Université Paris-Saclay
Institut de Mathématique d'Orsay

M1 Internship Report

Stochastic Approximation in Optimal Transport

by

Leonardo MARTINS BIANCO

Supervisor: **Lénaïc CHIZAT**

September 2021

Introduction

This report documents the main discoveries and techniques learned during the internship concluding the first year of masters in applied mathematics at Université Paris-Saclay. The internship was conducted in the *Laboratoire de Mathématiques d'Orsay* under the supervision of Lénaïc Chizat, and lasted four months. The methodology was the same as usual in mathematical research : we held weekly meetings to discuss progress, getting unstuck, and setting goals for the next meeting.

The structure of the report is as follows. The first chapter contains the basic elements of optimal transport theory, kernel methods, and stochastic gradient descent. These will be then "mixed" in the second chapter, where we will apply a SGD to solve for the coefficients of an expansion of the optimal Kantorovich potentials along kernel evaluations at random points sampled from the source and target distributions. In this same chapter we will see that this method needs to be accelerated, and we will see how to do so using random features. The third chapter contains an analysis of the various types of error that are implied in such an algorithm, and a look into why it should be interesting to consider quadratic regularization instead of the typical entropic one. Finally, the fourth chapter develops numerical experiments to validate the theory developed. We emphasize that the original content is contained in the last two chapters, while the first two are merely a necessary review of the existing bibliography.

The motivation for this internship came from noticing that even after acceleration by random features, kernel based algorithms for optimal transport converged too slowly [1]. Our goal was to understand why this happened, and how to remedy it. As this report will show, the explosive nature of the exponential term in the entropic regularization of optimal transport forces the norm of the (unbiased estimate of) gradient to be either enormous or negligible. This explains the instabilities observed as well as the slow convergence. Therefore our proposed solution is to change the entropic regularization by a tamer regularization. We argue that the quadratic one is a good example of such a well-behaved regularization, and thus it might be intrinsically better than the commonly used entropic one for these situations.

Finally, we remark that the interest in these algorithms lie in the search for efficient algorithms for determining the optimal transport between continuous distributions (which can be accessed only by sampling). The problem to be solved is infinite dimensional, therefore the use of kernel methods is very natural. We also believe that this approach should be more robust to higher dimensionality of the source and target distributions, but this is a subject for another internship on itself.

Although this was a very short internship, a brief word of acknowledgement is nevertheless needed. First of all I thank of course my family and Bianca, who are my core support. Then my friends, who heard more than a couple of times whenever I was stuck. I would like to thank Lénaïc Chizat for not only teaching me a lot of theory, but also to value quality over quantity. Finally, I would like the staff of Université Paris-Saclay and the LMO for helping out with all the bureaucracies needed.

Contents

List of Figures

Chapter 1

Mathematical Preliminaries

1.1 Optimal transport

The problem of optimal transport originated as a logistics problem. Suppose for example that you need to transport construction material from source depots to target construction sites while minimizing the cost of this transport. This is the idea behind Monge's problem described below. Of course, today optimal transport is useful for much more abstract applications, notably in machine learning, since computing this cost of transport between data distributions is a (geometrically) meaningful way of comparing them.

1.1.1 Monge's problem and Kantorovich's relaxation

Without further ado, let us define the problem first posed by Monge.

Definition 1.1.1. Given two probability measures $\alpha \in \mathcal{P}(\mathcal{X})$ and $\beta \in \mathcal{P}(\mathcal{Y})$ and a cost function $c : \mathcal{X} \times \mathcal{Y} \to [0, +\infty]$, solve for $T : \mathcal{X} \to \mathcal{Y}$

$$\text{(MP)} \quad \inf_T \left\{ \int c(x, T(x)) d\alpha(x) : T_\# \alpha = \beta \right\}$$

where the *pushforward* $T_\#\alpha$ is the measure satisfying $\forall B \in \mathcal{Y}, T_\#\alpha(B) = \alpha\left(T^{-1}(B)\right)$.

Difficulty. This problem is very difficult to solve due to its constraint : it is technically, not closed under weak convergence [2]. Monge first proposed it in 1781, but it was only by the half of the twentieth century, almost two hundred years later, that significant advances were made after Kantorovich's idea of relaxing the problem to a probabilistic framework.

Kantorovich's relaxation. Instead of dealing with deterministic maps T which intuitively indicate exactly to what point $T(x) \in \mathcal{Y}$ the departing point $x \in \mathcal{X}$ will be transported to, Kantorovich proposed in 1942 to work with maps $\gamma : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ which tells us a *probability* of transporting x to y. This is a much less rigid constraint, and thus a much easier problem to solve. In particular, it will be convex (enabling us to profitably explore its dual), and the set of transport plans is weakly-compact (enabling us to prove the existence of solutions, which is not the case for Monge's problem).

Definition 1.1.2. Given two probability measures $\alpha \in \mathcal{P}(\mathcal{X})$ and $\beta \in \mathcal{P}(\mathcal{Y})$ and a cost function $c : \mathcal{X} \times \mathcal{Y} \to [0, +\infty]$, solve

$$\text{(KP)} \quad \inf_{\gamma} \left\{ \int_{X \times Y} c \, d\gamma : \gamma \in \Pi(\alpha, \beta) \right\}$$

where $\Pi(\alpha, \beta)$ is the set of *transport plans*, that is,

$$\Pi(\alpha, \beta) := \left\{ \gamma \in \mathcal{P}(X \times Y) : (\mathrm{Proj}_{\mathcal{X}})_{\#} \gamma = \alpha, (\mathrm{Proj}_{\mathcal{Y}})_{\#} \gamma = \beta \right\}$$

These constraints assert, as before, that we are indeed transporting the measure α to β. The minimizers to this problem are called the *optimal transport plans*.

1.1.2 Duality

The problem (KP) is a linear optimization problem with convex constraints, since the push-forward map is linear and we then have equality constraints. Thus it will be frequently desirable to work with its dual formulation instead. Later this dual point of view will be even more important, as we will see that the dual of the entropic regularized problem is unconstrained.

Constraints as penalisation. Notice that if you take a general positive measure $\gamma \in \mathcal{M}_+(\mathcal{X} \times \mathcal{Y})$ and bounded continuous functions u and v, then

$$\sup_{u,v} \int_{\mathcal{X}} u \, d\alpha + \int_{\mathcal{Y}} v \, d\beta - \int_{\mathcal{X} \times \mathcal{Y}} \big(u(x) + v(y)\big) \, d\gamma$$

$$= \begin{cases} 0 & \text{if } \gamma \in \Pi(\alpha, \beta) \\ +\infty & \text{otherwise} \end{cases}$$

Since if γ does not agree with, say, α after push-forwarding it by the projection to $\mathcal{X}$, then there is a region of $\mathcal{X}$ where

$$\int_{\mathcal{X}} u \, d\alpha - \int_{\mathcal{X}} u \, d(\mathrm{Proj}_{\mathcal{X}})_{\#}\gamma > 0$$

(Inverting the sign of u if necessary). Then, we can take u arbitrarily large in this region and make the difference above grow arbitrarily. This allows us to write the constraints of (KP) as a penalisation term directly on the objective function, since if the constraints are satisfied we are left with the original problem, and if they are not satisfied we won't have a solution. The rewritten problem reads

$$\inf_{\gamma} \int_{\mathcal{X} \times \mathcal{Y}} c \, d\gamma + \sup_{u,v} \int_{\mathcal{X}} u \, d\alpha + \int_{\mathcal{Y}} v \, d\beta - \int_{\mathcal{X} \times \mathcal{Y}} \big(u(x) + v(y)\big) \, d\gamma$$

As usual, at some step we would like to switch the minimisations by the maximisations, in this case yielding the problem

$$\sup_{u,v} \int_{\mathcal{X}} u \, d\alpha + \int_{\mathcal{Y}} v \, d\beta + \inf_{\gamma} \int_{\mathcal{X} \times \mathcal{Y}} c(x, y) - \big(u(x) + v(y)\big) \, d\gamma$$

Proving that these two problems are equivalent is not trivial and I will not show it here, but a proof can be found in [3].

Reverting the penalisations. The final step to arrive at the dual problem is to see the infimum term above as the penalisation term of the constraint of a maximisation problem. Indeed, this can be done, since by the same technique as before

$$\inf_{\gamma} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y) - \big(u(x)+v(y)\big)\, d\gamma = \begin{cases} 0 & \text{if } u+v \le c \text{ on } \mathcal{X}\times\mathcal{Y} \\ -\infty & \text{otherwise} \end{cases}$$

The optimisation problem corresponding to this penalisation is the dual problem we seek.

Definition 1.1.3. Given $\alpha \in \mathcal{P}(\mathcal{X})$, $\beta \in \mathcal{P}(\mathcal{Y})$ and a cost function $c : \mathcal{X}\times\mathcal{Y} \to [0,+\infty[$, solve

$$\text{(DP)} \qquad \max_{u\in C_b(\mathcal{X}) v\in C_b(\mathcal{Y})} \left\{ \int_{\mathcal{X}} u\, d\alpha + \int_{\mathcal{Y}} v\, d\beta : u+v \le c \right\}$$

1.1.3 Regularization

The dual problem (DP) is still constrained. An approach that has recently [4] become very popular is to regularize optimal transport. Then, the resulting dual is unconstrained, and one can use *Sinkhorn's algorithm* to solve it efficiently in the discrete case (cf. [5]). It is interesting to notice that the idea goes back at least to Schrödinger in problems of statistical physics.

The idea. We will try to search simpler solutions to problem (MK) by penalizing solutions as they get more complex. More precisely, we will penalize solutions by their "distance" to the product measure $\alpha\otimes\beta$. Of course, providing a good notion of distance between measures is something that optimal transport does, but since our problem is itself to determine the optimal cost of transport we need to use some other notion of distance between measures to do so.

φ-divergences. This is the simplest way of comparing measures. As explained in [5], these quantities compare measures pointwise, without introducing any notion of mass transportation. Quoting this reference, "divergences are functionals which, by looking at the pointwise ratio between two measures, give a sense of how close they are". We will define $\mathcal{D}_\varphi$ essentially this way, but with a small modification mainly to ensure that they are continuous functionals for the weak topology of measures.

Definition 1.1.4. Let φ be a convex, lower semi-continuous function such that $\varphi(1)=0$. The φ-divergence $\mathcal{D}_\varphi$ between two probability measures α and β is defined as

$$\mathcal{D}_\varphi(\alpha \mid \beta) \overset{\text{def.}}{=} \int_{\mathcal{X}} \varphi\left(\frac{\mathrm{d}\alpha}{\mathrm{d}\beta}(x)\right) \mathrm{d}\beta(x) + \varphi_\infty \alpha^\perp(\mathcal{X})$$

where $\varphi_\infty := \lim_{x\to+\infty} \varphi(x)/x$ and $\alpha_\perp(\mathcal{X})$ denotes the mass of the part of α that is not absolutely continuous with respect to β in the Radon-Nikodym decomposition of α.

This definition might be a little complicated the first time one sees it, but there are two important points. First, it is essentially $\mathbb{E}\left[\varphi\left(\frac{d\alpha}{d\beta}\right)\right]$. Second, we will be using only two specific

divergences, which are simpler to understand.

Regularized optimal transport. As described at the beginning of the section, we regularize the problem (MK) by penalizing it by a φ-divergence with respect to the product measure.

Definition 1.1.5. The regularized optimal transport problem is defined as

$$(\mathcal{P}_{\varepsilon,\varphi}) \qquad \min_{\gamma\in\Pi(\alpha,\beta)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)\, d\gamma(x,y) + \varepsilon \int_{\mathcal{X}\times\mathcal{Y}} \varphi\left(\frac{d\gamma(x,y)}{d\alpha(x)d\beta(y)}\right) d\alpha(x)d\beta(y)$$

Entropic and quadratic regularizations The entropic regularized optimal transport problem corresponds to taking $\varphi(s) = s\log(s) - s + 1$. The quadratically regularized optimal transport problem corresponds to taking $\varphi(s) = \frac{1}{2}s^2$.

As we now show, the main advantage of regularization is that it yields a dual problem free of constraints, which is much easier to solve with efficient numerical methods.

Dual of the regularized optimal transport. We finally arrive at the problem of main interest in the report. We dualize the regularized optimal transport above.

Proposition 1.1.1. *Consider OT between two probability measures α and β with a convex regularizer φ with domain $\mathbb{R}^+$. Then strong duality holds and $(\mathcal{P}_{\varepsilon,\varphi})$ is equivalent to the following dual formulation $(\mathcal{D}_{\varepsilon,\varphi})$:*

$$\sup_{u,v\in C(\mathcal{X})\times C(\mathcal{Y})} \int_{\mathcal{X}} u(x)\, d\alpha(x) + \int_{\mathcal{Y}} v(y)\, d\beta(y) - \varepsilon \int_{\mathcal{X}\times\mathcal{Y}} \varphi^*\left(\frac{u(x)+v(y)-c(x,y)}{\varepsilon}\right) d\alpha(x)d\beta(y)$$

where φ^* is the Legendre transform of φ, defined by $\varphi^*(p) := \sup_w wp - \varphi(w)$.

