KYOTO UNIVERSITY

DOCTORAL THESIS

Measure Transport Approaches for Data Visualization and Learning

Author:
Vivien Pierre François SEGUY

Supervisor:
Pr. Akihiro YAMAMOTO

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Informatics

in the

Yamamoto-Cuturi Laboratory
Graduate School of Informatics

June 26, 2018

Abstract

Measure Transport Approaches for Data Visualization and Learning

by Vivien Pierre François SEGUY

Statistics and probability theory are today's main mathematical frameworks to analyze data and learn from them. The central concept is the notion of probability measure: a given data set can be seen as a discrete probability measure and many machine learning and statistical algorithms boil down to fitting a parameterized probability measure to an empirical measure. In some cases, a datum itself may be represented as a probability measure, as for instance for histogram data. Within this formalism, a *transport map* describes a way to map one probability measure to another: a map f is said to be a transport map between a source random variable X and a target random variable Y when $f(X)$ and Y have the same distribution. Such maps can be useful in numerous tasks, and play for instance a central role in current state-of-the-art techniques for learning generative models, or in Bayesian statistics. When a transport map is chosen so as to minimize the cost of transporting the mass from the source probability measure μ to the target probability measure ν with respect to some ground cost, we say that this map is *optimal*. *Optimal maps* or their relaxed counterpart *optimal couplings* define in particular the notion of Wasserstein distance, also known as *optimal transport metric* or *earth mover distance*, which appears to be a powerful metric between probability measures. Hence, understanding how to leverage the geometry of the space of probability measures equipped with the Wasserstein distance in order to build efficient learning algorithms is a major concern nowadays.

In the present thesis, I tackle both computational and methodology challenges related to the concept of measure transport, as well as showcase its potential in numerous experiments for data visualization and learning. In Chapter 2 a fast flow-based algorithm is proposed in order to compute efficiently a transport map between 2-dimensional probability measures supported on a rectangular domain, and it is shown that this provides a means to compute *cartograms*, which are 2-dimensional geographic maps where each region is distorted so that its surface area becomes proportional to some given data. Cartograms provide a useful visualization of statistical geographic data as highlighted on several examples. The following chapters then focus on *optimal* transport. In Chapter 3, stochastic algorithms are proposed for the computation of regularized optimal transport in the large-scale or continuous setting. The convex regularization of the primal problem results in an unconstrained dual problem which can be solved by stochastic gradient ascent. Approximate optimal maps are then computed by parameterizing them with deep neural networks and approximating the barycentric projection of a regularized optimal plan. Chapters 4 and 5 finally investigate how to use the geometry of the 2-Wasserstein space to build algorithms capable of finding principal geodesics of a data set of probability measures, in an attempt to perform a generalized version of PCA in the Wasserstein space. For that purpose, generalized Riemannian geometry concepts such as tangent spaces, tangent vectors and geodesics are leveraged in order to formulate appropriate optimization problems that can be solved by proximal or gradient-based methods.

Acknowledgements

I would like to deeply thank my supervisor Akihiro Yamamoto who welcomed me at the Yamamoto-Cuturi laboratory and provided me with a perfect working environment, and also Marco Cuturi who has supervised my research during my first two years and half at the laboratory.

I am also very grateful to Jean-François Aujol, Jérémie Bigot, Nicolas Papadakis, and Elza Cazelles for welcoming me several times at the *Institut de Mathématiques de Bordeaux*, where we could pursue collaborations in the best conditions.

I am also extremely thankful to all my other co-authors, Mathieu Blondel, Nicolas Courty, Bharath Bhushan Damodaran, Rémi Flamary, Michael T. Gastner, Pratyush More and Antoine Rolet for the very valuable discussions and fruitful collaborations which were carried out during the last four years.

I would also like to thank the organizers of the several workshops I had the chance to participate and and where I could present my research. Each of these workshop has been a turning point in my PhD research thanks to some important encounters with other PhD students, researchers or professors who were always generous with their time. I am hence thankful to Sanvesh Srivastava who welcomed me at the *Stochastic Optimization for Large Scale Optimal Transport* workshop of the JSM2017 conference, to Youssef Marzouk who invited me at the *Measure transport approaches for statistical problems* mini-symposium of the CSE17 conference, and to Guillaume Carlier and Marco Cuturi (again) who invited me at the *Optimal Transport meets Probability, Statistics and Machine Learning* workshop in Oaxaca, Mexico.

Finally, I am thankful to the reviewers of my thesis, Pr. Toshiyuki Tanaka, Pr. Naonori Ueda, Pr. Akihiro Yamamoto and Pr. Nobuo Yamashita for their careful reading and valuable comments which have helped me improve this manuscript.

List of Publications

Chapter 2 consists of:

- Michael T Gastner, Vivien Seguy, and Pratyush More. Fast flow-based algorithm for creating density-equalizing map projections. *Proceedings of the National Academy of Sciences*, 115(10):E2156–E2164, 2018

Chapter 3 consists of:

- Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-scale optimal transport and mapping estimation. In *Proceedings of the International Conference in Learning Representations*, 2018

Chapter 4 consists of:

- Elsa Cazelles, Vivien Seguy, Jérémie Bigot, Marco Cuturi, and Nicolas Papadakis. Geodesic PCA versus log-PCA of histograms in the Wasserstein space. *SIAM Journal on Scientific Computing*, 40(2):B429–B456, 2018. doi: 10.1137/17M1143459. URL https://doi.org/10.1137/17M1143459

Chapter 5 consists of:

- Vivien Seguy and Marco Cuturi. Principal geodesic analysis for probability measures under the optimal transport metric. In *Advances in Neural Information Processing Systems*, pages 3312–3320, 2015

Contents

List of Figures

List of Tables

To my family and friends.

Chapter 1

Measure Transport in Machine Learning

1.1 Introduction

The modern approach to machine learning [Vapnik, 1999, 2013, Friedman et al., 2001] casts a learning problem in a probabilistic framework. In that framework, one is given a data set, usually referred to as *training set*, and it is assumed that each sample has been sampled independently from some underlying distribution. In some settings, one may have several data sets at hand, each of them assumed to be sampled from a different underlying probability distribution. For instance in domain adaptation, one is given a labeled source data set and one or several unlabeled target data sets whose distributions deviate from the source data set distribution. Similarly, in multi-task learning one is given several training sets each made of different features and/or labels from which we want to learn different tasks. In some cases, a data set can be itself made of probability measures. This happens when a given datum can be represented as a probability measure: histograms, grayscale images, bag of words, Fourier spectra etc., can all be seen as probability measures once they have been normalized.

This abundance of problems where several probability distributions come into play motivates the concept of *measure transport*. Measure transport refers to the concept of transporting one probability measure to another. The most natural way to transport a source measure μ supported on a space $\mathcal{X}$ to a target measure ν supported on a space $\mathcal{Y}$ is through a measurable mapping $f : \mathcal{X} \to \mathcal{Y}$. Reasoning in terms of random variables, with $X \sim \mu$ and $Y \sim \nu$, the mapping f is said to transport μ to ν whenever $f(X)$ and Y have the same distribution, denoted $f(X) \sim Y$ henceforth. Such *transport maps* can be useful in numerous machine learning and statistics applications. In generative modeling for instance, one can sample from a discrete target distribution by sampling from a Gaussian source measure and applying the mapping f to that sample, where f has been learned to satisfy $f(X) \sim Y$ approximately [Goodfellow et al., 2014, Li et al., 2015, Salimans et al., 2016, Li et al., 2017]. The same process is used in Bayesian statistics to sample efficiently from a target distribution without resorting to Monte-Carlo or sequential Monte-Carlo methods [Marzouk et al., 2016]. In domain adaptation, a transport map can also be used to align the source and target distributions in order to learn a classifier on the mapped source set which generalizes better to the target set [Courty et al., 2014].

