Project report
SISTA: learning optimal transport costs under sparsity constraints

Florian Le Bronnec

December 2021

Abstract

Optimal transport often amounts at finding the best transport map between points. Best in the sense that it minimizes a given cost. In this project, we are going to study the inverse problem. That is, we observe a mapping and we want to determine what is the underlying cost. This kind of problem arises a soon as one observes matched entities and want to know what makes them match this way. Such a situation is really common in economics or sociology, where a lot of studies are first based on rigorous observations.

However, this problem has only been a little studied, compared to other topics in optimal transport. To address this problem, [7], whose authors were dealing with a topic of assignment problem, proposed a parametric model and a formulation that is equivalent to a convex minimization problem.

In this project, we are going adopt this formulation and study a new algorithm, SISTA, proposed by [6] that solves this problem by imposing sparsity constraints on the parameters with a l_1 penalty term. This algorithm alternates exact minimization steps (corresponding to Sinkhorn's iterations) with a proximal gradient descent step. The authors of [6] also provide theoretical guarantees on the convergence of this algorithm. The goals of this project are to present the main theoretical results, provide additional details on the some aspects of the original article and conduct numerical experiments to explore properties of the new algorithm. Finally, we will run it on a real world example.

Contents

1 Introduction

In this project we are going to study the optimal transport problem in the discrete case. We have a $N \times N$ matrix $\mathbf{C}$ corresponding to a *transport cost* of mapping a point $i \in [1..N]$ to a point $j \in [1..N]$ and two associated (discrete) probability distributions $\mathbf{p}$ and $\mathbf{q}$ with support in $[1..N]$. Our goal is then to find an *optimal coupling* that minimizes the cost of the coupling $\langle \boldsymbol{\pi}, \mathbf{C} \rangle$ and whose marginals are $\mathbf{p}$ and $\mathbf{q}$. This linear programming problem can be regularized using *entropic regularization*, *i.e*, finding coupling $\boldsymbol{\pi}$ that minimizes $\langle \boldsymbol{\pi}, \mathbf{C} \rangle - T\mathbf{H}(\boldsymbol{\pi})$ where $T > 0$ is a regularization parameter that can be seen as a temperature. This regularized problem has a lot of nice computational properties and its solution converges to the solution of the original problem as $T \longrightarrow 0$, see [8].

In this project we will aim at solving the inverse problem, *i.e*, one observes the optimal coupling and aims at learning the cost that led to this mapping. This problem often arises in economics or in sociology when one has to deduce the inner interactions that created a change. For example [7] faces the problem of optimal assignment and in [6] they give an example about international migrations.

This problem is actually extremely important in optimal transport, because optimal transport has mostly only one parameter and considering this metric as given is not always a mild assumption.

1.1 Related work

This inverse problem has not been widely studied in the literature, maybe because in machine learning applications one should often predict rather than observe.

For related work we can first refer to [3] that gives a really nice intuition on the problem and presents a supervised way of learning a *ground metric* over a set of histograms features. The authors formulate it as minimization of two convex functions and suggest applications to computer vision or natural language processing. Rather than considering a fully parametric formulation, as it will be the case with our reference paper [6], they consider optimising over the convex set of *metric matrices*. They use the optimal transport cost to define distances between both similar and dissimilar histograms in order to build a metric that can be expressed as the difference of convex functions. To minimize this objective function, they propose a projected subgradient descent with a local linearization to approximate the minimization of a non-convex function.

[4] proposes to use the optimal transport formulation to improve, still in a supervised way, the computation of the word mover's distance.

[5] proposes a parametric formulation. The authors consider an approach using maximum likelihood estimation of the cost matrix based on the observed transport plan. They parametrized the cost matrix as a bilinear form: $\mathbf{C}_{ij} = \langle \mathbf{x}_i, \mathbf{A}\mathbf{y}_j \rangle$, where $\mathbf{x}_i$ and $\mathbf{y}_j$ represent vectors of characteristics corresponding to points i and j. They showed that maximizing the likelihood with respect to $\mathbf{A}$ with $\hat{\boldsymbol{\pi}}$ observed amounts at looking for a matrix $\mathbf{C}$ such like the moments of the observed pairs computed for both the observed and the estimated plan are matched.

[7] proposes another parametric formulation, that may looks like a relaxation of what appeared in [5], where they define a set of basis functions of which the cost matrix $\mathbf{C}$ should be a linear combination, with the constraint that the moments of each of these basis functions should match for both the observed and the estimated plan corresponding to the estimated cost matrix. We can see it as a relaxation because we do not impose the basis function to really correspond to vectors' moments, they can even be non symmetric. We will discuss it more in details in subsection 2.2.

During the first semester of the MVA master, we have also seen related methods. First, in the course [10] we saw the dual formulation of the regularized problem, on which [6] relies heavily. Associated to that, the Sinkhorn's algorithm which is one of the foundations of the SISTA algorithm has also been described and implemented thanks to the Numerical Tours.

A lot of these articles and especially [6] relies on convex optimisation methods such like proximal gradient descent or local methods that have been studied in the course [9], with a strong focus on practical implementation.

1.2 Contribution

The work presented in [6] proposes a parametric method to learn the cost matrix. It provides both a flexible way to model the problem, because the choice of basis functions can be quite large, and a computational efficient way to solve it. Indeed, [7] shown that it can be solved as a convex optimisation problem. In [6], the authors add a l_1 penalty term in order to add sparsity to the learned parameters. With this penalty term, the problem seems well adapted to proximal splitting methods and indeed, methods like ISTA described in [1] are supposed to solve such problems. But taking inspiration of the well known coordinates descent of the Sinkhorn's algorithm, the authors propose a new algorithm that uses both proximal descent as well as coordinates descent. Taking advantage of coordinates descent result in a faster algorithm.

In this project, we are going to illustrate this in several numerical illustrations, by comparing the results of several algorithms on toy problems and then on a more serious application using real data, corresponding to commutes between residence and work place in major french cities.

2 Learning the cost

2.1 Optimal transport with entropic regularization

In this section we recall some facts about the optimal transport problem and we define the notations we are going to use in this report.

In this project we are going to study the optimal transport problem with entropic regularization, in the discrete case.