Proof. The proof is standard. Penalize the constraints $\left(\mathrm{Proj}_{\mathcal{X}}\right)_{\#}\gamma = \alpha$ and $\left(\mathrm{Proj}_{\mathcal{Y}}\right)_{\#}\gamma = \beta$ to construct the Lagrangian

$$\begin{aligned}\mathcal{L}(\pi,u,v) \overset{\text{def.}}{=} &\int_{\mathcal{X}\times\mathcal{Y}} c(x,y)\, d\pi(x,y) + \varepsilon \int_{\mathcal{X}\times\mathcal{Y}} \varphi\left(\frac{d\pi(x,y)}{d\alpha(x)\, d\beta(y)}\right) d\alpha(x)d\beta(y) \\ &+ \int_{\mathcal{X}} u(x)\left(d\alpha(x) - \int_{\mathcal{Y}} d\pi(x,y)\right) + \int_{\mathcal{Y}} v(y)\left(d\beta(y) - \int_{\mathcal{X}} d\pi(x,y)\right)\end{aligned}$$

as usual, the primal problem then writes $\min_\pi \sup_{u,v} \mathcal{L}(\pi,u,v)$ and the dual problem writes $\sup_{u,v}\min_\pi \mathcal{L}(\pi,u,v)$. One can simplify $\min_\pi \mathcal{L}(\pi,u,v)$ by immediately recognizing the Legendre trasform $\varphi^*(p) := \sup_w wp - \varphi(w) = -\inf_w \varphi(w) - wp$ in it :

$$\min_\pi \mathcal{L} = \int_{\mathcal{X}} u(x)\, d\alpha(x) + \int_{\mathcal{Y}} v(y)\, d\beta(y)$$

$$+\varepsilon \min_\pi \left[\int_{\mathcal{X}\times\mathcal{Y}} \left(\varphi\left(\frac{\mathrm{d}\pi(x,y)}{\mathrm{d}\alpha(x)\mathrm{d}\beta(y)}\right) - \frac{u(x)+v(y)-c(x,y)}{\varepsilon}\frac{\mathrm{d}\pi(x,y)}{\mathrm{d}\alpha(x)\mathrm{d}\beta(y)}\right) \mathrm{d}\alpha(x)\mathrm{d}\beta(y)\right]$$

implying that

$$\min_{\pi} \mathcal{L} = \int_{\mathcal{X}} u(x)\, d\alpha(x) + \int_{\mathcal{Y}} v(y)\, d\beta(y) - \varepsilon \int_{\mathcal{X}\times\mathcal{Y}} \varphi^*\left(\frac{u(x)+v(y)-c(x,y)}{\varepsilon}\right) d\alpha(x) d\beta(y)$$

as we wished to show. □

In our two cases of main interest, the conjugates are easy to calculate. For the entropic regularization one has $\varphi^*(t) = \exp(t) - 1$, and for the quadratic regularization one has $\varphi^*(t) = \frac{t^2}{2}$. Finally, a key proposition we will need states that this problem can be cast as the maximization of an expectation.

Proposition 1.1.2. *The dual formulation* $(\mathcal{D}_{\varepsilon,\varphi})$ *has the following equivalent formulation :*

$$(\mathcal{D}_\varepsilon) \qquad \max_{u,vX,Y\sim\alpha\otimes\beta} \mathbb{E}[f_\varepsilon^{X,Y}(u,v)]$$

where

$$f_\varepsilon^{xy}(u,v) \overset{def.}{=} u(x) + v(y) - \varepsilon\varphi^*\left(\frac{u(x)+v(y)-c(x,y)}{\varepsilon}\right)$$

1.2 Kernels

Motivation. Kernel methods aim at bringing infinite dimensional analyses to a finite dimensional setting. This is useful in a general sense because problems with a very high dimensionality can be treated with tools of infinite dimensional analysis. In particular, for us these methods will be useful because we want to work with the dual regularized optimal transport problem in the case where the source and target distributions are continuous and this problem is infinite dimensional.

1.2.1 Reproducing Kernel Hilbert Spaces

Basic definitions. We start with an abstract definition of what are the spaces that we are interested in.

Definition 1.2.1. Let $\mathcal{H}$ be a Hilbert space of functions $f : \mathcal{X} \to \mathbb{R}$ defined on a non-empty set $\mathcal{X}$. For a fixed point $x \in \mathcal{X}$, denote by L_x the evaluation map : $L_x(f) = f(x)$. The space $\mathcal{H}$ is said to be a Reproducing Kernel Hilbert Space (RKHS) if L_x is continuous $\forall x \in \mathcal{X}$.

Definition 1.2.2. Let $\mathcal{H}$ be a Hilbert space of $\mathbb{R}$-valued functions defined on a non-empty set $\mathcal{X}$. A function $\kappa : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is called a reproducing kernel of $\mathcal{H}$ if it satisfies the so-called reproducing properties :

$$(1) \quad \forall x \in \mathcal{X}, \kappa(\cdot, x) \in \mathcal{H}$$
$$(2) \quad \forall x \in \mathcal{X}, \forall f \in \mathcal{H}, \left\langle f, k(\cdot,x)\right\rangle_{\mathcal{H}} = f(x)$$

Observe that the notion of a reproducing kernel does not appear in the definition of a RKHS. The following theorem establishes the link between these two definitions.

Theorem 1.2.1. *A Hilbert space $\mathcal{H}$ is a RKHS if and only if $\mathcal{H}$ has a reproducing kernel.*

One can prove [6] that if a reproducing kernel exists, it is unique. Up until here we have been purely analytic in these definitions. Let us now show the link of these notions to machine learning and optimization.

1.2.2 Positive definite kernels and Moore-Aronsajn

Treating non-linearity. In machine learning one often has to work with data $x \in \mathcal{X}$ that is not well described by linear models and methods, i.e., data arising from non-linear phenomena or better described by non-linear structures. A typical example of this is non-linear classification in supervised learning. This is complicated for obvious reasons. The ideal situation would be one where one can encode the non-linear information of the problem in a map $\varphi : \mathcal{X} \to \mathcal{H}$ in such a way that in the new space $\mathcal{H}$ the model describing the data is linear. One would then be able to fit the linear model in $\mathcal{H}$ and somehow return information to the original space $\mathcal{X}$. This is the basic idea of how kernel methods are used in machine learning. The map φ is called the feature map, and $\mathcal{H}$ is called the feature space. The function $\kappa : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ containing the information of the inner products is what is called a kernel in machine learning : $\kappa(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}$. This is very important because of something called the "kernel trick", which states that for the procedure just described, one does not need to know the feature function φ (which would be very difficult to find since it contains non-linear information), but only these inner products, i.e., only the kernel function matters. So in reality one does things in the opposite sense : one fixes a function to be the kernel and then take the feature map φ and feature space $\mathcal{H}$ arising as a consequence.

Definition 1.2.3. Let $\mathcal{X}$ be a non-empty set. A function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is said to be a positive-definite kernel if there exists a real Hilbert space $\mathcal{H}$ and a map $\varphi : \mathcal{X} \to \mathcal{H}$ such that $\forall x, y \in \mathcal{H}$,

$$k(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}$$

The link between the purely analytic definition of RKHS and reproducing kernels to the machine learning oriented definition of a positive-definite kernel is made by the Moore-Aronsajn theorem, stated below.

Theorem 1.2.2. (Moore-Aronsajn) *We have the equivalence*

$$\kappa \text{ reproducing kernel} \iff \kappa \text{ positive-definite kernel}$$

The Hilbert space $\mathcal{H}$ associated to κ when seen as a positive-definite kernel is a RKHS.

Wrapping it up. When proving Moore-Aronsajn's theorem, one needs to build a RKHS starting from the given kernel. The idea is to consider all functions that can be expanded as

$$f = \sum_{i=1}^{n} c_i \kappa(x_i, \cdot)$$

for any number n of terms and any n points $x_i \in \mathcal{X}$. Therefore one sees that kernels evaluations $\kappa(x_i, \cdot)$ can be seen as basis functions for the RKHS created. What we will do is to go the other way around : we will try to perform such an expansion in kernel evaluations for the potentials u^* and v^*, solutions to $(\mathcal{D}_\varepsilon)$. Of course, given an arbitrary kernel these kernels might not belong to the associated RKHS. This is why we will consider only "universal kernels", which are those having a dense RKHS in the space of real continuous functions on $\mathcal{X}$ and thus able to approximate arbitrarily well even functions outside the RKHS.

1.3 Classical Stochastic Gradient Descent

1.3.1 Acceleration by randomness

The idea. The so-called "gradient methods" are one of the best known family of methods to solve an optimization problem. The idea is to maximize a function f by following its gradient at each iteration (or its opposite in the case of a minimization problem). However, nowadays we are frequently faced with objective functions of the type $f = \sum_{i=1}^n f_i$ with an enormous number n of observations. By linearity, the gradient of such a function is $\nabla f = \sum_{i=1}^n \nabla f_i$, therefore evaluating it requires the calculation of all n different gradients ∇f_i, which is impractical for many modern applications of machine learning. The idea behind stochastic gradient descent is to use a bit of randomness to our favor. Instead of using the full gradient $\nabla f|_t$ to perform the ascent (or descent) at step t, we use an unbiased estimation g_t of it which is easier to calculate and we expect that this unbiasedness imply convergence on average for the algorithm.

Definition 1.3.1. For $\theta \in \mathbb{R}^d$, consider the optimization problem $\min f(\theta)$. Starting from θ_0 and considering a step-size sequence $(\gamma_t)_{t\geqslant 0}$, let g_t be an unbiased estimation of $f'(\theta_{t-1})$, i.e., $\mathbb{E}\left[g_t(\theta_{t-1}) \mid \theta_{t-1}\right] = f'(\theta_{t-1})$. Then, the iterations

$$\theta_{t+1} = \theta_t - \gamma_t g_t(\theta_t)$$

define the stochastic gradient descent algorithm.

1.3.2 Convergence

Having given the definition of the SGD algorithm, we now state the convergence results that will be needed.

Theorem 1.3.1. *Assume that f is convex, B-Lipschitz and admits a minimizer θ_* that satisfies $\|\theta_* - \theta_0\| \leqslant D$. Assume that g_t is an unbiased estimation of the gradient, i.e., that $\mathbb{E}\left[g_t(\theta_{t-1}) \mid \theta_{t-1}\right] = f'(\theta_{t-1}), \forall t$, and that it is bounded, i.e., that $\left\|g_t(\theta_{t-1})\right\|_2^2 \leqslant B^2, \forall t$ almost surely. Then, by choosing a decreasing sequence of step-sizes $\gamma_t = (D/B)/\sqrt{t}$, the iterates $(\theta_t)_{t\geqslant 0}$ of SGD satisfy*

$$\mathbb{E}\left[f\left(\bar{\theta}_t\right) - f(\theta_*)\right] \leqslant DB\frac{2+\log(t)}{\sqrt{t}}$$

where $\bar{\theta}_t = \left(\sum_{s=1}^t \gamma_s \theta_{s-1}\right) / \left(\sum_{s=1}^t \gamma_s\right)$ is the weighted average of the iterates.

A proof can be found in [7]. It is important to remark that the same bound is obtained by projecting the iterates on a ball of finite radius centered on θ_0. This is necessary when the function B is not globally Lipschitz but is Lipschitz inside the said ball.

Remarks. For the interpretation of our numerical results in Chapter 4, it is important to remark that the convergence in Theorem 1.3.1 is proportional to the product of the distance D between the initial point and the typically unknown solution and the Lispchitz constant B of the objective function (inside the projecting ball if we project). This constant might be large, implying that even though in theory the algorithm converges with infinite steps, in practice its progress can halt after some thousand iterations. For example, if $DB = 1000$, the bound garantees that the expected error should be less than 60. Running an additional 1000 iterations only drops the bound to 57.6. It is therefore common to execute the algorithm, let it run for the first few effective thousands iterations, observe the behaviour of the coefficients obtained, and then adjusting parameters to re-execute the algorithm with a smaller initial step-size and initial point equal to the final iteration of the previous execution. It is also theoretically desirable to keep B as small as possible, and we will see in Section 4.1 that changing the entropic regularization for the quadratic one significantly decreases it.

Chapter 2

Kernel Stochastic Gradient Descent

2.1 Kernel stochastic gradient descent

2.1.1 Stochastic approximation

We have seen that the dual problem $(\mathcal{D}_\varepsilon)$ we want to solve can be written generally as the maximization of an expectation. This formulation makes it natural to try applying methods of stochastic optimization to solve it numerically.