Among the potentially many transport maps between two probability measures μ and ν, one particular transport map which minimizes the transportation cost with respect to some ground cost $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$ leads to the important concept of *optimal transport* (OT). The problem of finding a transport map minimizing the transportation cost was initially formulated by Monge [1781]. In the general case

however, the existence of a transport map between arbitrary probability measures is not guaranteed: this can for instance happen when μ is a Dirac measure but ν is not. The modern formulation of optimal transport was introduced by Kantorovich [1942] who relaxed the Monge problem by minimizing the transportation cost over the set of couplings rather than the set of transport maps. A coupling π between μ and ν is a joint probability distribution $\pi \in P(\mathcal{X} \times \mathcal{Y})$ whose marginals are equal to μ and ν respectively. Contrary to transport maps which may not exist, couplings always do. It suffices to consider for instance the independent coupling $\mu \times \nu$. The *Monge-Kantorovich* formulation is hence always feasible, and the existence of an optimal coupling is guaranteed under very general assumptions [Villani, 2008, Theorem 4.1]. Moreover, when $\mathcal{Y} = \mathcal{X}$ and the ground cost is a metric on $\mathcal{X}$, the optimal transport objective is a metric between probability distributions [Villani, 2003, Theorem 7.3]. This resulting *optimal transport metric* has been used actively and successfully in the recent years in order to train machine learning models using the OT metric as a loss [Frogner et al., 2015, Arjovsky et al., 2017, Gulrajani et al., 2017, Genevay et al., 2018].

In the present thesis, the concept of measure transport plays a central role. Chapter 2 addresses the fast computation of visually appealing geographic maps known as *cartograms*, by casting this problem as the computation of a transport map. Chapters 3, 4 and 5 then focus on *optimal transport*, both on its computational aspects (Chapter 3) and its methodological aspects (Chapter 4 and Chapter 5). Before presenting this research in the following chapters, this introductory chapter provides more details on the transport of measures as well as some background on optimal transport. A brief overview is given about how these concepts are used in the recent machine learning literature, and what are the current challenges associated. The detailed listing of the contributions achieved in this thesis is then provided at the end of this chapter.

1.2 Transport of Measures

In the introduction of this chapter, two ways of transporting one probability measure to another have been mentioned, either through a mapping or a coupling. In this section I provide a brief overview of each concept.

1.2.1 Transport maps

Consider a random variable X distributed according to $\mu \in P(\mathcal{X})$. Given a measurable map $f : \mathcal{X} \to \mathcal{Y}$, the measure image $f\#\mu$ is defined as the distribution of the random variable $f(X)$. Such maps can be of considerable interest in numerous fields. For instance, in Bayesian computation they can be used to sample from a target distribution ν by sampling from μ and applying a transport map f to that sample [Marzouk et al., 2016]. Transport maps can also be used to provide insightful visualization of statistical data: *cartograms* for instance are geographic maps where each region is rescaled proportionally to a given number. Fig. 1.1 shows (top) the deformation of a rectilinear grid and (below) the distorted topographic map. In machine learning, transport maps are learned and used as generators for generative modeling [Goodfellow et al., 2014, Arjovsky et al., 2017].

The concept of measure image is easy to understand on a discrete distribution: the measure image of $\mu = \sum_i a_i \delta_{\mathbf{x}_i}$ by a map f is obtained by moving each Dirac

[1] http://sedac.ciesin.columbia.edu/

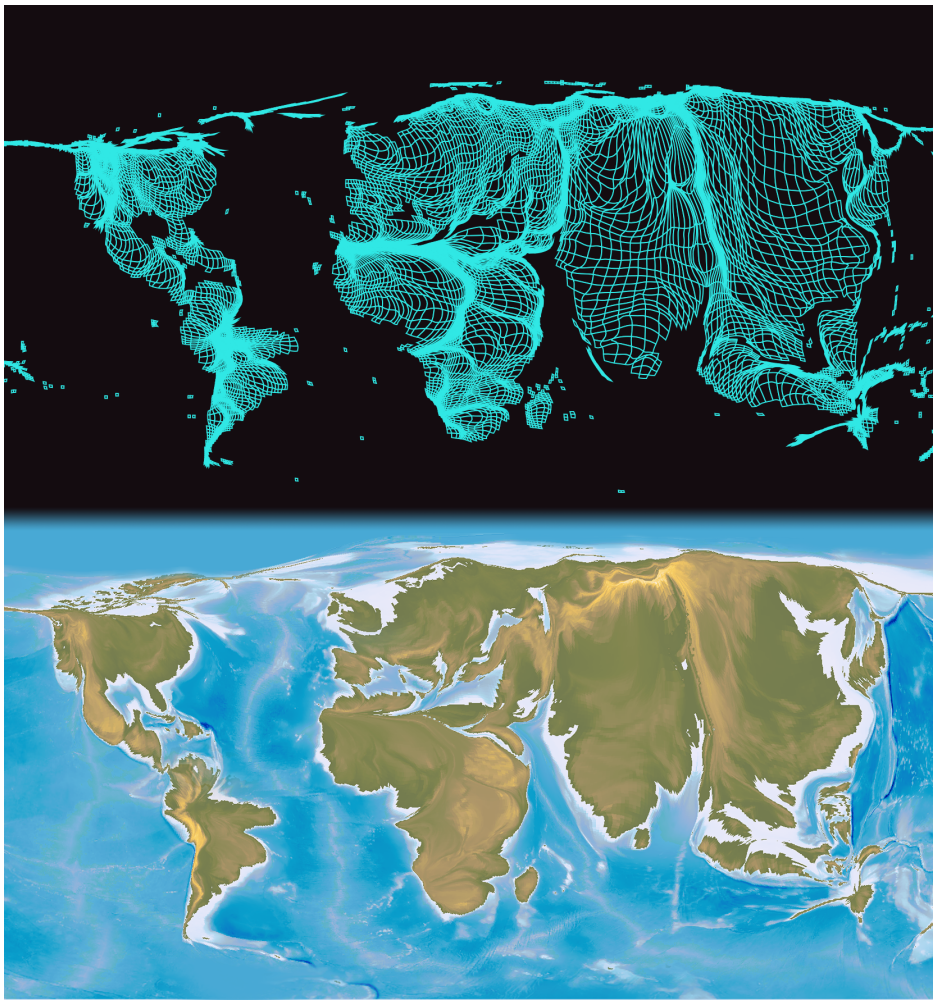

FIGURE 1.1: Population cartogram of the world obtained by computing a transport map between the gridded word population density taken from the Socioeconomic Data and Applications Center (SEDAC[1]) and a uniform probability measure. (Top) Visualization of the effect of the transport map on a regular grid and (Bottom) Topographic population cartogram resulting topographic map obtained by applying the transport map to the initial topographic map.

measure location $\mathbf{x}_i$ according to this map

$$f\#\mu \stackrel{\text{def.}}{=} \sum_i a_i \delta_{f(\mathbf{x}_i)}. \tag{1.2.1}$$

Here, the symbol # is usually called the *pushforward* operator. The general definition is the following.