We suppose we have two probability distributions $\mathbf{p}, \mathbf{q} \in \mathbb{R}^N$ and we define the set of probability distributions over $[1..N]^2$ whose marginals are $\mathbf{p}$ and $\mathbf{q}$:

$$\Pi = \left\{ \boldsymbol{\pi} \in \mathbb{R}^{N \times N}, \boldsymbol{\pi} \geq 0 \mid \boldsymbol{\pi} \mathbf{1}_N = \mathbf{p},\ \boldsymbol{\pi}^T \mathbf{1}_N = \mathbf{q} \right\}.$$

If we define the entropy $\mathbf{H}$ as:

$$\forall \boldsymbol{\pi} \geq 0,\ \mathbf{H}(\boldsymbol{\pi}) = - \sum_{1 \leq i,j \leq N} \boldsymbol{\pi}_{ij} \ln(\boldsymbol{\pi}_{ij}), \quad \text{where} \quad 0 \ln(0) = 0,$$

the optimisation transport problem with entropic regularization is defined as:

$$\min_{\boldsymbol{\pi} \in \Pi} \ \langle \boldsymbol{\pi}, \mathbf{C} \rangle - T\mathbf{H}(\boldsymbol{\pi}), \tag{1}$$

where $T > 0$ is a regularization constant.

We also recall the dual formulation of (1):

$$\max_{\mathbf{u},\mathbf{v}\in\mathbb{R}^N} \langle \mathbf{p},\mathbf{u}\rangle + \langle \mathbf{q},\mathbf{v}\rangle - T \sum_{1\leq i,j\leq N} \exp\left(\frac{\mathbf{u}_i+\mathbf{v}_j-\mathbf{C}_{ij}}{T}\right) \tag{2}$$

An interesting thing to notice is that if we note:

$$J_{T,\mathbf{C}} : (\mathbf{u},\mathbf{v}) \longmapsto \langle \mathbf{p},\mathbf{u}\rangle + \langle \mathbf{q},\mathbf{v}\rangle - T \sum_{1\leq i,j\leq N} \exp\left(\frac{\mathbf{u}_i+\mathbf{v}_j-\mathbf{C}_{ij}}{T}\right),$$

then for $\lambda \in \mathbb{R}^*$:

$$J_{(\lambda T,\lambda\mathbf{C})}(\mathbf{u},\mathbf{v}) = \lambda J_{(T,\mathbf{C})}(\mathbf{u}/\lambda,\mathbf{v}/\lambda). \tag{3}$$

This shows that the problem is left invariant by multiplying T and $\mathbf{C}$ by the same constant, so without loss of generality, one can assume that $T = 1$. See section 4.1 for a discussion about this property. The dual problem now reads:

$$\max_{\mathbf{u},\mathbf{v}\in\mathbb{R}^N} \langle \mathbf{p},\mathbf{u}\rangle + \langle \mathbf{q},\mathbf{v}\rangle - \sum_{1\leq i,j\leq N} \exp\left(\mathbf{u}_i+\mathbf{v}_j-\mathbf{C}_{ij}\right). \tag{4}$$

And (4) is equivalent to a convex minimization problem (by taking the negative):

$$\min_{\mathbf{u},\mathbf{v}\in\mathbb{R}^N} \sum_{1\leq i,j\leq N} \exp\left(\mathbf{u}_i+\mathbf{v}_j-\mathbf{C}_{ij}\right) - \langle \mathbf{p},\mathbf{u}\rangle - \langle \mathbf{q},\mathbf{v}\rangle. \tag{5}$$

The first order conditions (5) give the relation:

$$\boldsymbol{\pi}_{ij} = \exp\left(\mathbf{u}_i+\mathbf{v}_j-\mathbf{C}_{ij}\right) \tag{6}$$

This is the form of the classical regularized optimal transport problem where one want to determine the optimal transport map. In what follows, we are going to address the inverse problem, *i.e*, to learn $\mathbf{C}$ when one observes $\boldsymbol{\pi}$.

2.2 Inverse problem

We now assume that we observe an optimal mapping $\hat{\boldsymbol{\pi}}$ whose marginals $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$ are known, and we aim solving an inverse problem, which is learning the cost $\mathbf{C}$ that led to this mapping. We are going to present the formulation adopted by [7], which is the one on which [6] is based on.

In this project we are going to learn a parametric representation of the cost $\mathbf{C}^{\boldsymbol{\beta}}$, with the parameters vector $\boldsymbol{\beta} \in \mathbb{R}^K$. In [7], the authors propose to learn the cost as a linear combination of basis matrices $(\mathbf{D}^k)_{1\leq k\leq K}$. This now defines a problem of parametric estimation, *i.e*, find $\boldsymbol{\beta} \in \mathbb{R}^K$ such that the cost can be expressed as:

$$\mathbf{C}^{\boldsymbol{\beta}} = \sum_{k=1}^{K} \boldsymbol{\beta}^k \mathbf{D}^k. \tag{7}$$

Remark 2.1. ***Evaluation of*** $\mathbf{C}^{\boldsymbol{\beta}}$ *Once we decided to learn a parametric estimation of the cost, we also have to define what will make a good estimation. And the question is not that easy: it is not obvious to find a good optimization problem to solve. One could think of a classical criterion such as* $\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^K}{\operatorname{argmin}} \|\hat{\boldsymbol{\pi}} - \boldsymbol{\pi}^{\boldsymbol{\beta}}\|^2$*, where* $\boldsymbol{\pi}^{\boldsymbol{\beta}}_{ij} = \exp\left(\mathbf{u}^{\boldsymbol{\beta}}_i + \mathbf{v}^{\boldsymbol{\beta}}_j - \mathbf{C}^{\boldsymbol{\beta}}_{ij}\right)$ *and* $(\mathbf{u}^{\boldsymbol{\beta}}, \mathbf{v}^{\boldsymbol{\beta}}) = \underset{\mathbf{u},\mathbf{v}\in\mathbb{R}^N}{\operatorname{argmin}} \sum_{1\leq i,j\leq N} \exp\left(\mathbf{u}_i + \mathbf{v}_j - \mathbf{C}_{ij}\right) - \langle \mathbf{p}, \mathbf{u}\rangle - \langle \mathbf{q}, \mathbf{v}\rangle$*. But we can anticipate a potential problem. First, we see that the* argmin *will not behave well inside such non linear expressions, meaning that we will have to solve a nested optimisation problem. And with such a nested problem one can wonder about the guarantees such a metric can bring us. We are not saying that this would be such a bad idea, but it seems rather complicated.*

Remark 2.2. ***Basis functions*** *A basis function is a* dissimilarity measure *between characteristics of the distributions* $\mathbf{p}$ *and* $\mathbf{q}$*. Here are some useful examples given in [6]:*

- *if* i *and* j *corresponds to entities, and* $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^K$ *correspond to some vectors characterizing these entities, then we can consider classical squared distance between the components of the vectors:* $\mathbf{D}^k_{ij} = (\mathbf{x}^k_i - \mathbf{x}^k_j)^2$*.*
- $\mathbf{D}^k_{ij}$ *can also be another dissimilarity measure, that does not need to be symmetric. In [6] they give the example of a situation where* i, j *are countries and* $\mathbf{D}^k_{ij}$ *is the proportion of inhabitants from* i *that does not speak the language of country* j*.*

Following the previous remark, we now explain how the authors proposed to solve this problem in [7] and what is the elegant solution they derived.