Equations for stochastic gradient descent. Were u and v finite dimensional, the equations corresponding to a stochastic gradient descent (SGD) applied to the problem $(\mathcal{D}_\varepsilon)$ would be

$$\begin{cases} u^{(k+1)} & \overset{\text{def.}}{=} & u^{(k)} + \frac{C_u}{\sqrt{k+1}} \nabla_u f_\varepsilon^{x_{k+1}, y_{k+1}} \left(u^{(k)}, v^{(k)}\right) \\ v^{(k+1)} & \overset{\text{def.}}{=} & v^{(k)} + \frac{C_v}{\sqrt{k+1}} \nabla_v f_\varepsilon^{x_{k+1}, y_{k+1}} \left(u^{(k)}, v^{(k)}\right) \end{cases}$$

where $k \geq 0$ and we take initial functions $u^{(0)}$ and $v^{(0)}$ to be identically zero. However, when α and β are continuous we only know that the solutions to $(\mathcal{D}_\varepsilon)$ are on a Hilbert space of functions of infinite dimension [8]. Therefore we won't apply the theory of SGD but its generalization to Hilbert spaces, which deals exactly with problems of this type.

Stochastic approximation in Hilbert spaces. As developed in [9], the strategy is the following. First, suppose you can identify the Hilbert space containing the solution g as a RKHS for some kernel k. Then, you can use the reproducing property to exchange any evaluation $g(x)$ appearing in your optimization problem by $\langle g, k(x, \cdot)\rangle$. We can then rigorously take gradients of functions of g by using Riesz representation theorem, which characterizes $\nabla\xi(p)$ (for a generic function ξ) as being the vector such that $D\xi(p) \cdot v = \langle \nabla\xi(p), v\rangle$. For example, the following calculation will be important for us. Let $L_x(g) := g(x) = \langle g, k(x, \cdot)\rangle$ be the evaluation operator. It is linear in g, therefore one sees that $DL_x(g) = L_x, \forall g$. Now $DL_x(g) \cdot \delta = L_x(\delta) = \langle \delta, k(x, \cdot)\rangle$. Using the characterization just described, we conclude that $(\nabla L_x)(g) = k(x, \cdot)$. While doing calculations, we might abuse the notation and express this fact as $\nabla_u \langle u, k(x, \cdot)\rangle = k(x, \cdot)$, because once we have a function on two variables u and v we will want to consider one of them fixed and take the gradient of the resulting function in the way just described. We are therefore able to take gradients of $f_\varepsilon^{x,y}$ in the RKHS and ascent along them. The second situation that can occur is that the solution g is not in a RKHS, and this intuitively means that g is not sufficiently regular or smooth. Remember though that RKHS are usually seen as function spaces inside $C(\mathcal{X})$, and if we build a dense RKHS it should be able to approximate the solution outside it anyway. Of course, we will then have

an extra error term corresponding to the error of approximation of a function g outside the RKHS by a function $\hat{g}$ inside it. This $\hat{g}$ can be obtained by the same way as we would obtain a solution if it were inside the RKHS.

2.1.2 The algorithm

Now that we described the approach to be used, let us write the equations properly. We had

$$f_\varepsilon^{x,y}(u,v) = u(x) + v(y) - \varepsilon\varphi^*\left(\frac{u(x)+v(y)-c(x,y)}{\varepsilon}\right)$$

The potentials u and v might not be in the RKHS associated to the chosen kernel k, but in such a case we approximate them by $\tilde{u}$ and $\tilde{v}$ that are. Then we expand the evaluations as inners products using the reproducing property

$$f_\varepsilon^{x,y}(\tilde{u},\tilde{v}) = \langle \tilde{u}, k(x,\cdot)\rangle + \langle \tilde{v}, k(y,\cdot)\rangle - \varepsilon\varphi^*\left(\frac{\langle \tilde{u}, k(x,\cdot)\rangle + \langle \tilde{v}, k(y,\cdot)\rangle - c(x,y)}{\varepsilon}\right)$$

Now calculate $\nabla_u f_\varepsilon^{x,y}(\tilde{u},\tilde{v})$. By this we mean the gradient of the function obtained by fixing $\tilde{v}$. We calculate easily that

$$\nabla_u f_\varepsilon^{x,y}(\tilde{u},\tilde{v}) = k(x,\cdot)\left(1-\left(\varphi^*\right)'\left(\frac{\langle \tilde{u}, k(x,\cdot)\rangle + \langle \tilde{v}, k(y,\cdot)\rangle - c(x,y)}{\varepsilon}\right)\right)$$

and in a similar way

$$\nabla_v f_\varepsilon^{x,y}(\tilde{u},\tilde{v}) = k(y,\cdot)\left(1-\left(\varphi^*\right)'\left(\frac{\langle \tilde{u}, k(x,\cdot)\rangle + \langle \tilde{v}, k(y,\cdot)\rangle - c(x,y)}{\varepsilon}\right)\right)$$

Now we can perform gradient ascent in the RKHS with these gradients. Let $\tilde{u}^{(0)} = 0$ and $\tilde{v}^{(0)} = 0$. Then at iteration $k+1$ we sample $x_{k+1} \sim \alpha$ and $y_{k+1} \sim \beta$ and define new iterates by

$$\tilde{u}^{(k+1)} = \tilde{u}^{(k)} + \frac{C_u}{\sqrt{1+k}} k(x_{k+1},\cdot)\left(1-\left(\varphi^*\right)'\left(\frac{\langle \tilde{u}^k, k(x_{k+1},\cdot)\rangle + \langle \tilde{v}^{(k)}, k(y_{k+1},\cdot)\rangle - c(x_{k+1},y_{k+1})}{\varepsilon}\right)\right)$$

and

$$\tilde{v}^{(k+1)} = \tilde{v}^{(k)} + \frac{C_v}{\sqrt{1+k}} k(y_{k+1},\cdot)\left(1-\left(\varphi^*\right)'\left(\frac{\langle \tilde{u}^k, k(x_{k+1},\cdot)\rangle + \langle \tilde{v}^{(k)}, k(y_{k+1},\cdot)\rangle - c(x_{k+1},y_{k+1})}{\varepsilon}\right)\right)$$

This can be rewritten in a more amenable form as

$$\tilde{u}^{(k+1)} = \tilde{u}^{(k)} + \omega^{(k+1)} k(x_{k+1},\cdot)$$

and if $C_u = C_v$ (something that in practice will rarely differ), then

$$\tilde{v}^{(k+1)} = \tilde{v}^{(k)} + \omega^{(k+1)} k(y_{k+1},\cdot)$$

i.e.

$$\begin{cases} \tilde{u}^{(k+1)} = \sum_{i=0}^{k+1} \omega^{(i)} k(x_i,\cdot) \\ \tilde{v}^{(k+1)} = \sum_{j=0}^{k+1} \omega^{(j)} k(y_j,\cdot) \end{cases}$$

Each new iteration requires evaluating all previous ones on newly sampled data x_{k+1} and y_{k+1}. Thus the complexity of the algorithm grows with the number of iterations, making it slow. We need ways to accelerate it. This will be done by approximating the kernel k using random features.

2.2 Random features

2.2.1 Random feature approximation of a kernel

The idea. The goal of the random features method is to find a map $\Phi : \mathcal{H} \to \mathbb{R}^m$ such that

$$\kappa(x, y) \approx \Phi(x)^\top \Phi(y)$$

We will call the vector $\Phi(x) \overset{\text{def.}}{=} \frac{1}{\sqrt{m}} \big(\phi_1(x), \dots, \phi_m(x)\big)$ the random feature vector, and each component ϕ_i a random feature. This is useful because, as we will see, Φ maps to a relatively low dimensional space and it allows us to apply fast linear learning methods.

Finding Φ. In practice, the idea to find Φ is frequently the same. First you choose some symmetry for your kernel. Examples of symmetries are translation invariance (i.e. $\kappa(x, y) = t(x - y)$ for some t), rotation invariance, dot product invariance, and so on. Then, since kernels are positive-definite functionals, you search for a theorem characterizing the combination of positive-definiteness and the chosen symmetry. By this we mean, a theorem of the form

$$\text{Symmetry + positive-definiteness} \iff \text{Decomposition of } \kappa \text{ as an expectation}$$

A clarifying example. Random features were introduced in [10], and there they considered translation invariant kernels. In such a case, remember Bochner's theorem.

Theorem 2.2.1 (Bochner). *A continuous kernel $\kappa(x, y) = t(x - y)$ on $\mathbb{R}^d$ is positive-definite if and only if $\kappa(\delta)$ is the Fourier transform of a non-negative measure. Under proper scaling this measure is a probability measure $p(\omega)$, and if we define $\zeta_\omega(x) := e^{i\omega x}$ then we have the following decomposition as an expectation :*

$$\kappa(x - y) = \int_{\mathbb{R}^d} p(\omega) e^{i\omega(x-y)} d\omega = \mathbb{E}_\omega \left[\zeta_\omega(x)\zeta_\omega(y)^*\right]$$

Random features can then easily be obtained by approximating this expectation by Monte-Carlo. Sample $\omega_1, \dots, \omega_m \sim p$ and then

$$\mathbb{E}_\omega \left[\zeta_\omega(x)\, \zeta_\omega(y)^*\right] \approx \frac{1}{m} \sum_{i=1}^{m} \zeta_{\omega_i}(x)\zeta_{\omega_i}(y) = \Phi(x)^*\Phi(y)$$

where $\Phi(x) \overset{\text{def.}}{=} \frac{1}{\sqrt{m}} \left(e^{i\omega_1 x}, \dots, e^{i\omega_m x}\right)$.

The other way around. We can also (and we will in our case) go the other way around. Start with some feature function $\phi_\theta(x)$ depending on a parameter θ to be sampled randomly according to some distribution μ on some parameter space $\mathcal{V}$. Then define

$$\kappa(x, y) \overset{\text{def.}}{=} \mathbb{E}_\mu \left[\phi_\theta(x)\, \phi_\theta(y)\right] = \int_{\mathcal{V}} \phi_\theta(x)\phi_\theta(y)\, d\mu(\theta)$$

Of course, after sampling $\theta_1, \dots, \theta_m \sim \mu$ we have that

$$\kappa(x, y) \approx \Phi(x)^\top \Phi(y)$$

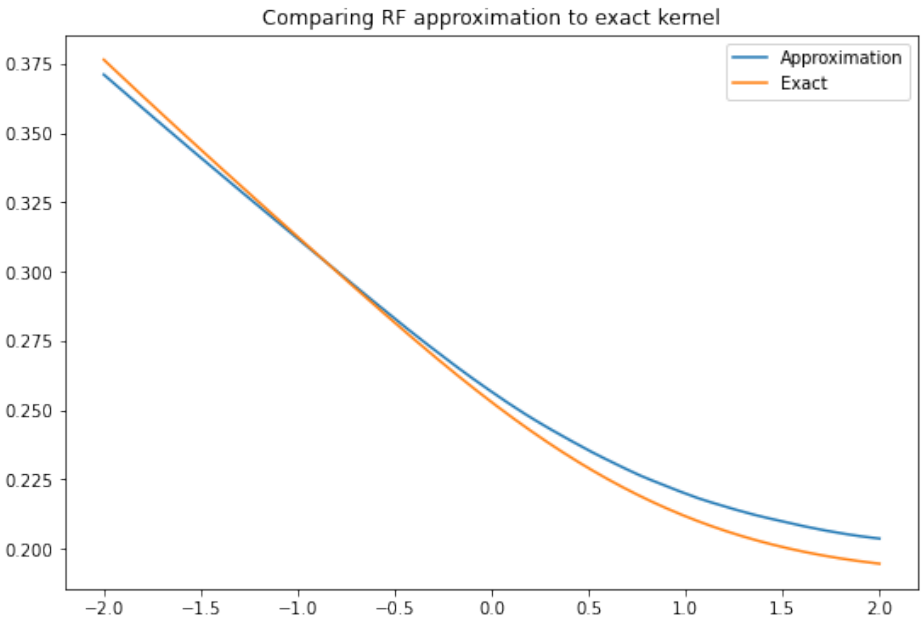

Figure 2.1: Approximating the exact ReLU kernel

for

$$\Phi(x) \overset{\text{def.}}{=} \frac{1}{\sqrt{m}} \left(\phi_{\theta_1}(x), \ldots, \phi_{\theta_m}(x)\right)$$

It is then necessary to verify that κ indeed defines a positive-definite kernel, but this depends on the features chosen.

2.2.2 Neural network inspired random features

We will folow [11], where it is proposed the use of common homogeneous activation functions of neural networks as feature functions. We concentrate in the use of ReLU features

$$\phi^{\text{ReLU}}_{\theta=(\omega,R)}(x) \overset{\text{def.}}{=} \max(0, \omega x + R) = (\theta^\top z)_+$$

where $z = (x, 1)$. The associated kernels are indeed kernels and these have explicit expressions [11]. In Figure 2.1 we can observe the exact ReLU kernel (in this example evaluated at $x = -1$) being approximated by its random feature counterpart. We will comment on their approximation capacity in Section 3.3.