Definition 1. *Let $\mu \in P(\mathcal{X})$ and $f : \mathcal{X} \to \mathcal{Y}$ a measurable map from $\mathcal{X}$ to $\mathcal{Y}$. The image measure of μ through f, denoted $f\#\mu$, is the probability distribution on $\mathcal{Y}$ defined by*

$$f\#\mu(B) = \mu(f^{-1}(B)). \tag{1.2.2}$$

for any measurable set $B \in \mathcal{Y}$.

Hence, we see that the image measure is well-defined as soon as f is a measurable

map, which is a fairly general assumption. The definition 1 is also equivalent to the fact that for any bounded continuous function g, we have

$$\int_{\mathcal{X}} g(f(\mathbf{x}))d\mu(\mathbf{x}) = \int_{\mathcal{Y}} g(\mathbf{y})d(f\#\mu)(\mathbf{y}). \tag{1.2.3}$$

If μ is absolutely continuous w.r.t. the Lebesgue measure, i.e. μ admits a density h_μ w.r.t. the Lebesgue measure, and if the map $f : \mathcal{X} \to \mathcal{X}$ is differentiable and injective, then the change of variable formula applied to Eq. (1.2.3) shows that $f\#\mu$ also admits a density $h_{f\#\mu}$ w.r.t. the Lebesgue measure, given by,

$$h_{f\#\mu}(\mathbf{y}) = \frac{h_\mu(f^{-1}(\mathbf{y}))}{|Df(f^{-1}(\mathbf{y}))|} \tag{1.2.4}$$

where $|Df(\cdot)|$ is the absolute value of the determinant of the Jacobian of f.

Consider two probability measures $\mu \in P(\mathcal{X})$ and $\nu \in P(\mathcal{Y})$, which we will often refer to as the source measure and target measure respectively. We say that f *pushes forward*, or *transports*, μ to ν when we have $f\#\mu = \nu$. In general, such a map may not exist. Consider for instance discrete probability measures such as $\mu = \delta_0$ and $\nu = \frac{1}{2}\delta_{-1} + \frac{1}{2}\delta_1$, then there is no transport map between μ and ν since a map cannot split the mass located in 0 to the weights of ν located at two distinct locations $\{-1, 1\}$. Nevertheless, when μ does not give mass to small sets, then the existence of such maps is guaranteed [Brenier, 1991]. This is in particular true when μ is absolutely continuous w.r.t. the Lebesgue measure. On the computational side, finding such maps between arbitrary probability measures supported on arbitrary spaces is a very challenging task.

Computational approaches

One approach to construct maps between probability measures supported on the same d-dimensional Euclidean space $\mathbb{R}^d$ (or on a d-dimensional manifold) is based on the integration of flows. Consider a family of probability densities $(\rho_t)_{t\in[0,T]}$ w.r.t. the Lebesgue measure $\mathcal{L}$ and assume there is a family of velocity fields $(v_t)_{t\in[0,T]}$ such that the *continuity equation*, also known as *mass conservation equation*, is verified

$$\frac{\partial \rho_t}{\partial t} + \nabla(v_t\rho_t) = 0. \tag{1.2.5}$$

Then defining the map f_T by integrating the following ordinary differential equation (ODE),

$$f_T(\mathbf{x}) = \mathbf{x} + \int_{[0,T]} v_t(f_t(\mathbf{x}))dt, \tag{1.2.6}$$

provides a map f_T which, up to some smoothness assumptions of (ρ_t) and (v_t), verifies the Jacobian equation $\rho_T(f_T(\mathbf{x})) = \frac{\rho_0(\mathbf{x})}{|Df_T(\mathbf{x})|}$, so that f_T pushes forward the probability measure $\mu \sim \rho_0\mathcal{L}$ to the probability measure $\nu \sim \rho_T\mathcal{L}$. One of the most general formulation, where the continuity equation is written with probability measures rather than densities and hence must be understood in the sense of distributions, can be found in [Ambrosio et al., 2006].

Theorem 1. *([Ambrosio et al., 2006], Proposition 8.1.8) Let $(\mu_t)_{[0,T]}$ be a (narrowly continuous) family of probability measures such that the continuity equation*

$$\frac{\partial \mu_t}{\partial t} + \nabla(v_t \mu_t) = 0 \tag{1.2.7}$$

holds w.r.t. a family of locally Lipschitz velocity fields $(v_t)_{[0,T]}$ satisfying some technical assumptions described in details in [Ambrosio et al., 2006, Proposition 8.1.8]. Then the ODE system $\frac{d}{dt} f_t(\mathbf{x}) = v_t(f_t(\mathbf{x}))$, $f_0(\mathbf{x}) = \mathbf{x}$, admits a globally defined solution (f_t) on $[0,T]$ such that

$$\mu_t = f_t \# \mu_0. \tag{1.2.8}$$

These considerations can provide an efficient way of computing a transport map between μ and ν, by constructing a family (η_t) such that $\eta_0 = \mu$ and $\eta_T = \nu$, and a family of velocity fields (v_t) which verify the continuity equation, and then integrating these velocity fields on $[0,T]$. This was the approach used by [Moser, 1965, Dacorogna and Moser, 1990] to prove constructively the existence of a mapping verifying the Jacobian equation (1.2.4). This is also the approach we use in Chapter 2 to build an efficient algorithm for the computation of cartograms [Gastner et al., 2018].

Another simple construction of a transport map between two probability measures supported on $\mathbb{R}^d$ is obtained through the *Knothe-Rosenblatt rearrangement*, which was proposed independently by Knothe et al. [1957] and Rosenblatt [1952]. When one of the probability measures is absolutely continuous w.r.t. the Lebesgue measure, the Knothe-Rosenblatt rearrangement builds a transport map whose Jacobian is triangular with non-negative eigenvalues. Such properties can be leveraged in order to compute efficiently marginals of the target distribution [Marzouk et al., 2016]. Interestingly, a link with optimal transport was made by Carlier et al. [2010]: the Knothe-Rosenblatt rearrangement can be seen as the limit of solutions of the optimal transport problem (i.e. optimal transport plans) when using a weighted squared Euclidean norm as ground cost.

The two approaches described above work well in practice for measures supported on small-dimensional Euclidean spaces, typically $\mathbb{R}^d$ with $d \leqslant 3$. For the flow-based approach, this is because it is necessary to integrate an ODE on each location of a discretized support of the source measure. Since a fine-grained discretized support is necessary to approximate the solution of the continuity equation sufficiently well, its size grows exponentially with the dimension of the space and makes the method intractable. This is also true with the Knothe-Rosenblatt rearrangement which requires to finely discretize the support of the source and target measures in order to compute their marginal distributions.

The previous discussion highlighted that there are two major challenges when one wants to compute a transport map between a source measure and a target measure,

- there is no known efficient computational method for arbitrary measures supported on a high-dimensional space,
- when the source measure is discrete, a transport map may not exist.