To determine the optimal parameter $\boldsymbol{\beta}$, the authors of [6] propose to look for

$$\boldsymbol{\pi}^{\boldsymbol{\beta}}_{ij} = \exp\left(\mathbf{u}_i + \mathbf{v}_j - \mathbf{C}^{\boldsymbol{\beta}}_{ij}\right),$$

respecting the following conditions:

$$\boldsymbol{\pi}^{\boldsymbol{\beta}} \in \hat{\Pi}, \tag{8a}$$

$$\forall k \in [1..K], \left\langle \boldsymbol{\pi}^{\boldsymbol{\beta}}, \mathbf{D}^k \right\rangle = \left\langle \hat{\boldsymbol{\pi}}, \mathbf{D}^k \right\rangle, \tag{8b}$$

where

$$\hat{\Pi} = \left\{\boldsymbol{\pi} \in \mathbb{R}^{N\times N}, \boldsymbol{\pi} \geq 0 \mid \boldsymbol{\pi}\mathbf{1}_N = \hat{\mathbf{p}},\ \boldsymbol{\pi}^T\mathbf{1}_N = \hat{\mathbf{q}}\right\},$$

where $\hat{\mathbf{p}}$ and $\hat{\mathbf{q}}$ are the observed marginals of $\hat{\boldsymbol{\pi}}$.

Condition (8b) is called *moment matching* condition because it amounts at computing the first order moment of $\mathbf{D}^k$ according to both transport maps.

We see that, at first glance, the conditions are more unusual that what we could have think of in the paragraph above 2.1. Here we do not want to minimize explicitly something, but we require the transport maps to coincide on basis functions. We refer to [5] to get an intuition of where this idea comes from.

But actually, what is extremely handy with this formulation is that it is equivalent to a more natural convex optimization problem.

Theorem 2.1. *Finding* $\boldsymbol{\pi}^{\boldsymbol{\beta}}_{ij} = \exp\left(\mathbf{u}_i + \mathbf{v}_j - \mathbf{C}^{\boldsymbol{\beta}}_{ij}\right)$ *satisfying conditions* (8) *is equivalent to solve the two following minimization problems:*

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^K} \left|\left\langle \hat{\boldsymbol{\pi}}, \mathbf{C}^{\boldsymbol{\beta}} \right\rangle - \mathbf{H}(\hat{\boldsymbol{\pi}}) - \mathbf{L}(\boldsymbol{\beta})\right| \tag{9}$$

$$\min_{\mathbf{u},\mathbf{v},\boldsymbol{\beta}} \mathbf{F}(\mathbf{u},\mathbf{v},\boldsymbol{\beta}) \tag{10}$$

Where $\mathbf{F}$ *is defined as:*

$$\mathbf{F}(\mathbf{u},\mathbf{v},\boldsymbol{\beta}) = \sum_{1\le i,j\le N} \exp\left(\mathbf{u}_i + \mathbf{v}_j - \mathbf{C}^{\boldsymbol{\beta}}_{ij}\right) + \sum_{1\le i,j\le N} \hat{\boldsymbol{\pi}}_{ij}\left(\mathbf{C}^{\boldsymbol{\beta}}_{ij} - \mathbf{u}_i - \mathbf{v}_j\right). \tag{11}$$

Remark 2.3. (9) *amounts at finding the* $\boldsymbol{\beta}$ *that minimizes the difference between the observed cost and the cost corresponding to the solution of the regularized optimal transport.*

Proof. A proof for this theorem can be stated using the envelope theorem. This proof is adapted from [7] theorem 3.

We are going to study the following optimization problem:

$$\operatorname*{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^K} \left|\left\langle \hat{\boldsymbol{\pi}}, \mathbf{C}^{\boldsymbol{\beta}} \right\rangle - \mathbf{H}(\boldsymbol{\pi}^{\boldsymbol{\beta}}) - \mathbf{L}(\boldsymbol{\beta})\right|, \tag{12}$$

where:

$$\mathbf{L}(\boldsymbol{\beta}) = \min_{\boldsymbol{\pi}\in\hat{\Pi}} \left\{\left\langle \boldsymbol{\pi}, \mathbf{C}^{\boldsymbol{\beta}} \right\rangle - \mathbf{H}(\boldsymbol{\pi})\right\}$$

We simply search for $\boldsymbol{\beta}$ minimizing the difference between the regularized cost of the observed $\hat{\boldsymbol{\pi}}$ and the one of $\boldsymbol{\pi}^{\boldsymbol{\beta}}$, learned by solving the regularized optimal transport problem with the parametrized cost. We can simplify this expression by removing the absolute value (by definition of $\mathbf{L}(\boldsymbol{\beta})$) and the constant term. (12) amounts at solving the optimization problem:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^K} \left\langle \hat{\boldsymbol{\pi}}, \mathbf{C}^{\boldsymbol{\beta}} \right\rangle - \mathbf{L}(\boldsymbol{\beta}). \tag{13}$$

As $\boldsymbol{\beta} \mapsto \left\langle \boldsymbol{\pi}, \mathbf{C}^{\boldsymbol{\beta}} \right\rangle - \mathbf{H}(\boldsymbol{\pi})$ is linear in $\boldsymbol{\beta}$, $\mathbf{L}(\boldsymbol{\beta})$ is concave in $\boldsymbol{\beta}$. Hence $\boldsymbol{\beta} \mapsto \left\langle \hat{\boldsymbol{\pi}}, \mathbf{C}^{\boldsymbol{\beta}} \right\rangle - \mathbf{L}(\boldsymbol{\beta})$ is convex as a sum of convex functions.

Suppose that its minimum is reached at $\hat{\boldsymbol{\beta}}$, it is hence characterized by the critical point condition:

$$\forall k \in [1..K],\ \frac{\partial}{\partial \boldsymbol{\beta}^k}\left\langle \hat{\boldsymbol{\pi}}, \mathbf{C}^{\hat{\boldsymbol{\beta}}} \right\rangle = \left\langle \hat{\boldsymbol{\pi}}, \mathbf{D}^k \right\rangle = \frac{\partial}{\partial \boldsymbol{\beta}^k}\mathbf{L}(\hat{\boldsymbol{\beta}}).$$

And using the Envelope Theorem:

$$\frac{\partial}{\partial \boldsymbol{\beta}^k}\mathbf{L}(\hat{\boldsymbol{\beta}}) = \left\langle \boldsymbol{\pi}^{\hat{\boldsymbol{\beta}}}, \mathbf{D}^k \right\rangle,$$

which explains the relation with the moment matching condition.