2.3 Kernel SGD with random features approximation

2.3.1 Random feature expansion of Kantorovich's potentials

Now that random features have been introduced, observe that by substituting the (ReLU) random feature approximation of the kernel in the previous formulas for our iterations, we get

$$\tilde{u}^{(k+1)} = \sum_{i=0}^{k+1} \omega^{(i)} k(x_i, \cdot) \approx \sum_{i=0}^{k+1} \omega^{(i)} \Phi(x_i)^\top \Phi(\cdot) = \left(\sum_{i=0}^{k+1} \omega^{(i)} \Phi(x_i)\right)^\top \Phi(\cdot)$$

the term in parentheses is a vector of fixed dimension equal to the number m of random features and it gets updated at each iteration. It is therefore natural to view it as a (finite-dimensional) vector of coefficients $\mathbf{w}_u$ to be determined by maximizing the dual problem of regularized optimal transport, and over which we *can* apply SGD. That is, we are now equivalently performing a regression

$$\tilde{u}(x) = \mathbf{w}_u^\top \Phi(x)$$

where the iterations can now updated by proper SGD (and of course a similar reasoning stands for v). It might seem a trivially simple idea, but it wouldn't have been thought if we were not trying to approximate our kernel to accelerate the evaluation of a "kernel machine". It is easy to derive the equations for this new iterative scheme. Substituting this new expression for $\tilde{u}$ and $\tilde{v}$ into the expression for $f_\varepsilon^{x,y}$ and taking the gradients we obtain

$$\begin{cases} \mathbf{w}_u^{(k+1)} = \pi_B \left(\mathbf{w}_u^{(k)} + \frac{C_u}{\sqrt{k+1}} \Phi\left(x_{k+1}\right) \left(1 - \lambda_{k+1}\right) \right) \\ \mathbf{w}_v^{(k+1)} = \pi_B \left(\mathbf{w}_v^{(k)} + \frac{C_v}{\sqrt{k+1}} \Phi\left(y_{k+1}\right) \left(1 - \lambda_{k+1}\right) \right) \end{cases}$$

where

$$\lambda_{k+1} \overset{\text{def.}}{=} \left(\varphi^*\right)' \left(\frac{\left(\mathbf{w}_u^{(k)}\right)^\top \Phi\left(x_{k+1}\right) + \left(\mathbf{w}_v^{(k)}\right)^\top \Phi\left(y_{k+1}\right) - c\left(x_{k+1}, y_{k+1}\right)}{\varepsilon} \right)$$

π_B is the projection on the ball of radius B (it is explained in 4 why we do this), and $x_{k+1} \sim \alpha$, $y_{k+1} \sim \beta$. Remember that to use convergence bounds Theorem 1.3.1, we must actually look at the weighted average of the iterates.

Chapter 3

Error Analysis

3.1 General error decomposition

Wasserstein distances. Given a source distribution α and a target distribution β, the solution to the Kantorovich problem (KP) is a coupling γ minimizing the cost of transport from α to β. The p-th root of this cost is in fact a distance function (in the sense of a metric space), and it is called the p-Wasserstein distance $\mathcal{W}^p(\alpha, \beta)$. The most common one is the 2-Wasserstein distance, and for simplicity it will be denoted by $\mathcal{W}(\alpha, \beta)$. This quantity can be estimated by solving the regularized problem using SGD as described previously, recovering the coupling γ_ε from the potentials, and then calculating $\mathcal{W}_\varepsilon^{k,m,B}(\alpha, \beta) \overset{\text{def.}}{=} \sqrt{\int c d\gamma_\varepsilon}$. The difference of this estimation to the actual value of the Wasserstein distance is a sensible way of measuring error, and such error is what we will analyze in this chapter.

A general decomposition of the error. There are multiple sources of error in the approach used on this report. First, there is a regularization error E_{Reg} : we did not solve Kantorovich's problem (KP), but an ε-regularized version of it. When $\varepsilon \neq 0$ the regularized problem is an approximation to the non-regularized problem, and thus we have an error associated with it. After picking a regularization (in our case either the entropic or the quadratic one), one must show that this error goes to zero when ε tends to zero and if possible describe at what rate this convergence takes place. We will do so for the quadratic regularization in the next paragraph, and although it is a simple result it is a new one and one of the main theoretical results of the internship.

We have also error contribution coming from the fact that we perform a finite number k of iterations, and thus use a finite number of samples from our distributions in our algorithm. This is the error of SGD, which we bounded in Theorem 1.3.1 and denoted by E_{Sample}. We can also include the error E_{RF} coming from the fact that we expand our potentials in a finite number m of random features. Finally it is important to include the error coming from the projection of the coefficients $\mathbf{w}_u$ and $\mathbf{w}_v$ back to a ball of radius B at each iteration (the maximizer might be outside of such a ball).

We write $\mathcal{W}_\varepsilon^{k,m,B}$ to denote the square root of the optimal cost of transport obtained by taking all these factors into account (this is the quantity that we really obtain once we run the algorithm in a computer). The total error is then

$$E = \left| \mathcal{W}(\alpha, \beta) - \mathcal{W}_\varepsilon^{k,m,B}(\alpha, \beta) \right|$$

In practice, we bound it by decomposing it using triangle inequality into multiple terms,

each corresponding a parameter in its "ideal" limit (infinite iterations, zero regularization, no projection, and infinite random features). Explicitly, we calculate (ommiting α and β for simplicity of notation)

$$\begin{aligned} E = |\mathcal{W} - \mathcal{W}_\varepsilon^{k,m,B}| &= |\mathcal{W} + (\mathcal{W}_\varepsilon^{\infty,\infty,\infty} - \mathcal{W}_\varepsilon^{\infty,\infty,\infty}) - \mathcal{W}_\varepsilon^{k,m,B}| \\ &\leqslant \underbrace{|\mathcal{W} - \mathcal{W}_\varepsilon^{\infty,\infty,\infty}|}_{\text{regularization error}} + |\mathcal{W}_\varepsilon^{\infty,\infty,\infty} - \mathcal{W}_\varepsilon^{k,m,B}| \end{aligned}$$

This second term might be split up in various ways, depending on which parameter we want to study the error associated with. We will do then the calculation for each parameter.

$$\begin{aligned} E &\leqslant |\mathcal{W} - \mathcal{W}_\varepsilon^{\infty,\infty,\infty}| + |\mathcal{W}_\varepsilon^{\infty,\infty,\infty} + \left(\mathcal{W}_\varepsilon^{\infty,\infty,B} - \mathcal{W}_\varepsilon^{\infty,\infty,B}\right) - \mathcal{W}_\varepsilon^{k,m,B}| \\ &\leqslant \underbrace{|\mathcal{W} - \mathcal{W}_\varepsilon^{\infty,\infty,\infty}|}_{\overset{\text{def.}}{=} E_{\text{Regularization}}} + \underbrace{|\mathcal{W}_\varepsilon^{\infty,\infty,\infty} - \mathcal{W}_\varepsilon^{\infty,\infty,B}|}_{\overset{\text{def.}}{=} E_{\text{Projection}}} + |\mathcal{W}_\varepsilon^{\infty,\infty,B} - \mathcal{W}_\varepsilon^{k,m,B}| \end{aligned}$$

Finally we split the third term as

$$|\mathcal{W}_\varepsilon^{\infty,\infty,B} - \mathcal{W}_\varepsilon^{k,m,B}| \leq \underbrace{|\mathcal{W}_\varepsilon^{\infty,\infty,B} - \mathcal{W}_\varepsilon^{\infty,m,B}|}_{E_{\text{Random Features}}} + \underbrace{|\mathcal{W}_\varepsilon^{\infty,m,B} - \mathcal{W}_\varepsilon^{k,m,B}|}_{E_{\text{Optimization}}}$$

Yielding the complete decomposition

$$E \leq E_{\text{Regularization}} + E_{\text{Projection}} + E_{\text{Random Features}} + E_{\text{Optimization}}$$

As we will see, this decomposition reveals some interesting directions of research, as we do not know yet bounds for all of these terms in the case of quadratic regularization. For the moment, the important thing to notice in this decomposition is that once we show that the regularization error goes to zero as ε goes to zero, all other terms also go to zero when the parameters k, m, B are taken to infinity (since the terms become identical and cancel each other).

3.2 Bounds on the regularization error

The first source of error comes from the regularization of Kantorovich's problem (KP), E_{Reg}. Not only we should show that it goes to zero as ε goes to zero, but it is very important to understand the rate of this convergence. The following proposition was first proved in [12], and it shows the convergence rates for the case of entropic regularization.

Proposition 3.2.1. *Let α and β be probability measures on $\mathcal{X}$ and $\mathcal{Y}$ subsets of $\mathbb{R}^d$ such that $|\mathcal{X}| = |\mathcal{Y}| \leqslant D$ and assume that c is L-Lipschitz w.r.t. x and y. It holds*

$$\begin{aligned} 0 \leqslant \mathcal{W}_\varepsilon^{Ent}(\alpha, \beta) - \mathcal{W}(\alpha, \beta) &\leqslant 2\varepsilon d \log\left(\frac{e^2 \cdot L \cdot D}{\sqrt{d} \cdot \varepsilon}\right) \\ &\sim_{\varepsilon \to 0} 2\varepsilon d \log(1/\varepsilon) \end{aligned}$$

With some modification of the proof, we showed in this intership the analogous proposition to the quadratic regularization, which is of interest to us. Although this has the same structure as the previous proposition, we state it as a theorem to emphasize its importance.

Theorem 3.2.2. *Let α and β be probability measures on $\mathcal{X}$ and $\mathcal{Y}$ subsets of $\mathbb{R}^d$ such that $|\mathcal{X}| = |\mathcal{Y}| \leqslant D$ and assume that c is L-Lipschitz w.r.t. x and y. It holds*

$$0 \leqslant \mathcal{W}_\varepsilon^Q(\alpha,\beta) - \mathcal{W}(\alpha,\beta) \lesssim \varepsilon^{\frac{1}{d+1}}$$

Proof. Before starting, we establish some notation useful for the proof. For a probability measure π on $\mathcal{X}\times\mathcal{Y}$, define $C(\pi) \overset{\text{def.}}{=} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)\,d\pi(x,y)$, i.e., the associated cost of transportation along π. Let π_0 be a minimizer of $\min_{\pi\in\Pi(\alpha,\beta)} C(\pi)$. We denote by $Q(\pi)$ the quadratic regularization term, and so the quadratically regularized problem can be cast as $\min_{\pi\in\Pi(\alpha,\beta)} C(\pi) + \varepsilon Q(\pi)$.

Block approximation of the problem. We want to define a block approximation of the problem, with which it will be easier to get the desired bounds.

Definition 3.2.1. For a resolution $\Delta > 0$, we consider the block partition of $\mathbb{R}^d$ in hypercubes of side Δ, defined as

$$\{Q_k^\Delta \overset{\text{def.}}{=} [k_1\cdot\Delta,\,(k_1+1)\cdot\Delta[\times\cdots\times[k_d\cdot\Delta,\,(k_d+1)\cdot\Delta[\,;\,k=(k_1,\dots,k_d)\in\mathbb{Z}^d\}$$

The block approximation of π_0 of resolution Δ is the measure $\pi^\Delta\in\Pi(\alpha,\beta)$ characterized by

$$\pi^\Delta|_{Q_{ij}} = \frac{\pi_0\left(Q_{ij}^\Delta\right)}{\alpha_i^\Delta\cdot\beta_j^\Delta}\left(\alpha|_{Q_i^\Delta}\otimes\beta|_{Q_j^\Delta}\right)$$

Intuitively, this tries to approximate π_0 through the product measure $\alpha\otimes\beta$. To be clear, we apply the convention $0/0 = 0$ in this definition. We must check the consistency of this definition, i.e., that $\pi^\Delta\in\Pi(\alpha,\beta)$ really holds. It is clear that it is positive and that it sums up to 1. We must thereefore verify that its marginals are α and β. If $B\subset\mathbb{R}^d$ is any Borel subset, then

$$\begin{aligned}
\pi^\Delta(B\times) &= \sum_{i,j\in\mathbb{Z}^2}\frac{\pi_0(Q_{ij})}{\alpha_i\cdot\beta_j}\alpha(B\cap Q_i)\,\beta(Q_j)\\
&= \sum_{i,j\in\mathbb{Z}^2}\frac{\pi_0(Q_{ij})}{\alpha_i}\alpha(B\cap Q_i)\\
&= \sum_{i\in\mathbb{Z}}\frac{\alpha(B\cap Q_i)}{\alpha_i}\underbrace{\sum_{j\in\mathbb{Z}}\pi_0(Q_{ij})}_{=\alpha(Q_i)\text{ since }\pi_0\in\Pi(\alpha,\beta)}\\
&= \sum_{i\in\mathbb{Z}}\alpha(B\cap Q_i) = \alpha(B)
\end{aligned}$$

An analogous calculation shows that $\pi^\Delta(\mathbb{R}^d\times B) = \beta(B)$. Therefore we conclude that $\pi^\Delta\in\Pi(\alpha,\beta)$ and thus the consistency of this definition.