Regarding the first point, most recent approaches cast the problem of finding a transport map as an optimization problem, where one seeks to minimize a divergence/discrepancy/metric between the image measure $f\#\mu$ and the target probability measure ν w.r.t. some divergence/discrepancy/metric D between probability measures

$$\min_{f \in \mathcal{H}} \; D\left(f\#\mu | \nu\right). \tag{1.2.9}$$

This approach has become very popular recently in the machine learning community as a way to learn generative models [Goodfellow et al., 2014]. In that setting, μ is chosen as a probability measure from which one can easily sample, typically a Gaussian, the transport map is parameterized as a deep neural network, and the target measure ν is a discrete data set, such as a collection of images or texts. Many divergences have been proposed over the past few years in order to learn such models, among which the Jensen-Shannon divergence [Goodfellow et al., 2014, Salimans et al., 2016], the maximum-mean discrepancy (MMD) [Li et al., 2015, 2017] or the Wasserstein distance [Arjovsky et al., 2017, Gulrajani et al., 2017, Genevay et al., 2018]. Computing these divergences can become quickly intractable when the size of the support of the target measure is large. Moreover, since μ is usually chosen as a continuous probability measure, and so is $f\#\mu$ when f is a non-degenerate neural network, a naive approach would require to sample a large number of points from $f\#\mu$ in order to get a good approximation of $D\left(f\#\mu|\nu\right)$, making the approach computationally inefficient. In order to make the computation of D tractable, Goodfellow et al. [2014] showed that the Jensen-Shannon divergence can be obtained through a maximization problem where the variable belongs to the space of continuous functions [Goodfellow et al., 2014, Theorem 1.]

$$2 \cdot JS(\mu|\nu) - \log(4) = \max_{g \in \mathcal{C}(\mathcal{X})} \mathbb{E}_{X\sim\mu}\left[\log g(X)\right] + \mathbb{E}_{X\sim\nu}\left[\log(1-g(X))\right]. \tag{1.2.10}$$

where they parameterize the variable g as a deep neural network, making the optimization problem (1.2.9) a min-max problem

$$\min_{f\in\mathcal{H}} \max_{g\in\mathcal{C}(\mathcal{X})} \mathbb{E}_{X\sim\mu}\left[\log g(X)\right] + \mathbb{E}_{X\sim\nu}\left[\log(1-g(f(X)))\right]. \tag{1.2.11}$$

for which a saddle point can be found using an adversarial training procedure [Goodfellow et al., 2014]. Several similar formulae can be derived for the maximum-mean discrepancy [Li et al., 2015], the 1-Wasserstein distance [Arjovsky et al., 2017] and several other divergences [Nowozin et al., 2016] have been derived, resulting in a similar min-max formulation for learning generators. Alternatively, the expression $D\left(f\#\mu|\nu\right)$ can be approximated by computing D several times on mini-batches sampled from $\mu \times \nu$, i.e. substituting $D\left(f\#\mu|\nu\right)$ by,

$$\mathbb{E}_{\substack{(\mathbf{x}_1,\cdots,\mathbf{x}_p)\sim\mu^{(p)} \\ (\mathbf{y}_1,\cdots,\mathbf{y}_p)\sim\nu^{(p)}}} \left[D\left(\sum_{i=1}^{p} \delta_{f(\mathbf{x}_i)}, \sum_{i=1}^{p} \delta_{\mathbf{y}_i} \right) \right], \tag{1.2.12}$$

where $\mu^{(p)} = \mu \times \cdots \mu$ (p times) and $\nu^{(p)} = \nu \times \cdots \nu$ (p times). In practice, the expectation Eq. (1.2.12) is approximated as an average. This approach was used for instance for training generative models w.r.t. the energy distance [Bellemare et al., 2017], the maximum-mean discrepancy [Li et al., 2017] or the entropy-regularized optimal transport metric [Genevay et al., 2018]. Both the adversarial and mini-batch approaches assume that the target probability measure ν is discrete, or at least that we can sample from ν. When only the (possibly unnormalized) target density is available, then the formula for the pushforward density Eq. (1.2.4) can be used in order to match the pushforward density and the target density through some optimization procedures [Marzouk et al., 2016].

The fact that a transport map does not exist when the source measure μ is discrete is often not a fundamental problem, since statisticians and machine learners assume that μ may have been generated by sampling i.i.d. from some underlying

continuous measure. Hence, their goal is rather to find a transport map between the underlying continuous distributions which generated μ and ν. Accordingly, training a generative model f which reaches the minimum of $D\left(f\#\mu|\nu\right)$ would be of no interest, since having $f\#\mu = \nu$ exactly would only generate samples from the discrete target distribution. Yet, if the concept of measure transport between discrete probability measures is still needed, one may instead use the notion of coupling.

1.2.2 Couplings

A coupling between two probability measures $\mu \in P(\mathcal{X})$ and $\nu \in P(\mathcal{Y})$ is a probability distribution over the product space $\mathcal{X} \times \mathcal{Y}$ whose marginals are equal to μ and ν respectively.

Definition 2. *Let $\mu \in P(\mathcal{X})$ and $\nu \in P(\mathcal{Y})$. A coupling $\pi \in P(\mathcal{X} \times \mathcal{Y})$ between μ and ν is a joint probability distribution on $\mathcal{X} \times \mathcal{Y}$ whose first and second marginals are equal to μ and ν respectively, i.e.*

$$\pi(B_{\mathcal{X}} \times \mathcal{Y}) = \mu(B_{\mathcal{X}}), \quad \pi(\mathcal{X} \times B_{\mathcal{Y}}) = \nu(B_{\mathcal{Y}}). \tag{1.2.13}$$

for any measurable sets $B_{\mathcal{X}} \subset \mathcal{X}$ and $B_{\mathcal{Y}} \subset \mathcal{Y}$. We denote $\Pi(\mu, \nu)$ the set of couplings between μ and ν.

The latter definition can be understood equivalently as

$$\int_{\mathcal{Y}} d\pi(\mathbf{x}, \mathbf{y}) = d\mu(\mathbf{x}), \quad \int_{\mathcal{X}} d\pi(\mathbf{x}, \mathbf{y}) = d\nu(\mathbf{y}). \tag{1.2.14}$$

Contrary to transport maps, couplings always exist between arbitrary probability measures μ and ν: for instance, the product measure $\mu \times \nu$ defined as $\mu \times \nu(B_{\mathcal{X}} \times B_{\mathcal{Y}}) \overset{\text{def.}}{=} \mu(B_{\mathcal{X}})\nu(B_{\mathcal{Y}})$ belongs to $P(\mathcal{X} \times \mathcal{Y})$ and verifies trivially the marginal conditions (1.2.13). Other couplings can be constructed following some simple procedures such as the *North-West corner rule* (see for instance [Peyré et al., 2017], Section 3.4.2). The set of couplings verifying the two marginal constraints is usually referred to as the transport polytope, which we denote $\Pi(\mu, \nu)$ in the sequel. Compared to a transport map which sends all the mass contained at the location $\mathbf{x}$ to $f(\mathbf{x})$, a coupling allows the mass to be split: it can be seen as a weighted one-to-many map where the mass at $\mathbf{x}$ is sent to all locations $\mathbf{y}$ such that $d\pi(\mathbf{x}, \mathbf{y})$ is positive.

Couplings play an important role in the modern optimal transport theory initiated by Kantorovich [1942]. While the Monge problem Eq. (3.2.1) of finding the transport map which minimizes a total mass transportation cost can be infeasible, or the minimum is not achieved, the problem Eq. (3.2.2) of minimizing the transportation cost over couplings is always feasible and the minimum is achieved under very general assumptions on the spaces $\mathcal{X}$ and $\mathcal{Y}$ and the ground cost c. In some specific cases the two problems become equivalent: an important result by Brenier [1991] shows that when the source measure μ is absolutely continuous w.r.t. the Lebesgue measure, and the ground cost is the squared Euclidean norm, the solution to the relaxed OT problem (3.2.2) is identical to the solution of the Monge problem Eq. (3.2.1). Hence, in the latter case, solving the optimal transport problem can be a way to find a unique transport map. Such optimal maps can be useful in numerous applications. We provide in the following section more details about the optimal transport theory, which is playing an increasing role in machine learning and statistics, and is the main topic of Chapters 3 to 5.

