.

[67] Richard Bellman. *Dynamic Programming*. Dover Publications, 1957.

[68] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12(null):2825–2830, nov 2011.

[69] Bernhard Schölkopf, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 582–588, Cambridge, MA, USA, 1999. MIT Press.

[70] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, may 2000.

[71] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008.

[72] Peter J. Rousseeuw and Katrien van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.