Now, we still have to show that this amounts at solving the problem (10). Using the dual formulation (4), we have that:

$$\mathbf{L}(\boldsymbol{\beta}) = \max_{\mathbf{u},\mathbf{v}\in\mathbb{R}^N} \langle \mathbf{p}, \mathbf{u}\rangle + \langle \mathbf{q}, \mathbf{v}\rangle - \sum_{1\leq i,j\leq N} \exp\left(\mathbf{u}_i + \mathbf{v}_j - \mathbf{C}^{\boldsymbol{\beta}}_{ij}\right).$$

So we can rewrite our problem (13):

$$\begin{aligned}
(13) \Leftrightarrow \min_{\boldsymbol{\beta}\in\mathbb{R}^K} \left\langle \hat{\boldsymbol{\pi}}, \mathbf{C}^{\boldsymbol{\beta}}\right\rangle - \max_{\mathbf{u},\mathbf{v}\in\mathbb{R}^N} &\left\{\langle \mathbf{p}, \mathbf{u}\rangle + \langle \mathbf{q}, \mathbf{v}\rangle - \sum_{1\leq i,j\leq N} \exp\left(\mathbf{u}_i + \mathbf{v}_j - \mathbf{C}^{\boldsymbol{\beta}}_{ij}\right)\right\}, && \text{(14a)}\\
\Leftrightarrow \min_{\boldsymbol{\beta}\in\mathbb{R}^K} \left\langle \hat{\boldsymbol{\pi}}, \mathbf{C}^{\boldsymbol{\beta}}\right\rangle + \min_{\mathbf{u},\mathbf{v}\in\mathbb{R}^N} &\left\{\sum_{1\leq i,j\leq N} \exp\left(\mathbf{u}_i + \mathbf{v}_j - \mathbf{C}^{\boldsymbol{\beta}}_{ij}\right) - \langle \mathbf{p}, \mathbf{u}\rangle - \langle \mathbf{q}, \mathbf{v}\rangle\right\}, && \text{(14b)}\\
\Leftrightarrow \min_{\mathbf{u},\mathbf{v},\boldsymbol{\beta}} \mathbf{F}(\mathbf{u},\mathbf{v},\boldsymbol{\beta}). & && \text{(14c)}
\end{aligned}$$

Indeed, (14c) is because:

$$\sum_{1\leq i,j\leq N} \hat{\boldsymbol{\pi}}_{ij}\mathbf{u}_i = \sum_{i=1}^{N}\sum_{j=1}^{N} u_i \hat{\boldsymbol{\pi}}_{ij} = \langle \mathbf{u}, \hat{\mathbf{p}}\rangle,$$

and similarly:

$$\sum_{1\leq i,j\leq N} \hat{\boldsymbol{\pi}}_{ij}\mathbf{v}_j = \langle \mathbf{v}, \hat{\mathbf{q}}\rangle.$$

□

Another useful property of **F** is the following.

Property 2.1. **F** *defined in* (11) *is convex.*

Proof. First recall from (7) that $\mathbf{C}^{\boldsymbol{\beta}}$ is linear in $\boldsymbol{\beta}$.

exp is convex.

Then **F** is convex by composition of a convex function by a linear function and by sum of convex functions. □

The main contribution of [6] is to add a regularization term on problem (10). They considered a l_1 penalty cost, in order to promote sparse solutions for $\boldsymbol{\beta}$. Adding this regularization also makes the objective function coercive and strictly convex, that are very interesting properties (proof in the original article).

The final problem we are going to study is defined in (15):

$$\min_{\mathbf{u},\mathbf{v},\boldsymbol{\beta}} \mathbf{F}(\mathbf{u},\mathbf{v},\boldsymbol{\beta}) + \gamma\|\boldsymbol{\beta}\|_1, \tag{15}$$

where $\gamma > 0$ is a regularization parameter.

We define:

$$\boldsymbol{\Phi}(\mathbf{u},\mathbf{v},\boldsymbol{\beta}) = \mathbf{F}(\mathbf{u},\mathbf{v},\boldsymbol{\beta}) + \gamma\|\boldsymbol{\beta}\|_1. \tag{16}$$

Remark 2.4. ***Separability*** *What is nice with the form of* $\mathbf{F}$ *in* (11) *is that it is separable in* $\boldsymbol{\beta}$ *and* $(\mathbf{u},\mathbf{v})$*. And so is* $\boldsymbol{\Phi}$*. So, one can find a way to solve it first for* $(\mathbf{u},\mathbf{v})$ *and then for* $\boldsymbol{\beta}$*. We will discuss it more in details in subsection 2.3.*

2.3 Problem resolution

To solve this optimization problem, we are going to follow the sense of remark 2.4.

2.3.1 Solving for $(\mathbf{u},\mathbf{v})$

The ideas of both algorithms we are going to describe is to alternate minimization steps on $(\mathbf{u},\mathbf{v})$ with minimization on $\boldsymbol{\beta}$. The expression in (11) is actually the same as (5), with $\mathbf{C}=\mathbf{C}^{\boldsymbol{\beta}}$.

Hence, if $\boldsymbol{\beta}$ is fixed:

$$\underset{\mathbf{u},\mathbf{v}}{\operatorname{argmin}}\ \boldsymbol{\Phi}(\mathbf{u},\mathbf{v},\boldsymbol{\beta}) = \underset{\mathbf{u},\mathbf{v}}{\operatorname{argmin}} \sum_{1\le i,j\le N} \exp\left(\mathbf{u}_i+\mathbf{v}_j-\mathbf{C}_{ij}\right) - \langle \hat{\mathbf{p}},\mathbf{u}\rangle - \langle \hat{\mathbf{q}},\mathbf{v}\rangle\,. \tag{17}$$

Which shows that (15) can be solved for $(\mathbf{u},\mathbf{v})$ using, for example, Sinkhorn's algorithm.

2.3.2 Solving for $\boldsymbol{\beta}$

Now we suppose that $(\mathbf{u},\mathbf{v})$ are fixed.

$\boldsymbol{\Phi}(\mathbf{u},\mathbf{v},\cdot)$ is the sum of a convex and differentiable function with a convex non-differentiable function. For such configuration, one can use the ISTA framework, which is a special case of the Forward-Backward algorithm described during the course of [9] or in [2]. This is the algorithm the authors of [6] used for the minimization over $\boldsymbol{\beta}$.

2.4 Algorithms

2.4.1 ISTA

The first algorithm we are going to describe is called in [6] *ISTA*. We are also going to call it this way, despite a slight abuse of name (ISTA was orignally created to solve quadratic problems with l_1 regularization).