Rudimentary bound. The fact that $0\leqslant\mathcal{W}_\varepsilon^Q(\alpha,\beta) - \mathcal{W}(\alpha,\beta)$ is trivial, since $Q(\pi)\geqslant 0$. It is

also trivial to say that $\mathcal{W}_\varepsilon(\alpha,\beta) \leqslant C(\pi^\Delta) + \varepsilon Q(\pi^\Delta)$. Thus if we identify $\mathcal{W}(\alpha,\beta) = C(\pi_0)$, we have

$$0 \leqslant \mathcal{W}_\varepsilon^Q(\alpha,\beta) - \mathcal{W}(\alpha,\beta) \leqslant \left(C(\pi^\Delta) - C(\pi_0)\right) + \varepsilon Q(\pi^\Delta)$$

This is what we will refer to as our "rudimentary bound". The final bound will be achieved by first bounding $\left(C(\pi^\Delta) - C(\pi_0)\right)$, and then $Q(\pi^\Delta)$.

Bounding the first term. The bound on the term with the costs comes from the Lipschitz regularity of the ground cost c :

$$\begin{aligned}
C(\pi^\Delta) - C(\pi_0) &= \int_{\mathcal{X}\times\mathcal{Y}} c\, d\pi^\Delta - \int_{\mathcal{X}\times\mathcal{Y}} c\, d\pi_0 \\
&= \sum_{i,j\in\mathbb{Z}^2} \left(\int_{Q_{ij}} c\, d\pi^\Delta - \int_{Q_{ij}} c\, d\pi_0 \right) \\
&= \sum_{i,j\in\mathbb{Z}^2} \left(\int_{Q_{ij}} c \frac{\pi_0(Q_{ij})}{\alpha_i\beta_j}\, d\alpha|_{Q_i} d\beta|_{Q_j} - \int_{Q_{ij}} c\, d\pi_0 \right) \\
&\leqslant \sum_{i,j\in\mathbb{Z}^2} \left(\int_{Q_{ij}} c \frac{\pi_0(Q_{ij})}{\alpha_i\beta_j}\, d\alpha|_{Q_i} d\beta|_{Q_j} - \pi_0(Q_{ij}) \inf_{Q_{ij}} c \right) \\
&\leqslant \sum_{i,j\in\mathbb{Z}^2} \pi_0(Q_{ij}) \left(\sup_{Q_{ij}} c - \inf_{Q_{ij}} c \right)
\end{aligned}$$

Remember that the diameter of each set Q_i^Δ is $\Delta\sqrt{d}$. Using the L-Lipschitz condition of c we have

$$\left\| c(x,y) - c(\tilde{x},\tilde{y}) \right\|_2 \leqslant \left\| (x-\tilde{x}, y-\tilde{y}) \right\|_2$$

If $(x,y), (\tilde{x},\tilde{y}) \in Q_{ij}^\Delta$, then $\|x-\tilde{x}\|_2 \leqslant \Delta\sqrt{d}$ and $\left\|y-\tilde{y}\right\|_2 \leqslant \Delta\sqrt{d}$. This way,

$$\left\| (x-\tilde{x}, y-\tilde{y}) \right\|_2 = \left((x-\tilde{x})^2 + \left(y-\tilde{y}\right)^2 \right)^{\frac{1}{2}} \leqslant \left(2\Delta^2 d\right)^{\frac{1}{2}}$$

That is, we have $\left(C(\pi^\Delta) - C(\pi_0)\right) \leqslant 2L\sqrt{d}\Delta$.

Bounding the second term. To bound the quadratic regularization term, we write it in the block approximation :

$$\begin{aligned}
Q(\pi^\Delta) &= \sum_{i,j\in\mathbb{Z}^2} \frac{1}{2} \int_{Q_{ij}} \left(\frac{\pi_0(Q_{ij})}{\alpha_i^\Delta \beta_j^\Delta} \right) d\alpha|_{Q_i} d\beta|_{Q_j} \\
&= \frac{1}{2} \sum_{i,j\in\mathbb{Z}^2} \frac{\pi_0(Q_{ij})^2}{\alpha_i^\Delta \beta_j^\Delta} \\
&= \frac{1}{2} \sum_{i,j\in\mathbb{Z}^2} \left(\frac{\pi_0(Q_{ij})}{\alpha_i^\Delta} \right) \left(\frac{\pi_0(Q_{ij})}{\beta_j^\Delta} \right) \\
&\overset{\text{def.}}{=} \frac{1}{2} \sum_{i,j\in\mathbb{Z}^2} A_{ij} B_{ij}
\end{aligned}$$

Given $A = \left(A_{ij}\right)_{ij\in\mathbb{Z}^2}$ and $B = \left(B_{ij}\right)_{ij\in\mathbb{Z}^2}$ in $\mathbb{R}_+^{\mathbb{Z}\times\mathbb{Z}}$, the function $\phi\left(A,B\right) \overset{\text{def.}}{=} \sum_{i,j\in\mathbb{Z}^2} A_{ij}B_{ij}$ can be easily verified to be bilinear, symmetric, and positive definite. Therefore it defines an inner product. Consequently, we can apply the associated Cauchy-Schwarz inequality :

$$\langle A,B\rangle_\phi \leqslant \langle A,A\rangle_\phi^{\frac{1}{2}} \langle B,B\rangle_\phi^{\frac{1}{2}}$$

Applying this back to our case, we have

$$Q\left(\pi^\Delta\right) \leqslant \frac{1}{2}\left[\left(\sum_{i,j\in\mathbb{Z}^2}\left(\frac{\pi_0\left(Q_{ij}\right)}{\alpha_i^\Delta}\right)^2\right)^{\frac{1}{2}}\left(\sum_{i,j\in\mathbb{Z}^2}\left(\frac{\pi_0\left(Q_{ij}\right)}{\beta_j^\Delta}\right)^2\right)^{\frac{1}{2}}\right]$$

However, $\sum_j \pi_0\left(Q_{ij}\right)^2 \leqslant \left(\sum_j \pi_0\left(Q_{ij}\right)\right)^2 = \left(\alpha_i^\Delta\right)^2$, and therefore

$$\begin{aligned} Q\left(\pi^\Delta\right) &\leqslant \frac{1}{2}\left[\left(\sum_{Q_i\cap\mathcal{X}\neq\emptyset} 1\right)^{\frac{1}{2}}\left(\sum_{Q_j\cap\mathcal{Y}\neq\emptyset} 1\right)^{\frac{1}{2}}\right] \\ &\leqslant \frac{1}{2}\frac{D}{\Delta^d} \end{aligned}$$

Thus concluding the bound on the second term.

Putting it all together. If we write the bounds on the two terms of our rudimentary bound, we have

$$\left(C\left(\pi^\Delta\right) - C\left(\pi_0\right)\right) + \varepsilon Q\left(\pi^\Delta\right) \leqslant 2L\Delta\sqrt{d} + \varepsilon\frac{D}{2\Delta^d}$$

This is valid for every resolution Δ we pick, but to rid ourselves from this parameter we optimize over it to get the best bond in this family. By simply deriving with respect to Δ and equating the result to 0, we obtain the best resolution as being

$$\Delta = \left(\frac{\varepsilon D\sqrt{d}}{4L}\right)^{\frac{1}{d+1}}$$

Substituting it back in the rudimentary bound we get

$$\left(C\left(\pi^\Delta\right) - C\left(\pi_0\right)\right) + \varepsilon Q\left(\pi^\Delta\right) \leqslant 2L\sqrt{d}\left(\frac{\varepsilon D\sqrt{d}}{4L}\right)^{\frac{1}{d+1}} + \frac{\varepsilon D}{2}\left(\frac{\varepsilon D\sqrt{d}}{4L}\right)^{\frac{-d}{d+1}}$$

Simplifying and observing the behaviour only with respect to the regularization parameter, we get the final bound

$$0 \leqslant \mathcal{W}_\varepsilon^Q\left(\alpha,\beta\right) - \mathcal{W}\left(\alpha,\beta\right) \lesssim \varepsilon^{\frac{1}{d+1}}$$

□

3.3 A brief discussion on the other error terms

Having analyzed the behaviour of E_{Reg} in the previous paragraph, we are left with the other terms E_{Sample}, E_{RF}, $E_{\text{Projection}}$. As stated before, E_{Sample} is the error of SGD, already developed

in this report. The error associated with the fact that we project the coefficients at each iteration is difficult to analyze, and not known when the solution is outside the projection ball. Notice however that the problem in both regularizations is strongly convex, and thus has unique solution. If by the end of the iterations the coefficients found satisfy the optimality conditions and do not saturate the projection constraint, then we know that the solution is inside the ball and there is no error associated with B. Finally, in [11] they provide a bound of $L\log(m)m^{-\frac{1}{d}}$ on the error E_{RF} of approximating any L-Lipschitz function g by using only m random features

Chapter 4

Numerical Results

4.1 First test : gaussian to itself

This first test is discussed in great detail, as the ideas explained for it are also valid for subsequent tests.

Choice of distributions. The first test we are going to perform is one the simplest possible. We take both the source and target distributions to be the centered gaussian $\mathcal{N}(0, 0.04)$ (Figure 4.1). This might seem trivial if we think of it in terms of Monge's formulation of optimal transport : one would be correct to think that the solution would be not to transport anything. However, remember that we are in the regularized case, and it is a feature of such a setting that $\mathcal{W}_\varepsilon(\alpha, \alpha) \neq 0$. That is, the solution potentials are not identically zero, and so this is an interesting case afterall. This might seem odd, but it is in fact a motivation for the definition of Sinkhorn divergences [13].

Exact solution. Another reason why this case is interesting is that since we are transporting a simple gaussian to another gaussian, we can follow [14] to calculate the exact solutions u^* and v^*, therefore being able to exactly calculate the error $\left\|u^* - u^k\right\|_2$ (same for v) performed by the algorithm. The exact solution is the quadratic form $u^*(x) = v^*(x) = -\frac{\varepsilon}{2}(0.00089)\,x^2$, therefore near the origin and in the scale that we will be looking at it it will certainly look as being identically zero.

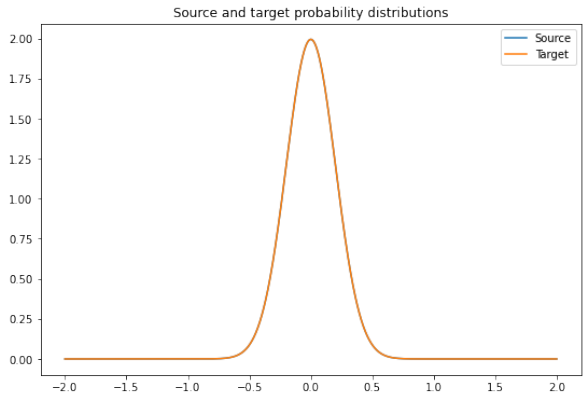

Figure 4.1: Source and target

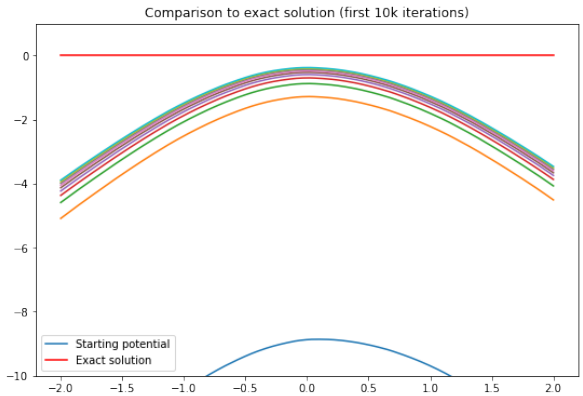

Figure 4.2: Evolution of u (10^4 iterations)

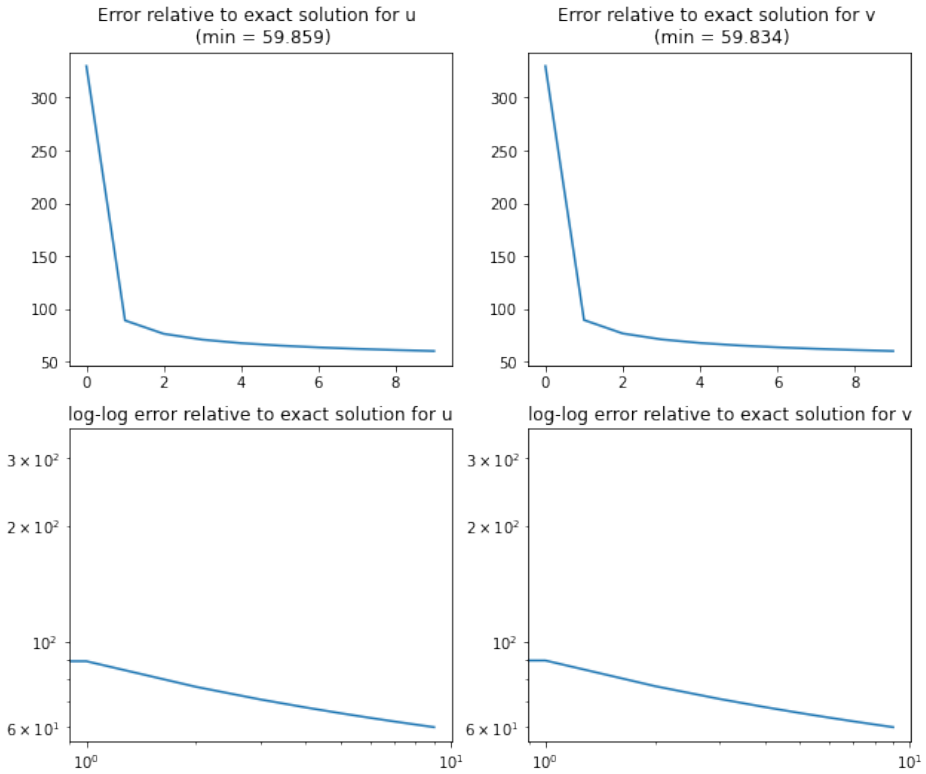

Figure 4.3: Error decay in the first 10^4 iterations (x axis in 10^3 units)

4.1.1 Entropic regularization

Parameters. We fix the regularization parameter to be $\varepsilon = 0.1$. We will expand the potentials over $m = 10^3$ ReLU random features. At each step we project the iterates to the ball of radius $B = 10^3$ (this should be as big as possible, increasing the probability of this ball containing the solution). We will execute the algorithm multiple times, each time allowing it to perform 10^4 iterations before analyzing the behaviour of the coefficients found and relaunching it. At the very first execution we take steps $c_u = c_v = 5$ and initial coefficients $\mathbf{w}_u^{(0)} = \mathbf{w}_v^{(0)} = -1$.