1.3 Optimal Transport

1.3.1 Background on optimal transport

Given two random variables X and Y distributed according to a source probability measure $\mu \in P(\mathcal{X})$ and a target probability measure $\nu \in P(\mathcal{Y})$ respectively, finding a transport map between X and Y refers to the problem of finding a measurable map which verifies $f(X) \sim Y$, or equivalently $f\#\mu = \nu$. Among the potentially several maps verifying $f(X) \sim Y$, it may be of interest to choose a specific one which satisfies some notion of optimality. Given a ground cost $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$, we have seen in earlier discussion that a natural optimality criterion is to ask to the map f to minimize the total transportation cost. This was the original formulation introduced by Monge [1781] in his famous *Mémoire sur la theorie des déblais et des remblais*.

The Monge problem

Given two probability measures $\mu \in P(\mathcal{X})$ and $\nu \in P(\mathcal{Y})$ and a ground cost $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$, the Monge problem consists in finding a transport map which minimizes the total transportation cost

$$\inf_{f \in \mathcal{M}(\mathcal{X},\mathcal{Y})} \int_{\mathcal{X}} c(\mathbf{x}, f(\mathbf{x}))d\mu(\mathbf{x}) \quad \text{subject to } f\#\mu = \nu, \tag{1.3.1}$$

where $\mathcal{M}(\mathcal{X},\mathcal{Y})$ is the set of measurable maps from $\mathcal{X}$ to $\mathcal{Y}$. In terms of random variables, given $X \sim \mu$ and $Y \sim \nu$, this can be written equivalently as

$$\inf_{f \in \mathcal{M}(\mathcal{X},\mathcal{Y})} \mathbb{E}_{X\sim\mu}\left[c(X, f(X))\right] \quad \text{subject to } f(X) \sim Y. \tag{1.3.2}$$

Monge was originally interested in the specific case which uses the Euclidean distance as the ground cost $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$. In the sequel, we shall refer to the Monge problem as (1.3.1) (or Eq. (1.3.2) equivalently) for any ground cost c, and we will call a minimizer of Problem (1.3.1) an *optimal map* or a *Monge map*. As stated previously, such a problem might be infeasible, i.e. there is no measurable map verifying the image measure constraint $f\#\mu = \nu$. For instance, when μ is a discrete probability measure, there is no guarantee of existence of a transport map. Moreover, even assuming that Eq. (1.3.1) is feasible, the infimum may not be achieved, so that a minimizer cannot be found. This is due to the fact that the set of transport maps is not (sequentially) compact. However, when the ground cost is the squared Euclidean distance $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$ and when μ does not give mass to small sets, then Brenier [1991] showed that the infimum of (1.3.1) is achieved, i.e. there exists a transport map which minimizes the total transportation cost. This result was then generalized to more general ground costs by several authors [Bernard and Buffoni, 2004, Champion et al., 2011, Ambrosio and Gigli, 2013, Feyel and Üstünel, 2004].

Even when the existence of an optimal map is guaranteed theoretically, its computation is very challenging: we have seen in the previous section that the easier task of finding a transport map, i.e. a feasible map of Problem (1.3.1), is already a challenging task. Most methods for finding exact optimal transport maps between two absolutely continuous (w.r.t. the Lebesgue measure on $\mathbb{R}^d$ or a reference measure on a manifold) probability measures μ and ν are based on a flow formulation. Probably one of the most famous results in this regard is the Benamou-Brenier formulation [Benamou and Brenier, 2000] which showed that the optimal transport

objective w.r.t. the squared Euclidean norm as ground cost can be written as

$$\min_{(\rho_t,v_t)} \int_{[0,1]} \rho_t \|v_t\|_2^2 dt \quad subject \;\; to \;\; \{\frac{\partial \rho_t}{\partial t} + \nabla(v_t\rho_t) = 0,\; \rho_0\mathcal{L} = \mu,\; \rho_1\mathcal{L} = \nu\}. \tag{1.3.3}$$

Similar flow formulations were also proved for other ground costs [Evans and Gangbo, 1999]. Benamou and Brenier [2000] proposed to solve this problem by an augmented Lagrangian algorithm. Other approaches to minimize the objective of Problem (1.3.3) such as proximal methods were also proposed [Papadakis et al., 2014]. All algorithms involve discretization of the space and time. When solving the problem on a manifold, a graph-based approach can be considered [Solomon et al., 2014]. The discretization of the space makes these aproaches however intractable for measures supported on a space of dimension $d > 3$.

In order to formulate an optimal transport problem which is always feasible and has its minimum achieved, Kantorovich [1942] relaxed Problem (1.3.1) by minimizing the transport cost over couplings rather than the set of transport maps.

Kantorovitch relaxation

The Kantorovich relaxation consists in allowing the mass located at $\mathbf{x}$ in the support of the source measure μ to be sent to several locations $\mathbf{y}$ in the support of the target measure ν, contrary to transport maps where all the mass in $\mathbf{x}$ is sent to $\mathbf{y}$. This can be formulated in terms of a coupling $\pi \in P(\mathcal{X} \times \mathcal{Y})$ between μ and ν, where $d\pi(\mathbf{x}, \mathbf{y})$ is the quantity of mass sent from $\mathbf{x}$ to $\mathbf{y}$. This results in the following minimization problem

$$\min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(\mathbf{x}, \mathbf{y}) d\pi(\mathbf{x}, \mathbf{y}) \tag{1.3.4}$$

which is usually referred to as the Monge-Kantorovich (MK) problem, or more simply the optimal transport (OT) problem. In terms of random variables, given $X \sim \mu$ and $Y \sim \nu$, this can be written equivalently as

$$\min_{\pi \in \Pi(\mu,\nu)} \mathbb{E}_{(X,Y)\sim\pi} \left[c(X, Y)\right]. \tag{1.3.5}$$

Here, we have written *min* instead of *inf* because under the rather general assumptions that $\mathcal{X}$ and $\mathcal{Y}$ are Polish spaces and c is lower semi-continuous, the infimum of the objective function is achieved. This follows essentially from the fact that the set of couplings $\Pi(\mu, \nu)$ is tight, and hence, by Prokhorov's theorem that $\Pi(\mu, \nu)$ is compact w.r.t. the topology of weak convergence (see [Villani, 2003, Chapter 1] for details). When $\mathcal{X} = \mathcal{Y}$ is a metric space and the ground cost is taken as the p-th power of a metric d on $\mathcal{X}$, i.e. $c(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, \mathbf{y})^p$, then the p-Wasserstein distance defined as,

$$W_p(\mu, \nu) \overset{\text{def.}}{=} OT(\mu, \nu)^{1/p} \overset{\text{def.}}{=} \left(\min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} d(\mathbf{x}, \mathbf{y})^p d\pi(\mathbf{x}, \mathbf{y})\right)^{1/p}, \tag{1.3.6}$$

is a metric on the space of probability distributions $P(\mathcal{X})$. In particular, it metrizes the weak convergence of probability measures [Villani, 2003, Chapter 7].