This algorithm alternates classical gradient descent steps on $(\mathbf{u},\mathbf{v})$ with proximal gradient descent steps. ρ will be the step size. This proximal gradient descent steps reads:

$$\boldsymbol{\beta}_{t+1} = \operatorname{prox}_{\gamma\rho|\cdot|}\left(\boldsymbol{\beta}^t - \rho\nabla_{\boldsymbol{\beta}}\mathbf{F}(\mathbf{u}_{t+1},\mathbf{v}_{t+1},\boldsymbol{\beta}^t)\right),$$

Where, if we define $\boldsymbol{\pi}^{\boldsymbol{\beta}^t}_{ij} = \exp\left(\mathbf{u}_i+\mathbf{v}_j-\mathbf{C}^{\boldsymbol{\beta}^t}_{ij}\right)$:

$$\nabla_{\boldsymbol{\beta}}\mathbf{F}(\mathbf{u},\mathbf{v},\boldsymbol{\beta}) = \left\langle \hat{\boldsymbol{\pi}} - \boldsymbol{\pi}^{\boldsymbol{\beta}}, \mathbf{D}^k \right\rangle$$

Where $\operatorname{prox}_{\gamma\rho|\cdot|}$ is the thresholding operator, defined as:

$$\operatorname{prox}_{\gamma\rho|\cdot|}(x) = \begin{cases} x-\gamma\rho & \text{if} \quad x>\gamma\rho,\\ 0 & \text{if} \quad |x|\le\gamma\rho,\\ x+\gamma\rho & \text{otherwise.}\end{cases}$$

Which can be efficiently summarized:

$$\operatorname{prox}_{\gamma\rho|\cdot|}(x) = \operatorname{sign}(x) \max\left\{|x| - \gamma\rho, 0\right\}.$$

We are going to need the gradient of $\mathbf{F}$ with respect to $\mathbf{u}$ and $\mathbf{v}$.

$$\forall i_0 \in [1..N],\ \frac{\partial}{\partial \mathbf{u}_{i_0}} \mathbf{F}(\mathbf{u}, \mathbf{v}, \boldsymbol{\beta}) = \sum_{1 \leq j \leq N} \exp\left(\mathbf{u}_{i_0} + \mathbf{v}_j - \mathbf{C}^{\boldsymbol{\beta}}_{ij}\right) - \hat{\mathbf{p}}_{i_0}.$$

For $\mathbf{v}$ we have a symmetric expression.

The corresponding algorithm is algorithm 1.

```
Algorithm 1: ISTA
Data: $\gamma > 0$ the regularisation parameter, $\rho > 0$ the step-size, $M$ the maximum number of
      iterations, $\mathbf{D} \in \mathbb{R}^{K \times N \times N}$ the dissimilarity measures matrices, $\hat{\boldsymbol{\pi}}$ the observed map,
      $\mathbf{u}_0, \mathbf{v}_0$ the initial potentials and $\boldsymbol{\beta}_0$ the initial parameters vector
Result: $\boldsymbol{\beta}$
for $i \in [0..M]$ do
    Set $\mathbf{C}^{\boldsymbol{\beta}^t} = \sum_{k=1}^{K} \boldsymbol{\beta}^k \mathbf{D}^k$.
    (Gradient step). Update:
        $\mathbf{u}_{t+1} = \rho \nabla_{\mathbf{u}} \mathbf{F}(\mathbf{u}^t, \mathbf{v}^t, \boldsymbol{\beta}^t)$
        $\mathbf{v}_{t+1} = \rho \nabla_{\mathbf{v}} \mathbf{F}(\mathbf{u}^t, \mathbf{v}^t, \boldsymbol{\beta}^t)$
    (ISTA step). Let $\boldsymbol{\pi}^{\boldsymbol{\beta}^t}_{ij} = \exp\left(\mathbf{u}^t_i + \mathbf{v}^t_j - \mathbf{C}^{\boldsymbol{\beta}^t}_{ij}\right)$. Update:
        $\forall k \in [1..K],\ \boldsymbol{\beta}^{t+1^k} = \operatorname{prox}_{\gamma\rho|\cdot|}\left((\boldsymbol{\beta}^t)^k - \rho\left\langle \hat{\boldsymbol{\pi}} - \boldsymbol{\pi}^{\boldsymbol{\beta}^t}, \mathbf{D}^k \right\rangle\right).$
end
```

2.4.2 SISTA

The main algorithm of [6] is the SISTA algorithm, which alternates between Sinkhorn's minimization steps and proximal gradient descent steps. This algorithm is described in algorithm 2.

The convergence of this algorithm is fully derived in [6] under the following conditions:

$$(\mathbf{D}^k)_{1 \leq k \leq K} \text{ are linearly independent matrices,} \tag{18a}$$

$$\forall k \in [1..K],\ ,\ \sum_{1 \leq i,j \leq N} \mathbf{D}^k_{ij} = 0, \tag{18b}$$

$$\forall (i,j) \in [1..N]^2,\ \hat{\boldsymbol{\pi}}_{ij} > 0. \tag{18c}$$

And the rate of convergence is linear (like for Sinkhorn's algorithm).

Remark 2.5. *The authors of [6] explain that the condition* (18b) *is without loss of generality. We are going to provide more details about this claim.*

```
Algorithm 2: SISTA
Data: $\gamma > 0$ the regularisation parameter, $\rho > 0$ the step-size, $M$ the maximum number of
      iterations, $\mathbf{D} \in \mathbb{R}^{K \times N \times N}$ the dissimilarity measures matrices, $\hat{\boldsymbol{\pi}}$ the observed map,
      $\mathbf{u}_0, \mathbf{v}_0$ the initial potentials and $\boldsymbol{\beta}_0$ the initial parameters vector
Result: $\boldsymbol{\beta}$
for $i \in [0..M]$ do
    Set $\mathbf{C}^{\boldsymbol{\beta}^t} = \sum_{k=1}^{K} \boldsymbol{\beta}^k \mathbf{D}^k$. Set $\mathbf{K}^{\boldsymbol{\beta}^t} = \exp(-\mathbf{C}^{\boldsymbol{\beta}^t})$.
    (Sinkhorn step). Update:
        $\exp(\mathbf{u}_{t+1}) = \frac{\hat{\mathbf{p}}}{\mathbf{K}^{\boldsymbol{\beta}^t} \mathbf{v}_t}$   # Element-wise exponentiation and division.
        $\exp(\mathbf{v}_{t+1}) = \frac{\hat{\mathbf{q}}}{(\mathbf{K}^{\boldsymbol{\beta}^t})^T \mathbf{u}_{t+1}}$
    (ISTA step). Let $\boldsymbol{\pi}^{\boldsymbol{\beta}^t}_{ij} = \exp\left(\mathbf{u}^t_i + \mathbf{v}^t_j - \mathbf{C}^{\boldsymbol{\beta}^t}_{ij}\right)$. Update:
        $\forall k \in [1..K],\ {\boldsymbol{\beta}^{t+1}}^k = \mathrm{prox}_{\gamma\rho|\cdot|}\left((\boldsymbol{\beta}^t)^k - \rho\left\langle \hat{\boldsymbol{\pi}} - \boldsymbol{\pi}^{\boldsymbol{\beta}^t}, \mathbf{D}^k \right\rangle\right).$
end
```