Results after 10^4 iterations. The first 10^4 took 81s to run. We observe visually that the distant initial potential seem to go in the right direction and tries to approach the exact solution (Figure 4.2). More precisely, we can observe the ℓ_2 error decay (Figure 4.3). We observe not only the decay of the error, but also by looking at the log-log plot we observe that its rate of decay is very close to $1/\sqrt{t}$, as predicted by Theorem 1.3.1. Finally, another indication that the algorithm is working correctly lies in the fact that we can observe the estimated objective function along the iterations, and we verify that the objective function is indeed growing (Figure 4.4). We do observe the phenomenon of slowing convergence due to the diminishing steps γ_t. Therefore, we will relaunch the algorithm from where we stopped for another 10^4 iterations (and so on, until we reach a good solution).

Results after 6×10^4 iterations. We run the algorithm five more times to obtain the following results. It took in total about 480s (8 minutes) of execution. At all times the objective function kept growing and the error kept falling. The final potentials are shown in Figure 4.5. The evolution of the error in these last 10^4 iterations is shown in Figure 4.6. In particular

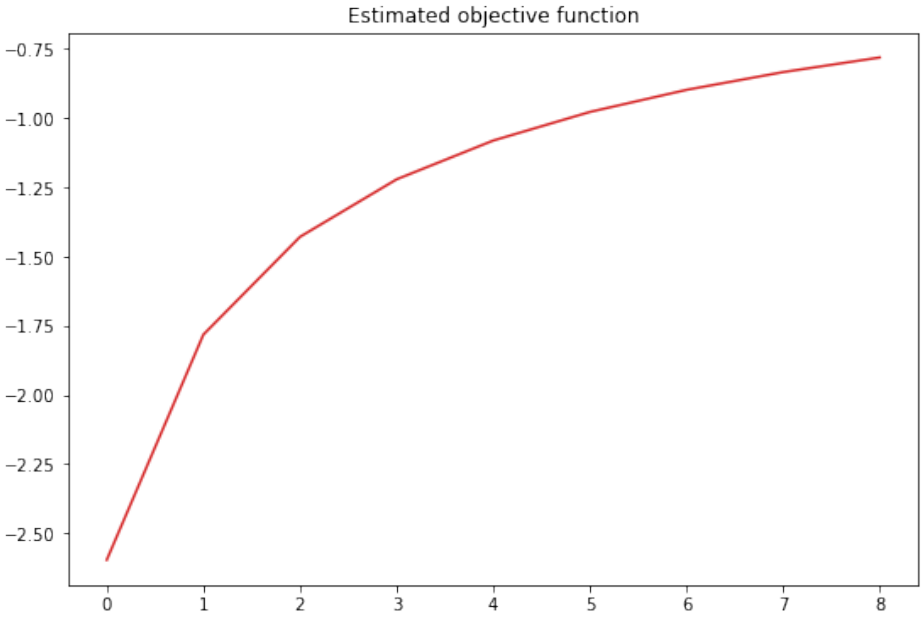

Figure 4.4: Growth of the estimated objective function on the first 10^4 iterations

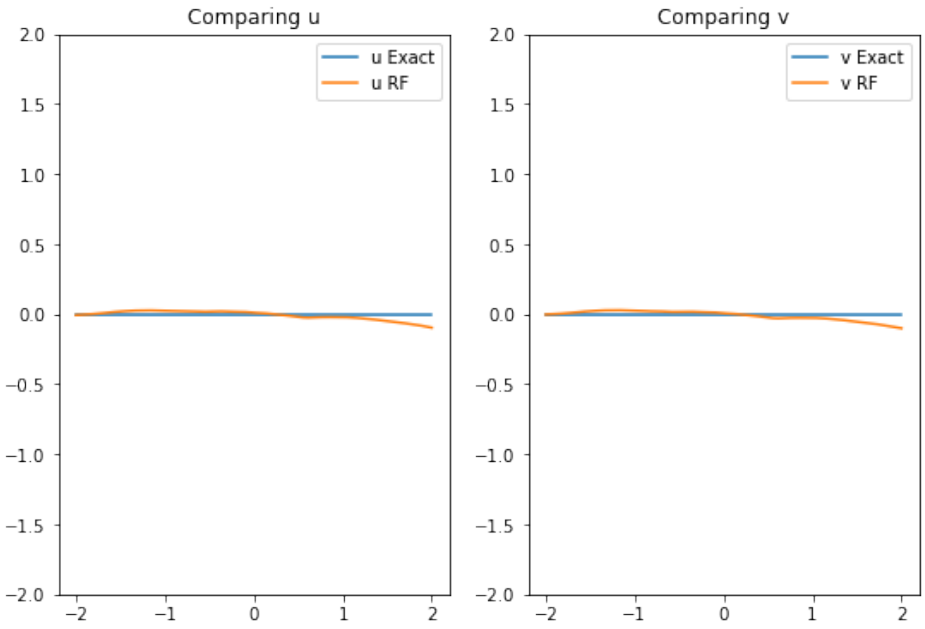

Figure 4.5: Final potentials after 6×10^4 iterations

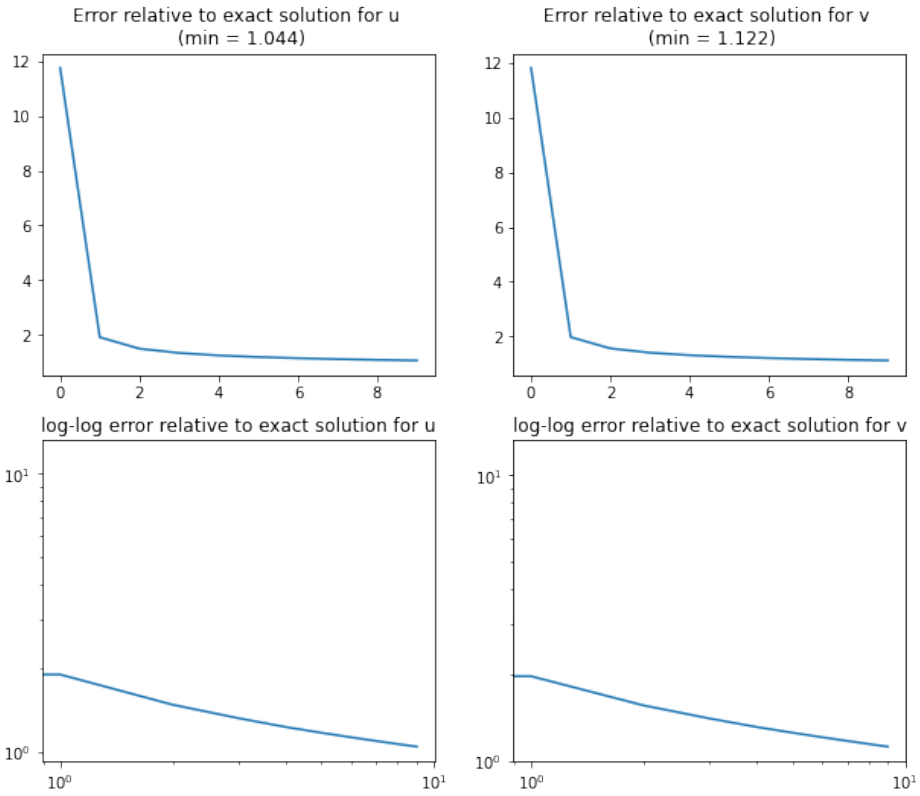

Figure 4.6: Evolution of error in the last 10^4 iterations (x axis in 10^3 units)

notice how in the span of these iterations the error went from over 300 to 1.

Complications of entropic regularization. The results above indicate that the algorithm works (at least in this case) as intended. However, there is a caveat. If the initial step-size at some execution happens to be too big, after a few iterations you may fall into a zone of instability, a "hole" in the objective function in which the gradient is enormous, and as a consequence the step performed will "throw" you far away from the solution. Figure 4.7 exemplifies this phenomenon. There is a way to see this directly in the objective function. Remember that our problem reads is to maximize

$$\mathbf{w}_u^\top \Phi(x) + \mathbf{w}_v^\top \Phi(y) - \varepsilon \exp\left(\frac{\mathbf{w}_u^\top \Phi(x) + \mathbf{w}_v^\top \Phi(y) - c(x,y)}{\varepsilon}\right)$$

What we are going to do is to pick any two coordinates, fix the others, and visualize the graph of the corresponding two-dimensional section. Suppose we take a coordinate $(\mathbf{w}_u)_i$ and $(\mathbf{w}_v)_j$, fixing the others. Then, by writing the previous expression putting these terms on evidence and simplifying the notation by changing $(\mathbf{w}_u)_i \to \mu$, $(\mathbf{w}_v)_j \to \nu$, $(\Phi(x))_i \to a$, and $\big(\Phi(y)\big)_j \to b$ we can look at the graph of

$$a\mu + b\nu + k - \varepsilon \exp\left(\frac{a\mu + b\nu + k - c}{\varepsilon}\right)$$

We will substitute a (respectively b) by the mean value of a component of $\Phi(x)$ (respectively $\Phi(y)$) which are both 0.01. We will also substitute c by its mean value, which is the variance

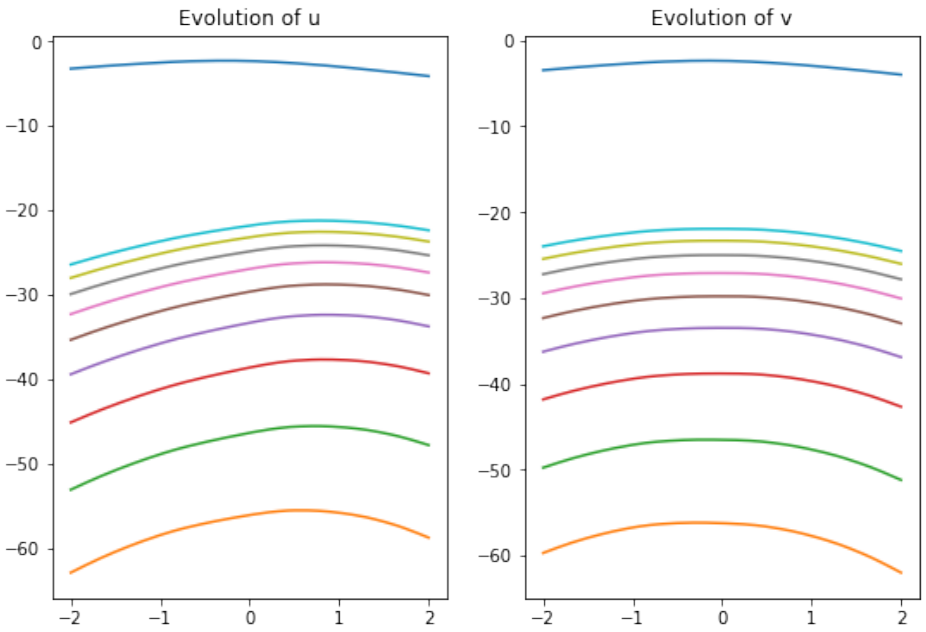

Figure 4.7: Instability from too big step-sizes (initial potential in blue, second potential in orange)

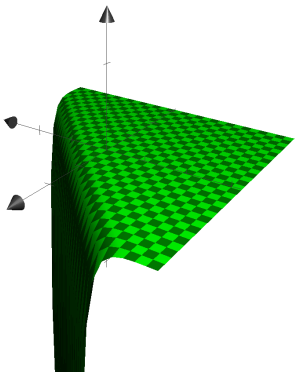

Figure 4.8: Projection (entropic)