Another important property of the MK problem is that the objective function is linear in the variable π, and this is also true for the marginal contraints, as a consequence of the linearity of the Lebesgue integral. Moreover, Kantorovich [1942]

proved that the dual of the OT problem is also a linear program

$$\max_{u\in L^1(d\mu,\mathcal{X}),\, v\in L^1(d\nu,\mathcal{Y})} \int_{\mathcal{X}} u(\mathbf{x})d\mu(\mathbf{x}) + \int_{\mathcal{Y}} v(\mathbf{y})d\nu(\mathbf{y}), \tag{1.3.7}$$

$$\text{s.t. } u(\mathbf{x}) + v(\mathbf{y}) \leqslant c(\mathbf{x},\mathbf{y}) \; for \; (\mu\times\nu) - almost \; all \; (\mathbf{x},\mathbf{y}).$$

The MK problem has hence greatly contributed to motivating advances in the field of linear programming [Kantorovich, 1942, Dantzig, 1951], which studies optimization problems with a linear objective and linear equality or inequality constraints. This thorough development of the linear programming theory has led to several algorithms to solve the discrete version of the OT problem (1.3.4)

$$\min_{T\in U(\mathbf{a},\mathbf{b})} \langle C, T\rangle, \quad U(\mathbf{a},\mathbf{b}) = \{T\in\mathbb{R}_+^{m\times n}, T\mathbf{1}_n = \mathbf{a},\; T^T\mathbf{1}_m = \mathbf{b}\}, \tag{1.3.8}$$

where $C \in \mathbb{R}_+^{m\times n}$ is the ground cost matrix and $U(\mathbf{a},\mathbf{b})$ is a bounded polyhedron usually referred to as the *transport polytope* of the two vectors $\mathbf{a}$ and $\mathbf{b}$ which belong to the simplex. Arguably one of the most important breakthroughs in linear programming is the development of the simplex algorithm [Dantzig, 1951]. Since a solution is known to be on an extremal point (vertex) of the transport polytope, which is a bounded polyhedron, a solution is sought by starting from an extremal point, checking if it is an optimal solution, then moving to a better extremal point iteratively in case it is not an optimal solution. Several other algorithms have also been proposed to solve the discrete OT problem (1.3.8), such as dual ascent-based methods [Kuhn, 1955] or the auction algorithm [Bertsekas, 1981]. These algorithms are able to solve the discrete OT problem exactly. However, they all have a super-cubic complexity in the size of the support of the discrete measures μ and ν, denoted m and n respectively in Problem (1.3.8), so that such algorithms are only tractable for measures supported on a few thousand points at most.

To address this computational challenge, several works have shown that solving a regularized version of the OT problem (1.3.8) can be easier [Cuturi, 2013, Genevay et al., 2016, Blondel et al., 2018, Altschuler et al., 2017]. We give more details on regularized OT in the next paragraph.

Regularized optimal transport

The idea of regularizing the OT problem (1.3.4) with an entropy term on the primal variable (the transport plan) dates back to the work of [Schrödinger, 1931], and was recently put back into light to the machine learning community by [Cuturi, 2013]. Considering the discrete setting, the entropy-regularized OT problem modifies the OT problem objective as follows

$$\min_{T\in U(\mathbf{a},\mathbf{b})} \langle C, T\rangle + \varepsilon \sum_{i}^{m}\sum_{j}^{n} T_{ij}\log(T_{ij}). \tag{1.3.9}$$

where ε is a hyperparameter controlling the amount of regularization. Denoting $OT(\mathbf{a},\mathbf{b})$ and $OT_\varepsilon(\mathbf{a},\mathbf{b})$ the objective values of the unregularized and regularized OT problems respectively, some bounds between these two values [Cominetti and San Martín, 1994, Cuturi, 2013, Blondel et al., 2018] show that when $\varepsilon \to 0$, then $OT_\varepsilon(\mathbf{a},\mathbf{b}) \to OT(\mathbf{a},\mathbf{b})$. This is also true for the solution T_ε of the regularized OT problem which converges to the solution T of the unregularized OT problem at an exponential rate [Cominetti and San Martín, 1994]. Other regularizers, for instance

the squared L^2 norm $\sum_{ij} T_{ij}^2$, have also been considered recently by several authors [Dessein et al., 2016, Blondel et al., 2018, Seguy et al., 2018, Muzellec et al., 2017].

Cuturi [2013] showed that the entropy-regularized OT problem (1.3.9) can be solved by using a block-coordinate descent on the dual objective of the regularized OT problem, where each block corresponds to one of the two dual variables. This provides an algorithm similar to the Sinkhorn-Knopp algorithm [Sinkhorn and Knopp, 1967] which iteratively scales the Gibbs kernel $\exp(-C/\varepsilon)$ to have the right marginals $\mathbf{a}$ and $\mathbf{b}$. Each scaling operation involves a matrix product and an elementwise division which can be parallelized, allowing the computation of many regularized OT distances in parallel using GPUs. Variations of the Sinkhorn algorithm have also been proposed in order to provide a more accurate solution given a computational budget [Altschuler et al., 2017], stabilize the algorithm [Chizat et al., 2016b], or consider an unbalanced version of the regularized OT problem [Chizat et al., 2018].

Recent advances in the computation of OT, regularized OT, as well as the development of deep learning, has lead recently to a more widespread use of optimal transport in machine learning applications. I review some of the main trends, before listing the contributions to the field made by my co-workers and I and presented in the following chapters.

1.3.2 Optimal transport in machine learning

Optimal transport was introduced into the machine learning community by Rubner et al. [2000] under the name of *earth mover distance* (EMD), where they showed that the OT distance is a relevant metric to compare histograms and perform tasks such as image retrieval based on image's color histograms. However, the complexity of the simplex algorithm has limited the widespread adoption of optimal transport until the work of Cuturi [2013], who proposed a fast algorithm to compute entropy-regularized OT. Several recent papers also showed that OT can be a powerful loss to train machine learning models [Frogner et al., 2015, Arjovsky et al., 2017, Gulrajani et al., 2017, Genevay et al., 2018].

An efficient loss for learning models

A typical supervised-learning task is to learn a classifier $f : \mathcal{X} \to \mathcal{Y} = \{1, \cdots, K\}$ from a labeled training set $\{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_N, \mathbf{y}_N)\}$ so that, given a new sample $\mathbf{x}$, the function f can infer the relevant label $f(\mathbf{x})$. For that purpose, one usually defines loss function $L : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ such that $L(\mathbf{x}, f(\mathbf{x}), \mathbf{y})$ quantifies the error of having the sample misclassified, i.e. $f(\mathbf{x}) \neq \mathbf{y}$. Ideally, we hence want to minimize the total loss $\mathcal{R}$ w.r.t. the underlying distribution of data, usually referred to as *risk*, over a family of functions $f \in \mathcal{H}$

$$\min_{f \in \mathcal{H}} \quad \mathcal{R}(f) \overset{\text{def.}}{=} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mu}[L(\mathbf{x}, f(\mathbf{x}), \mathbf{y})] \tag{1.3.10}$$

where μ is the underlying distribution from which data are assumed to be sampled, and $\mathcal{H}$ is a chosen family of functions, such as linear functions, kernel combinations or neural networks. As we only have access to the empirical distribution $\mu^N = \frac{1}{N}\sum_{i=1}^N \delta_{(\mathbf{x}_i,\mathbf{y}_i)}$, this is usually achieved by minimizing the *empirical risk* $\mathcal{R}^N$

$$\mathcal{R}^N(f) \overset{\text{def.}}{=} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mu^N}[L(\mathbf{x}, f(\mathbf{x}), \mathbf{y})] = \frac{1}{N}\sum_{i=1}^N L(\mathbf{x}_i, f(\mathbf{x}_i), \mathbf{y}_i) \tag{1.3.11}$$

hoping that a minimizer of the empirical risk may provide a low risk. The ability of the minimizer f_N^* of the empirical risk to provide a low risk w.r.t. the underlying distribution is called generalization. In the limit $N \to \infty$ of an infinite size training set, statistical learning theory provides guarantees, under some assumptions on the family $\mathcal{H}$, that this learning procedure is consistent [Vapnik, 1999], i.e. $R(f_N^*) \to R(f^*)$ where f^* is a minimizer of the risk $\mathcal{R}$. However, this asymptotic result is not very practical since we usually want to learn from a single and static data set $\mathcal{D}$. The generalization ability of the learning procedure will depend heavily on the choice of the loss function L, the family of functions $\mathcal{H}$ and some regularizer $r(f)$ added to the empirical risk.