Let's define $\boldsymbol{\Lambda} : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^K \longrightarrow \mathbb{R}^{N \times N}$*, defined entrywise by:*

$$\forall (i,j) \in [1..N],\ \boldsymbol{\Lambda}(\mathbf{u}, \mathbf{v}, \boldsymbol{\beta})_{ij} = \mathbf{u}_i + \mathbf{v}_j - \mathbf{C}^{\boldsymbol{\beta}}_{ij} \tag{19}$$

If (18b) *is not satisfied, the authors define:*

$$\widetilde{\mathbf{D}}^k_{ij} = \mathbf{D}^k_{ij} - \mathbf{a}^k ij - \mathbf{b}^k_{ij},$$

where $\mathbf{a}^k_{ij} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{D}^k_{ij}$ *and* $\mathbf{b}^k_{ij} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{D}^k_{ij} - \frac{1}{N^2} \sum_{1 \leq m,n \leq N} \mathbf{D}^k_{mn}$.

$\widetilde{\mathbf{D}}$ *verifies condition* (18b).

Now we define $\widetilde{\boldsymbol{\Lambda}}$ *as:*

$$\forall (i,j) \in [1..N],\ \widetilde{\boldsymbol{\Lambda}}(\mathbf{u}, \mathbf{v}, \boldsymbol{\beta})_{ij} = \mathbf{u}_i + \mathbf{v}_j - \widetilde{\mathbf{C}}^{\boldsymbol{\beta}}{}_{ij}. \tag{20}$$

And as $\boldsymbol{\Lambda}(\mathbf{u}, \mathbf{v}, \boldsymbol{\beta}) = \widetilde{\boldsymbol{\Lambda}}(\mathbf{u} - \sum_{k=1}^{K} \boldsymbol{\beta}^k \mathbf{a}^k, \mathbf{v} - \sum_{k=1}^{K} \boldsymbol{\beta}^k \mathbf{b}^k, \boldsymbol{\beta})$*, Lemma 3.2 of [6] stating the injectivity of* $\boldsymbol{\Lambda}$ *under condition* (18b) *can be applied to* $\widetilde{\boldsymbol{\Lambda}}$*, from which the injectivity of* $\boldsymbol{\Lambda}$ *is directly deduced (using the characterisation of injectivity of linear applications).*

Remark 2.6. *According to the authors of [6], condition* (18c) *is mild since we can define* $I^+ = \{(i,j) \mid , \hat{\boldsymbol{\pi}} > 0\}$. *Indeed, the whole proof of convergence still holds, but the interpretation of* $\mathbf{F}$ *given by theorem 2.1 does not hold anymore.*

3 Numerical applications

In this section we are going to compare SISTA with ISTA on several toy problems, with a known parametric cost. We sill study the speed of convergence, the stability and the quality of the output result. All numerics have been done with Python 3.9.5, using exclusively Numpy 1.21.

3.1 $\boldsymbol{\Phi}$ minimization

A first thing to check is that the algorithms do what we expected, *i.e*, minimizing $\boldsymbol{\Phi}$.

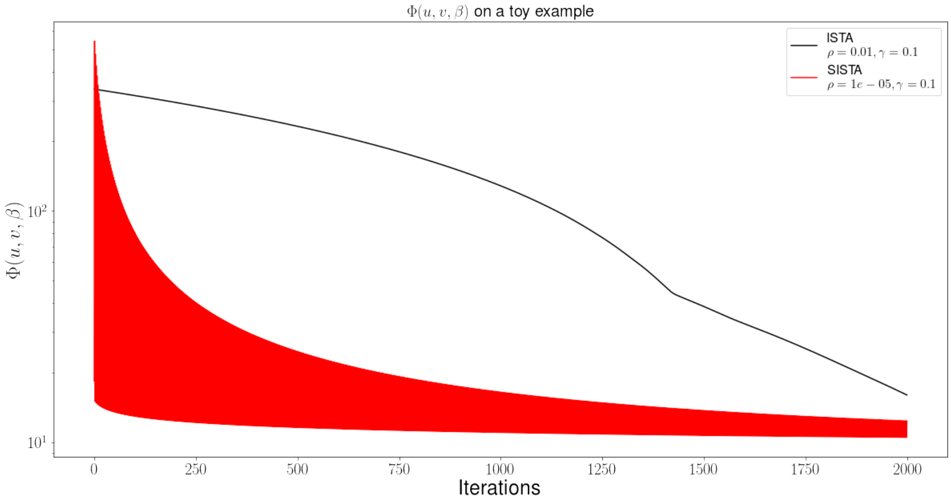

Figure 1: Evolution of $\boldsymbol{\Phi}$ for ISTA and SISTA. We observe oscillations for SISTA.

The data on Figure 1 have been generated using the following process:

1. Generate 10 random positive symmetric matrices in $\mathbb{R}^{100\times 100}$.
2. Generate a random $\boldsymbol{\beta} \in \mathbb{R}^{10}$.
3. Compute the associated cost $\mathbf{C}^{\boldsymbol{\beta}}$,
4. Solve the associated regularized optimal transport in order to obtain $\hat{\boldsymbol{\pi}}$.
5. Select for both methods a step size using a grid search.

On Figure 1 we see the evolution of $\boldsymbol{\Phi}$ for both ISTA and SISTA algorithm. SISTA converges quicker than ISTA, in term of number of iterations, and has the same complexity as the operations are similar between both (only matrices sum or products).

We notice that SISTA does not decrease $\boldsymbol{\Phi}$ at each iteration. Indeed, like for Sinkhorn's algorithm, the iterations have no reason to monotonously decrease the objective function.

Table 1: Example of $\boldsymbol{\beta}_{learned}$ and $\boldsymbol{\beta}_{real}$

$\boldsymbol{\beta}$	$\boldsymbol{\beta}^1$	$\boldsymbol{\beta}^2$	$\boldsymbol{\beta}^3$	$\boldsymbol{\beta}^4$	$\boldsymbol{\beta}^5$
$\boldsymbol{\beta}_{real}$	0.25	0.5	1.25	0.5	0.75
$\boldsymbol{\beta}_{learned}$	0.55	0.57	0.65	0.57	0.60

3.2 Convergence of the cost

Now, the most interesting part is to see if indeed these algorithms can *learn* the cost. Our next experiment analyzes the evolution of $\boldsymbol{\beta}$ during the iterations. The result is plotted on Figure 2.