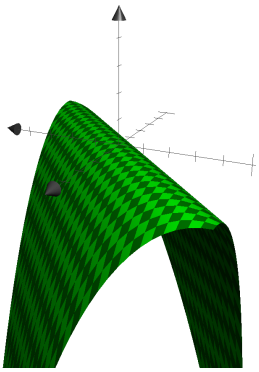

Figure 4.9: Projection (quadratic)

of the gaussian, i.e., $c \to 0.04$. Finally, k is left as a fixed parameter, but it is easy to see that it does not matter too much to the fundamental behaviour we will observe. The result is shown in Figure 4.8 for the case of entropic regularization. It reveals that there are two critical regions on the objective function we seek to maximize : a "hole" where the gradient has great norm, and an almost flat region. Once in the whole, the gradient is very big and the next step will "eject" you from it, but then you will fall in the flat region. We can verify numerically that indeed the gradient becomes practically constant. Its small norm implies slow convergence. If one changes the entropic regularization by the quadratic one, one obtains the graph in Figure 4.9. This immediately shows many advantages in favor of quadratic regularization, namely

1. No flat regions, implying lesser risk of slow convergence.
2. Polynomial variation instead of exponential one, implying no holes, that is, we do not observe sudden explosions of gradient. In particular, remember that in Theorem 1.3.1

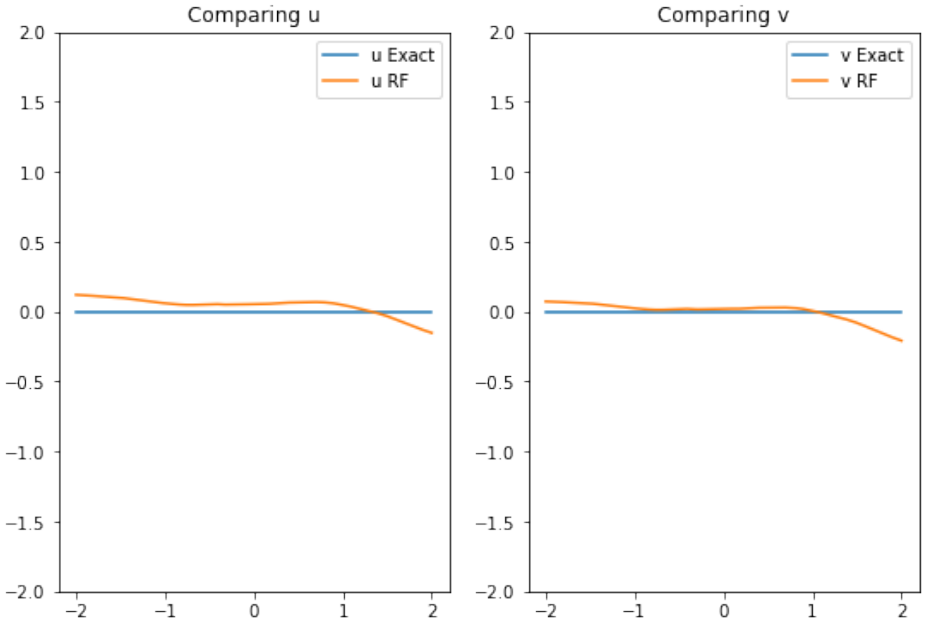

Figure 4.10: Final potentials for quadratic regularization

the bound is proportional to the Lipschitz constant of the objective. The fact that the quadratic regularization objective explodes slower implies that (after projecting to a ball of radius B) the convergence bound of SGD should be tighter, i.e., we should have faster convergence.

These figures also make it clear the need to project the algorithm, since we see that the objective function has unbounded derivatives in the whole space.

4.1.2 Quadratic regularization

Parameters. Changing the regularization from entropic to quadratic implies that for a fixed ε the solution potentials are not the same. However, using the bounds in Proposition 3.2.1 and Theorem 3.2.2 we calculate that to have an error equivalent to the error associated with $\varepsilon = 0.1$ in entropic regularization, one must use $\varepsilon = 0.05$ in quadratic regularization, and so we set the value of the regularization parameter. We continue to use $m = 10^3$ random features, and a projection parameter $B = 10^3$. We start from the same initial coefficients as before, and perform 10^4 iterations with initial step-size $c_u = c_v = 15$.

Results after 6×10^4 iterations. To compare with previous results, we perform the same number of total iterations. It took about the same time (480s on average) to run these iterations. The final results are displayed in Figure 4.10, while the error is displayed in Figure 4.11. We see that the error stops decreasing at around the same value as with entropic regularization. This is probably due to the fact that we are in fact approaching the exact solution to the entropic problem, not the quadratic one. The main feature however is the observed lack of explosions while performing these simulations.

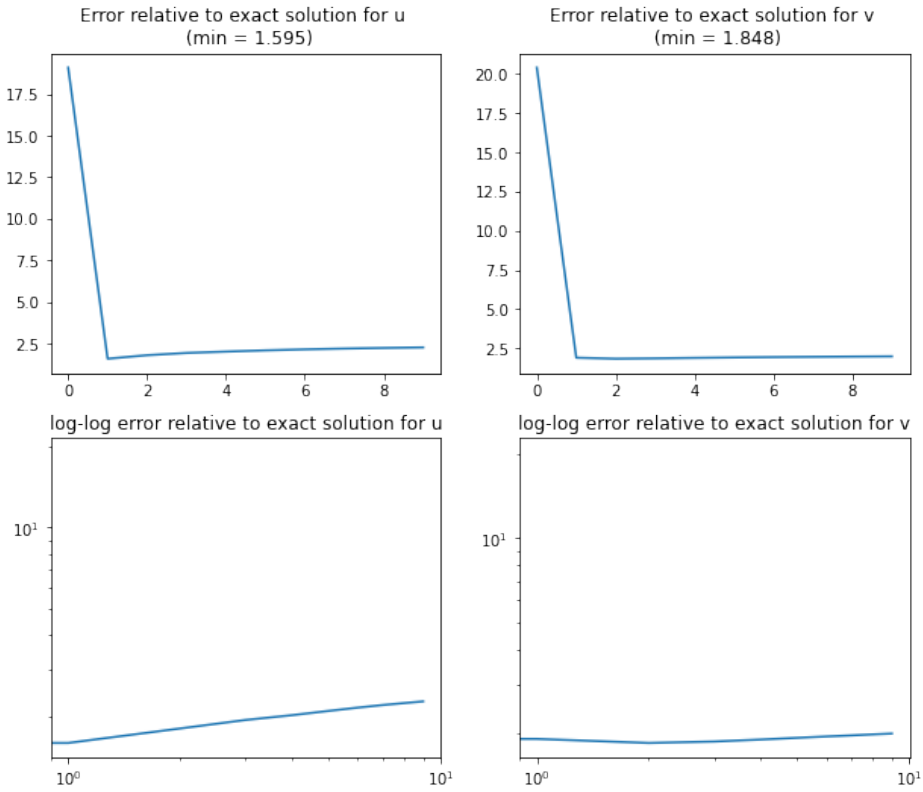

Figure 4.11: Error evolution in the last 10^4 iterations

4.2 Second test : single gaussian to gaussian mixture

Choice of distributions. Now we pick a more interesting test, appearing in [1]. We transport the same source gaussian as before, but now the target distribution is the normalized sum of three gaussians $\mathcal{N}(-1, 0.2)$, $\mathcal{N}(0, 3)$, and $\mathcal{N}(1, 0.3)$. These distributions are pictured in Figure 4.12. Already in this relatively simple case, we do not know explicit exact solutions. We will thus verify our error by calculating whether or not the optimality conditions for the problem are satisfied (but only roughly, since they are are very sensitive, this will be explained below).

4.2.1 Entropic regularization

Parameters. As before, we take $\varepsilon = 10^{-1}$, and start from the same initial position. We don't start at zero potentials because, as before, it is close to a critical region of the objective function and the potentials frequently explode unless we use nearly negligible step-sizes. We keep $B = 10^3$ as before. We start with $c_u = c_v = 1$ and perform 10^4 iterations before verifying if things are going fine.

Results after 10^4 iterations. After the first 10^4 iterations, which took 85s to run, we observe in Figure 4.13 that the estimated objective function indeed grows, and in Figure 4.14 we see the stable evolution of the potentials.

Results after 6×10^4 iterations. After 6×10^4 iterations the potentials and the objective function stop evolving, as exhibited in Figure 4.15 and 4.16.

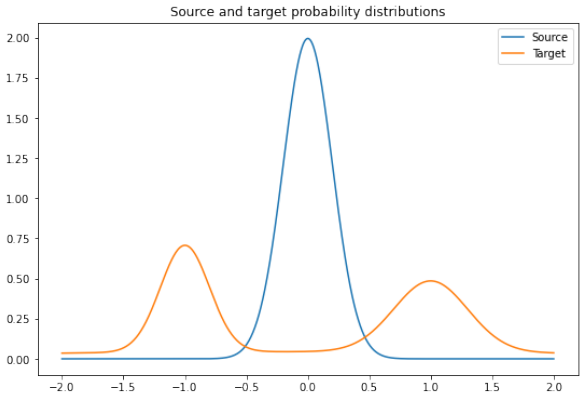

Figure 4.12: Source and target

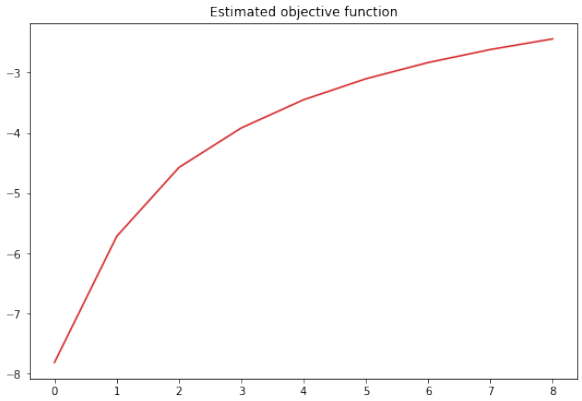

Figure 4.13: Growth of the estimated objective function on the first 10^4 iterations