In regression, when $\mathcal{Y} = \mathbb{R}$, a standard loss function is the squared Euclidean norm $L(\mathbf{x}, f(\mathbf{x}), \mathbf{y}) = \|\mathbf{y} - f(\mathbf{x})\|^2$. Assuming a generative process of the form $\mathbf{y} = f(\mathbf{x}) + \varepsilon$ where ε is an isotropic Gaussian noise with unit variance, the conditional distribution μ_f corresponding to that generative process has the closed form $\mu_f(\mathbf{y}|\mathbf{x}) = \frac{1}{\sqrt{2\pi}} e^{-\|\mathbf{y}-\mathbf{x}\|^2/2}$, so that minimizing the empirical risk becomes equivalent to maximizing the conditional log-likelihood of the training set

$$\sum_{i=1}^{N} \log\left(\mu_f(\mathbf{y}_i|\mathbf{x}_i)\right) = -\sum_{i=1}^{N} \left(\|\mathbf{y}_i - \mathbf{x}_i\|^2/2 + \frac{1}{2}\log(2\pi) \right). \tag{1.3.12}$$

In the same way, in classification where $\mathcal{Y} = \{1, \cdots, K\}$, a typical loss function used for instance in logistic regression is $L(\mathbf{x}, f(\mathbf{x}), \mathbf{y})] = \log\left(\frac{e^{f_{\mathbf{y}}(\mathbf{x})}}{\sum_{j=1}^{K} f_j(\mathbf{x})}\right)$, where f_j is the j^{th} component of the vector-valued classifier f. Assuming that the conditional label generation follows the categorical distribution $\mu_f(\mathbf{y} = k|\mathbf{x}) = \frac{e^{f_k(\mathbf{x})}}{\sum_{j=1}^{K} f_j(\mathbf{x})}$, the empirical risk is also equal to the conditional log-likelihood of the training set. In some cases, one may be willing to parameterize the joint distribution $\mu_f(\mathbf{x}, \mathbf{y})$ and maximize the log-likelihood of the data

$$\sum_{i=1}^{N} \log\left(\mu_f(\mathbf{x}_i, \mathbf{y}_i)\right). \tag{1.3.13}$$

Working with some parameterization of the joint-distribution rather than the conditional distribution can be useful for instance in semi-supervised learning [Ng and Jordan, 2002, Kingma et al., 2014].

This discussion shows the equivalence between minimizing the loss function on a training set $\mathcal{D}$ and the well-known process in statistics of maximizing the likelihood of this training set w.r.t. some parameterized distribution μ_f. Another interesting view-point is to also link the empirical risk with the concept of divergence/discrepancy/metric between probability distributions. Indeed, defining the Kullback-Leibler divergence [Kullback, 1997] between two probability measures μ and ν as,

$$\mathrm{KL}(\mu|\nu) \overset{\text{def.}}{=} \int \log \frac{d\mu}{d\nu} d\mu, \tag{1.3.14}$$

and noticing that $\log\left(\mu_f(\mathbf{y}_i|\mathbf{x}_i)\right) = \mathrm{KL}(\delta_{\mathbf{y}_i}|\mu_f(\cdot|\mathbf{x}_i))$, we see that the conditional log-likelihood of the training set in Eq. (1.3.12) can be rewritten as the sum of the Kullback-Leibler divergence between the conditional distributions over the label space,

$$\sum_{i=1}^{N} \mathrm{KL}(\delta_{\mathbf{y}_i}|\mu_f(\cdot|\mathbf{x}_i)). \tag{1.3.15}$$

Similarly, the joint log-likelihood of the training set can be rewritten as

$$\mathrm{KL}\left(\frac{1}{N}\sum_{i=1}^{N}\delta_{(\mathbf{x}_i,\mathbf{y}_i)}|\mu_f\right), \tag{1.3.16}$$

and hence minimizing the empirical risk becomes equivalent to minimizing the Kullback-Leibler divergence $\mathrm{KL}(\mu^N|\mu_f)$ between the empirical distribution μ^N and the parameterized distribution μ_f. This connection shows that by minimizing the empirical risk, we are actually trying to fit a parameterized distribution to an empirical distribution w.r.t. the Kullback-Leibler divergence. The Kullback-Leibler divergence is only one of the many well-known divergences between probability measures. In particular it is a special case of the family of *f-divergences* [Csiszár et al., 2004]. One may also consider minimizing for instance the squared L^2 distance $\|\mu^N - \mu_f\|_2^2$, the Hellinger distance $\|\sqrt{\mu^N} - \sqrt{\mu_f}\|_2$, the total variation distance or one particular metric inside the family of integral-probability metrics [Müller, 1997]. When choosing a specific divergence/discrepancy/metric, the following considerations come into play:

- the computational cost of evaluating the divergence and its gradient w.r.t. one of the input distribution,
- its smoothness and convexity properties,
- its ability to compare two distributions.

Choosing one divergence over another may have important effects on the minimizers of the empirical risk. The challenge is to choose a relevant metric for which the minimizer of the empirical risk provides a low risk w.r.t. the underlying distribution, i.e. *generalizes* well to unseen samples.

Numerous recent works have highlighted the power of the optimal transport metric to train machine learning models. One of the most recent and striking successes of using OT as a loss was made in the context of generative modeling [Arjovsky et al., 2017], where one wants to learn to generate samples which look as real as the samples from a training set. Current state-of-the-art approaches such as variational auto-encoders [Kingma and Welling, 2013], generative adversarial networks [Goodfellow et al., 2014] or generative moment matching networks [Li et al., 2015] rely on finding a map, often parameterized as a deep neural network, which can transport approximately a source distribution μ, from which it is simple to sample, to the discrete target distribution ν corresponding to the target set. Sampling from the generative model boils down to drawing a sample from μ and applying the map f. The idea of Arjovsky et al. [2017], coined Wasserstein GAN, is to use the 1-Wasserstein distance, whose dual formulation can be computed with stochastic gradient methods, in order to fit the generative distribution $f\#\mu$ to the taget distribution ν by solving

$$\min_{\theta} W_1(f_\theta\#\mu,\nu) = \min_{\theta}\max_{g\in \mathrm{Lip}_1(\mathcal{X})} \mathbb{E}_{X\sim\mu}\left[g(f_\theta(X))\right] - \mathbb{E}_{Y\sim\nu}\left[g(Y)\right] \tag{1.3.17}$$

where θ are the neural network parameters (the generator function) and g is the dual variable, also referred to as the *critic* function, which must be a Lipschitz function and is also parameterized in practice as a deep neural network. Different strategies to enforce the critic neural network to be Lipschitz have been proposed such as weight-clipping [Arjovsky et al., 2017] or gradient penalization [Gulrajani et al.,

2017]. The superior results obtained by training a generative model with a Wasserstein loss rather than the Jensen-Shannon divergence originally proposed in the GAN literature [Goodfellow et al., 2014] illustrates the power of Wasserstein distances to fit a parameterized distribution to a target distribution.