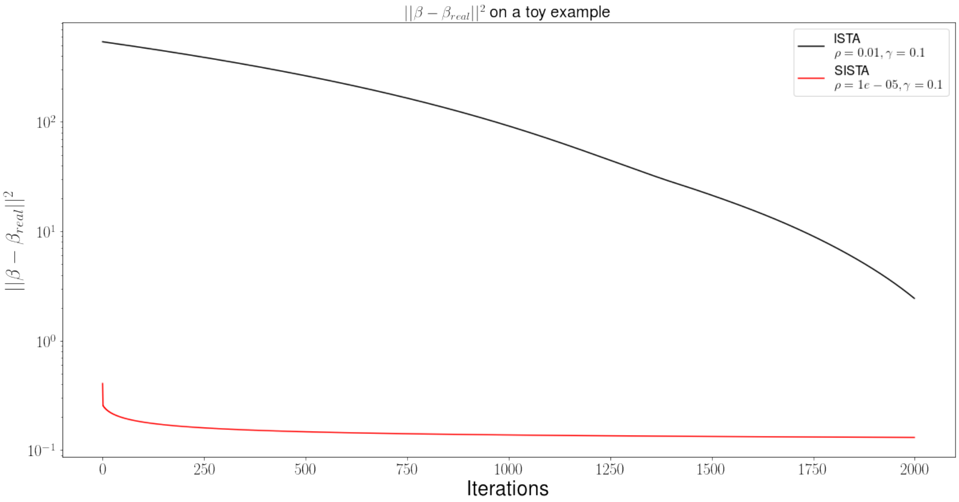

Figure 2: Evolution of $\|\boldsymbol{\beta} - \boldsymbol{\beta}_{real}\|^2$ for ISTA and SISTA.

We used the same data on Figure 2 than for Figure 1. We see that the $\boldsymbol{\beta}$ returned by SISTA get closer to $\boldsymbol{\beta}_{real}$ more quickly than for ISTA.

But the result of the convergence is not very precise. Indeed, $\boldsymbol{\beta}_{learned}$ was in all tested cases not really close to $\boldsymbol{\beta}_{real}$. The magnitudes of the coefficients matched, but they are still quite different. The coefficients of $\boldsymbol{\beta}_{learned}$ seems to be less contrasted than for $\boldsymbol{\beta}_{real}$. Table 1 shows the first coefficients of both vectors.

But on Table 1 we see that $\boldsymbol{\beta}_{learned}$ follows roughly $\boldsymbol{\beta}_{real}$ (the coefficients can sorted in the same increasing order). So at least our experiments shows than we can detect the trends that generated the transport map.

3.3 Real data example

Despite this work has not proven that the SISTA algorithm can provide a way to exactly learn a cost, it can still be useful to highlight key attributes. We provide another numerical experiment, using real world data. We studied the commute of french people between their cities of residence and their cities of work. We used 2017 Insee data [12] to conduct this work, as well as the IRIS Contours dataset from IGN [11].

We keep only the pairs of cities that had a people stream greater than 1000. Figure 3 shows a subset of this dataset on Île-de-France.

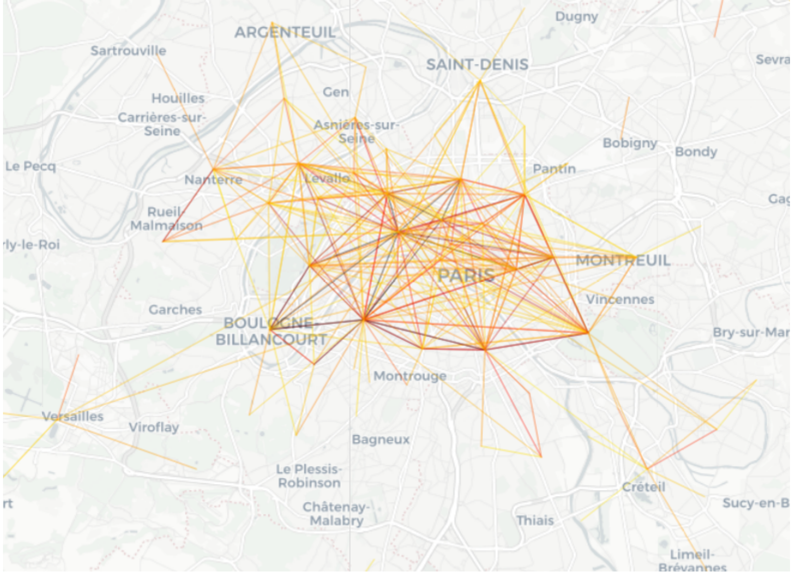

Figure 3: Home to work place commute on a subset of Île-de-France. The more dark orange is the color the more people do the commute between the cities.

For each town we have access to the following attributes:

1. Number of inhabitants, `NBPOP`
2. Median income: `MED`.
3. Number of working people, `NBEMP`.
4. Number of workers in agriculture, silviculture and fishing, `NA5_AZ`.
5. Number of industry workers, `NA5_BE`.
6. Number of construction worker `NA5_FZ`.
7. Number of worker in commerce, transports and diverse services, `NA5_GU`.
8. Number of workers in public administration, teaching, health and social action, `NA5_OQ`.
9. Number of created entreprises, `NBENT`.
10. Number of housing per inhabitant, `NBLOG`.
11. Geographical position (EPSG:2154), `x, y`. The corresponding dissimilarity will be refers as `DIST`.

It gives 11 attributes to compare the cities. For all of them, except for the distance which will be classical euclidean distance, the dissimilarity is the ratio between the attribute of city i and the attribute of city j:

$$\forall k \in [1..10],\ \mathbf{D}^k_{ij} = \frac{attribute^k_j}{attribute^k_i}.$$

Table 2: $\boldsymbol{\beta}_{learned}$ on the real dataset.

	NBPOP	MED	NBEMP	NA5_AZ	NA5_BE	NA5_FZ	NA5_GU	NA5-OQ	NBENT	NBLOG	DIST
$\boldsymbol{\beta}$	0	0.083	0	0	0.173	0	0.563	0	0	0	0.01

We used this definition for the basis matrices because the problem is not symmetric.

And finally the dissimilarity matrices $\mathbf{D}^k$ are normalized (by the standard deviation of their coefficients). We ran the algorithm on this real dataset. Figure 4 shows the evolution of $\boldsymbol{\Phi}$. We see that $\boldsymbol{\Phi}$ converges without oscillations, compared to what we had on the toy dataset.