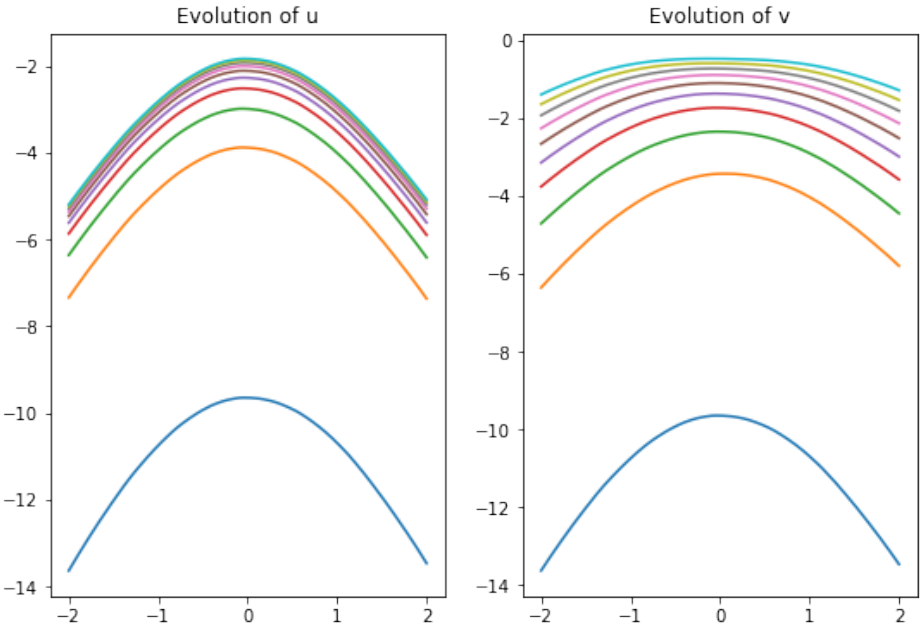

Figure 4.14: Potentials after 10^4 iterations

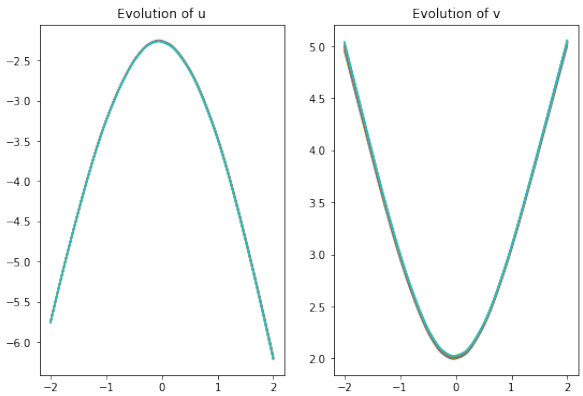

Figure 4.15: Stagnation of potentials

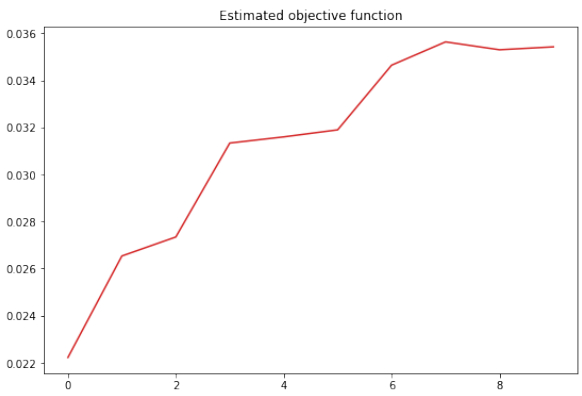

Figure 4.16: Stagnation of objective function

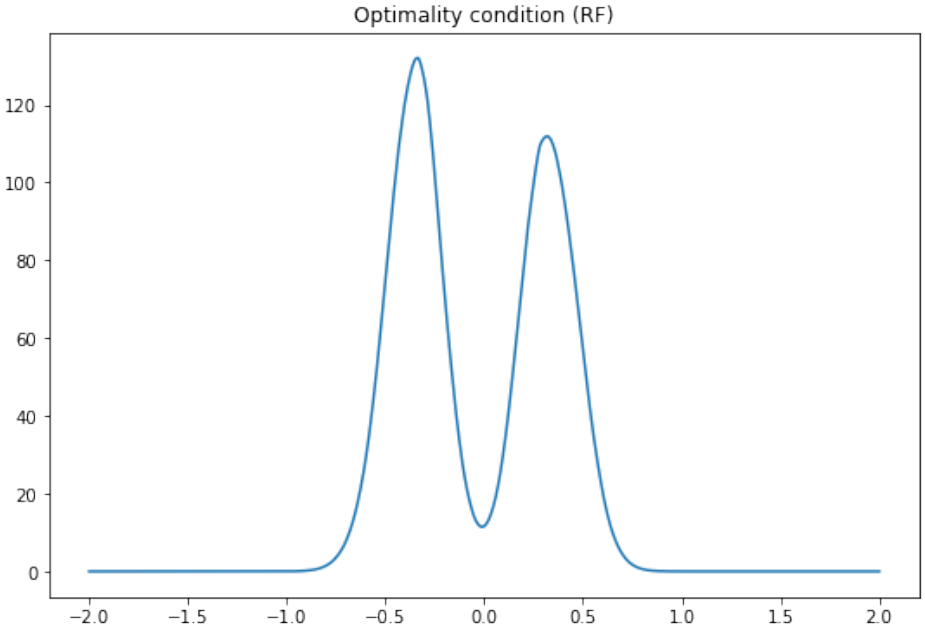

Figure 4.17: Entropic optimality conditions

A comment on the optimality conditions. We can calculate the optimality conditions to verify whether we arrived at the extremal point of the problem, see Figure 4.17. One might see this figure and regard the optimality conditions as being violated near the origin. However, observe that the equations are

$$\left[\sum_j \beta_j \exp\left(\left(\frac{u_i + v_j - C_{ij}}{\varepsilon}\right) - 1\right)\right] - 1 = 0$$
$$\left[\sum_i \alpha_i \exp\left(\left(\frac{u_i + v_j - C_{ij}}{\varepsilon}\right) - 1\right)\right] - 1 = 0$$

and thus we have an exponential term with small denominator. This means that these equations are in fact very sensitive: for example, if we perturb u by adding 0.2 to it, the violation of the optimality conditions increases tenfold. Thus the violation observed near the origin should not mean that we are too far from the solution.

4.2.2 Quadratic regularization

Finally, we perform the same experiment using quadratic regularization. We did not observe any explosive behavior, independently of the step-sizes or initial positions taken.

Parameters. We keep all parameters the same except for the regularization parameter, which we take to be $\varepsilon = 5 \times 10^{-2}$ for quadratic regularization, and start at the same initial position.

Results after 10^4 iterations. After these first iterations, which take on average 90s to run, we observe the potentials evolving as in Figure 4.19 and the growth of the objective function in

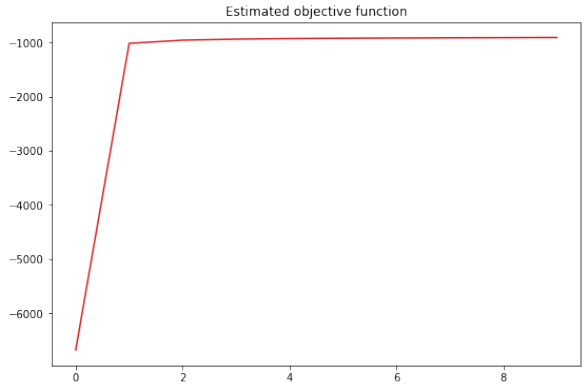

Figure 4.18: Growth of the estimated objective function on the first 10^4 iterations

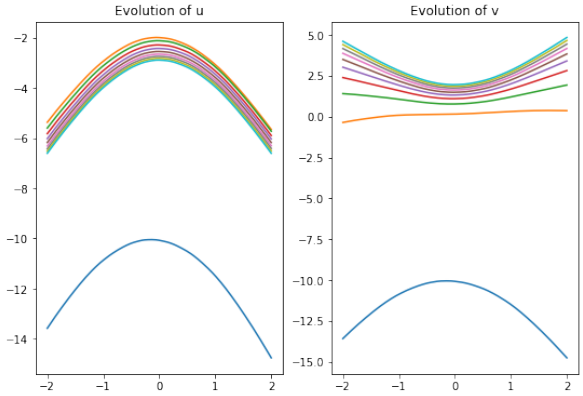

Figure 4.19: Potentials after 10^4 iterations

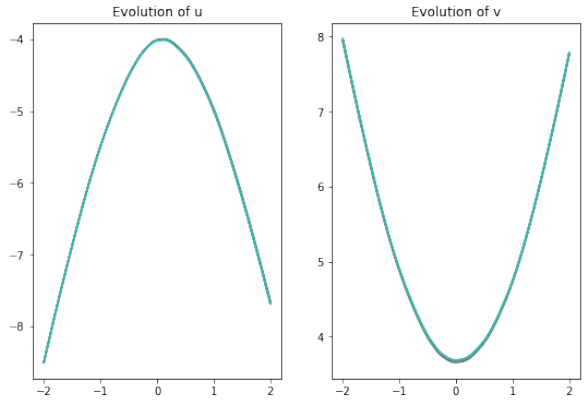

Figure 4.20: Final potentials

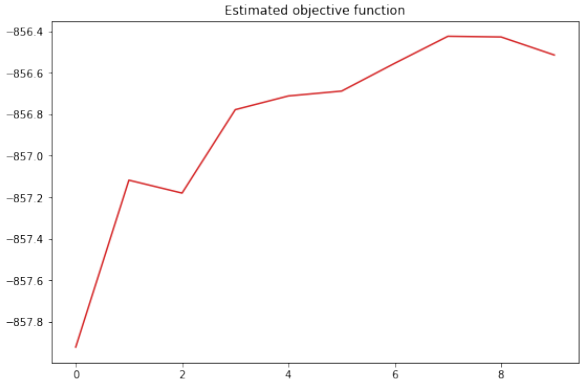

Figure 4.21: Stagnation of the objective function

Figure 4.18

Results after 6×10^4 iterations. As Figure 4.18 already suggests, we see the potentials already stabilizing after around 2×10^4 iterations, which is much faster than for the entropic case. But to observe better stability on the growth of the objective function (Figure 4.21), and for the sake of equality between experiments, we perform up to 6×10^4 iterations. The final potentials are observed in Figure 4.20.

Another comment on the optimality conditions. Finally, we observe as before the optimality conditions in Figure 4.22. There is some violation, but as before, the equations for it are

$$\sum_j \left(\frac{u_i + v_j - C_{ij}}{\varepsilon} \right) \beta_j - 1 = 0$$

$$\sum_i \left(\frac{u_i + v_j - C_{ij}}{\varepsilon} \right) \alpha_i - 1 = 0$$

Although the dependence on $1/\varepsilon$ is not exponential but rather linear, ε is half as small in the quadratic case.

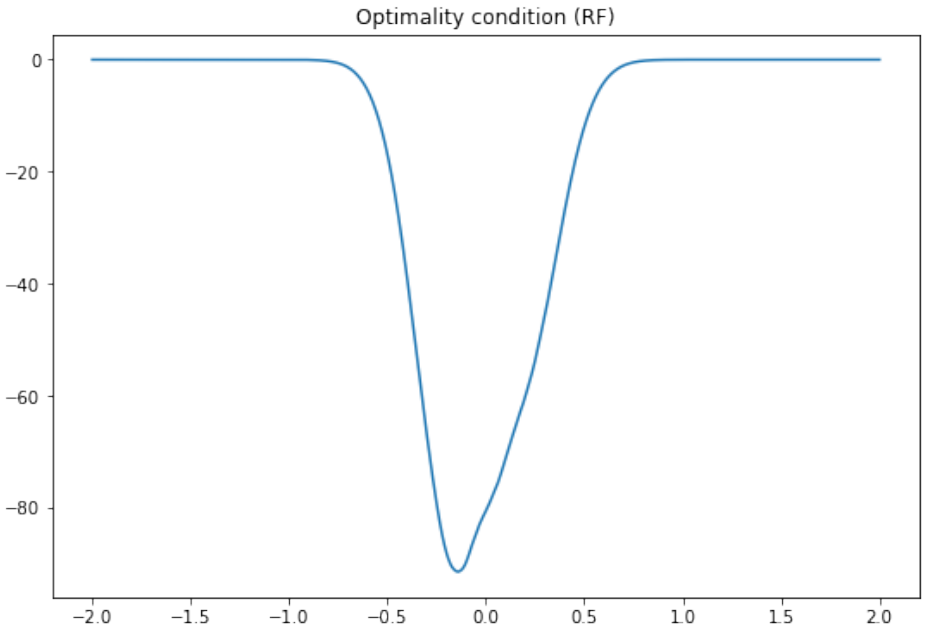

Figure 4.22: Quadratic optimality conditions

Conclusion

Recapitulating. This report presented the main theory developed during this internship, as well as the main numerical results obtained as a consequence of it. More explicitly, we developed a bound on the regularization error for quadratic regularization, explained conceptually why it should behave better than the entropic one in view of its smaller Lipschitz constant and absence of flat regions, and performed numerical simulations that illustrate these points.

Directions of research. There are some natural directions of research from now on. One is to try to give bounds on the error associated with the projection π_B performed at each step of our algorithm. Another one would be to generalize even further the theory and provide the same developments for L^p regularizations, or even for general convex regularizations. One could also continue to explore the advantages of quadratic regularization in the context of practical applications of optimal transport, particularly in machine learning.

Bibliography

[1] Genevay Aude et al. *Stochastic Optimization for Large-scale Optimal Transport*. 2016. arXiv: 1605.08527 [math.OC].

[2] Filippo Santambrogio. "Optimal Transport for Applied Mathematicians. Calculus of Variations, PDEs and Modeling". In: (2015). URL: https://www.math.u-psud.fr/~filippo/OTAM-cvgmt.pdf.

[3] C. Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008. ISBN: 9783540710509. URL: https://books.google.fr/books?id=hV8o5R7%5C_5tkC.

[4] Marco Cuturi. *Sinkhorn Distances: Lightspeed Computation of Optimal Transportation Distances*. 2013. arXiv: 1306.0895 [stat.ML].

[5] Marco Cuturi Gabriel Peyré. *Computational Optimal Transport*. International series of monographs on physics. Clarendon Press, 1981. ISBN: 9780198520115.

[6] Arthur Gretton Dino Sejdinovic. *What is a RKHS ?* 2012. URL: https://www.gatsby.ucl.ac.uk/~gretton/coursefiles/RKHS_Notes1.pdf.

[7] Francis Bach. *Learning theory from first principles*. 2021. URL: https://www.di.ens.fr/~fbach/ltfp_book.pdf.

[8] Aude Genevay. "Entropy-Regularized Optimal Transport for Machine Learning". Theses. PSL University, Mar. 2019. URL: https://tel.archives-ouvertes.fr/tel-02319318.

[9] Aymeric Dieuleveut. "Stochastic approximation in Hilbert spaces". 2017PSLEE059. PhD thesis. 2017. URL: http://www.theses.fr/2017PSLEE059/document.

[10] Ali Rahimi and Benjamin Recht. "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems*. Ed. by J. Platt et al. Vol. 20. Curran Associates, Inc., 2008. URL: https://proceedings.neurips.cc/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

[11] Francis Bach. "Breaking the Curse of Dimensionality with Convex Neural Networks". In: *Journal of Machine Learning Research* 18.19 (2017), pp. 1–53. URL: http://jmlr.org/papers/v18/14-546.html.

[12] Aude Genevay et al. *Sample Complexity of Sinkhorn divergences*. 2019. arXiv: 1810.02733 [math.ST].

[13] Jean Feydy et al. *Interpolating between Optimal Transport and MMD using Sinkhorn Divergences*. 2018. arXiv: 1810.08278 [math.ST].

[14] Hicham Janati et al. *Entropic Optimal Transport between Unbalanced Gaussian Measures has a Closed Form*. 2020. arXiv: 2006.02572 [math.ST].