Remark 3.1. *The authors of [6] says that if condition* (18c) *is not satisfied, we should only consider the summation set I^+, as explained in remark 2.6. However, we did not managed to make the SISTA algorithm converge with such considerations. Hence, we used* $\mathbf{F}$ *with a summation on all the indices.*

One should also note that we had to use a really low step-size ρ to avoid numerical issues. In order to get sparsity on $\boldsymbol{\beta}$, we used a really high regularization value. Table 2 shows the coefficients of $\boldsymbol{\beta}$ corresponding to each attribute of the cities. A lot of coefficients have been thresholded. According to this experiment, the major factors defining the commute are:

1. The number workers in commerce, transport and services.
2. The number of workers in industry.
3. The median income.
4. The distance between the cities.

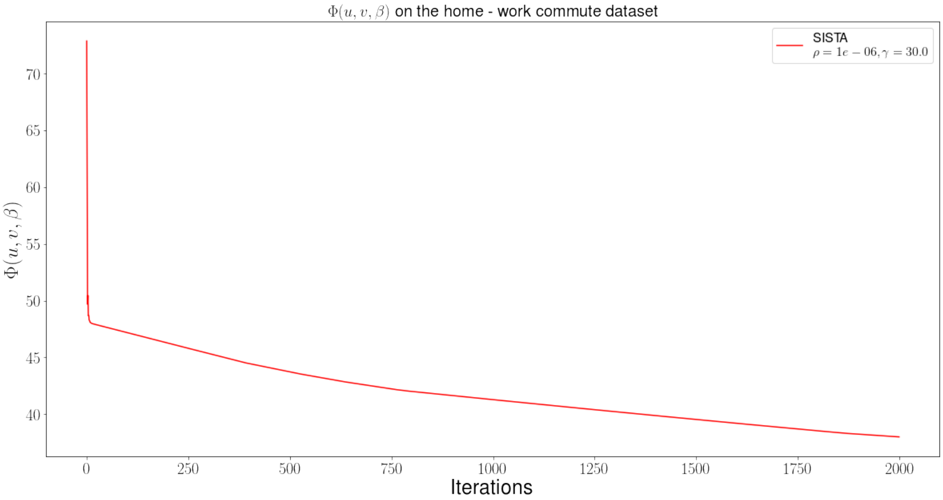

Figure 4: Minimization of $\boldsymbol{\Phi}$ on the commute dataset.

4 Conclusion and limits

Article [6] presents an interesting idea to learn the cost of optimal transport. Despite this method is not very precise for the estimation of $\boldsymbol{\beta}$, it can nevertheless highlight important features of the problem. We also would like to make some remarks on some unclear aspect of the article.

4.1 Remarks on the article

Lipschitzianity of $\nabla\mathbf{F}$. A major hypothesis for the convergence of ISTA and SISTA alogrithm is the Lipschitzianity of $\nabla\mathbf{F}$. In the convergence proof, the authors of [6] suppose that this hypothesis is true locally, but with potentially big Lipschitz constant. This can explain why we sometimes face numerical issues or why the step-size needs to be very small to have convergence.

Convergence of the cost. In [6] they do not provide any clear analysis about how the SISTA algorithm can retrieve a known cost (the only convergence simulation were conducted on $\mathbf{\Phi}$). In our experiments it performs badly on this task while it was supposed to be its primary task.

T scaling. Assuming that $T = 1$ may be true in theory, but in practice in implies to scale the cost of a factor $1/T$, which can be fine if T is big but can rapidly cause numeric instabilities, especially because we can not do all the computations in log-space.

Case where $\hat{\boldsymbol{\pi}} = 0$. As said in remark 2.6 and remark 3.1, the behavior of the algorithm when $\hat{\boldsymbol{\pi}}$ is not strictly positive should maybe be precised.

Extend the formulation. Maybe an solution to improve the quality of the estimate is to use stronger conditions than the moment-matching one on $\boldsymbol{\pi}^{\boldsymbol{\beta}}$, which is equivalent to a one dimensional constraint, (9).

4.2 Connections with the course

The connections with the courses are mainly on the parts related to Sinkhorns. We used extensively its dual formulation in theorem 2.1, and SISTA algorithm is based on Sinkhorn's iteration. It hence can be seens as an extension of Sinkhorn's algorithm. We also talk about the stability of Sinkhorn and played with it in the Numerical Tours. Hence, we are aware that scaling the cost by $1/T$ can cause numerical issues.

Bisections method This is more a remark on the project than on the original article, but we did not succeed in implementing the third algorithm the authors used (they are not very explicit on the method but they mention 1D bisections method). We didn't reach convergence in our implementation.

References

[1] Amir Beck and Marc Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 183–202. URL: https://doi.org/10.1137/080716542.

[2] H.H. Bauschke and Patrick Louis Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011, p. 468. DOI: 10.1007/978-1-4419-9467-7. URL: https://hal.inria.fr/hal-00643354.

[3] Marco Cuturi and David Avis. "Ground Metric Learning". In: *Journal of Machine Learning Research* 15.17 (2014), pp. 533–564. URL: http://jmlr.org/papers/v15/cuturi14a.html.

[4] Gao Huang et al. "Supervised Word Mover's Distance". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper/2016/file/10c66082c124f8afe3df4886f5e516e0-Paper.pdf.

[5] Arnaud Dupuy, Alfred Galichon, and Yifei Sun. "Estimating matching affinity matrices under low-rank constraints". In: *Information and Inference: A Journal of the IMA* 8.4 (Aug. 2019), pp. 677–689. ISSN: 2049-8772. URL: https://doi.org/10.1093/imaiai/iaz015.

[6] Guillaume Carlier et al. *SISTA: learning optimal transport costs under sparsity constraints*. 2020. arXiv: 2009.08564 [math.OC].

[7] Alfred Galichon and Bernard Salanié. "Cupid's Invisible Hand: Social Surplus and Identification in Matching Models". In: *SSRN* (2020). URL: http://dx.doi.org/10.2139/ssrn.1804623.

[8] Gabriel Peyré and Marco Cuturi. *Computational Optimal Transport*. 2020. arXiv: 1803.00567 [stat.ML].

[9] Emilie Chouzenoux. *Foundations of Distributed and Large Scale Computing Optimization*. ENS Paris-Saclay, Master MVA, 2021. URL: https://pages.saclay.inria.fr/emilie.chouzenoux/ECP/index.htm.

[10] Gabriel Peyré. *Computational Optimal Transport*. ENS Paris-Saclay, Master MVA, 2021.

[11] IGN. *Contours IRIS*. URL: https://geoservices.ign.fr/contoursiris (visited on 12/29/2021).

[12] Insee. *Insee website*. URL: https://www.insee.fr/ (visited on 12/29/2021).