In the context of domain adaptation, Courty et al. [2017a] proposed to learn a classifier f on an unlabeled target data set $\mathcal{T} = (\mathbf{z}_1, \cdots, \mathbf{z}_N)$ set by fitting the discrete distribution $\frac{1}{N}\sum_i \delta_{\mathbf{z}_i, f(\mathbf{z}_i)}$ to the training set discrete distribution $\frac{1}{N}\sum_i \delta_{\mathbf{x}_i, \mathbf{y}_i}$ by solving

$$\min_{f \in \mathcal{H}} OT\left(\frac{1}{N}\sum_i \delta_{\mathbf{z}_i, f(\mathbf{z}_i)}, \frac{1}{N}\sum_i \delta_{\mathbf{x}_i, \mathbf{y}_i}\right). \tag{1.3.18}$$

An interesting link can be made with the generative model view-point by considering that (1.3.18) boils down to training a *generator* f which is conditioned on $\mathbf{z}$ and outputs a deterministic label value. Hence, Eq. (1.3.18) can be seen as learning a conditional generative model [Mirza and Osindero, 2014, Isola et al., 2017] w.r.t. a Wasserstein loss. Contrary to Wasserstein GAN [Arjovsky et al., 2017], Courty et al. [2017a] solved the OT problem with the network simplex algorithm and were hence limited in the size of the source and target data sets.

A regularized Wasserstein loss has also been proposed as a substitute to the cross-entropy loss for training a classifier in supervised learning [Frogner et al., 2015]. In that case, the Wasserstein loss is computed between distributions over the label space. The learning procedure proceeds by finding a minimizer of

$$\sum_{i=1}^{N} OT_\varepsilon(\delta_{\mathbf{y}_i} | f(\mathbf{x}_i)), \tag{1.3.19}$$

where $f(\mathbf{x}_i)$ is a vector of weights over each label and $\delta_{\mathbf{y}_i}$ is the distribution of the single label $\mathbf{y}_i$, which can also be represented as a one-hot vector. This approach requires to have a ground cost between each label pair in order to define the OT metric. Taking into account a ground cost between labels can help learn a better classifier, especially if the labels are noisy.

The above examples show that the OT loss can be used in several ways in order to train either unsupervised or supervised models. Similar ideas have for instance been used to perform dictionary learning or density fitting [Rolet et al., 2016, 2018, Genevay et al., 2018].

A relevant metric for histogram data

Histograms are often a useful representation of some data. Consider for instance a text document, taking its histogram consists in counting, for each word, how many times it appears in the text. Such a process amounts to ignore the order in which words appear in the text and rather consider a text as a weighted set of words (Fig. 1.2). Discarding any information relative to the ordering has two benefits. First it provides a lightweight representation of the data which can fit in low memory devices. Secondly, for many typical machine learning tasks, the word ordering may be unnecessary information which is better to discard in order to compare text documents or to learn from them more efficiently. Indeed, considering that the quantity of available training data is limited, learning from over-detailed data representations may cause overfitting. This kind of histogram representation of data is often referred to as *bag of something*, where *something* may be *words* for text documents, *visual words* such as SIFT features for an image, or simply *features* in a more general case. Many

other types of data can be seen as histograms, such as grayscale images where each bin is a pixel location and the weight on that bin is its grayscale value [Cuturi and Doucet, 2014], 3-dimensional shapes where a 3-dimensional discrete grid provides the bin locations and the weights are set to 0 or 1 depending on whether the bin is inside the shape or not [Bonneel et al., 2016], non-negative time-series [Cazelles et al., 2018] or spectrograms [Rolet et al., 2018].

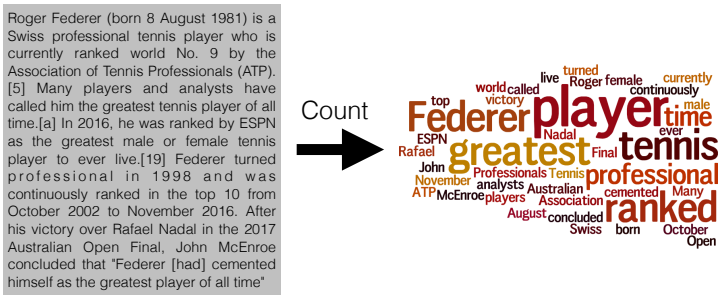

FIGURE 1.2: Taking the bag-of-words representation amounts to counting the occurrences of each word in the text, hence discarding the ordering of the words.

With histogram data at hand, one of the most basic concepts needed to build most machine learning algorithms (principal component analysis, linear regression, k-nearest neighbors etc.) is a metric between histograms, or at least some notion of divergence/discrepancy/metric or similarity. The most straightforward approach is to consider a histogram with d bins as a vector in $\mathbb{R}^d$ and then apply usual machine learning techniques within the standard Euclidean framework where the distance between two data points $\mathbf{h}_1$ and $\mathbf{h}_2$ is simply the Euclidean norm $d(\mathbf{h}_1, \mathbf{h}_2) = \|\mathbf{h}_2 - \mathbf{h}_1\|_2$. Other metrics can be considered such as the L^1 norm $\|\mathbf{h}_2 - \mathbf{h}_1\|_1$, the Hellinger distance $\|\sqrt{\mathbf{h}_2} - \sqrt{\mathbf{h}_1}\|_2$ or the Kullback-Leibler divergence $KL(\mathbf{h}_1||\mathbf{h}_2) = \sum_i h_{1i} \log(h_{1i}/h_{2i})$, the latter two being well-defined since histograms are non-negative. All the mentioned metrics are elementwise metrics, i.e. they are separable bin-wise. Hence when the support of each histogram, i.e. the bin with positive weights/count, are distinct, these metrics become less reliable as they are not sensitive to some translations of these supports as shown in Fig. 1.3.

On the contrary, the optimal transport metric $OT(\mathbf{h}_1, \mathbf{h}_2)$ computes the minimal cost to transport the weights of $\mathbf{h}_1$ to the weights of $\mathbf{h}_2$ w.r.t. a ground cost between the bins. Hence optimal transport is able to better evaluate the distance between (normalized) histograms, even when their respective supports are distinct. An illustration of this fact is shown in Fig. 1.3, where OT is computed between a reference spectrogram and a translated version of this spectrogram. When a relevant ground cost is available, i.e. a metric between the *features* of a bag-of-features, then the OT metric is powerful metric between histograms. This has been shown in text classification where a simple k-nearest neighbors algorithm using the optimal transport metric between bag-of-words, and where the ground cost is the Euclidean distance between the word2vec representations of the words [Mikolov et al., 2013], shows state-of-the-art performance [Kusner et al., 2015].

Domain adaptation

The previous section showed that the OT loss can be used as a powerful loss to train machine learning models. But optimal transport not only provides a powerful

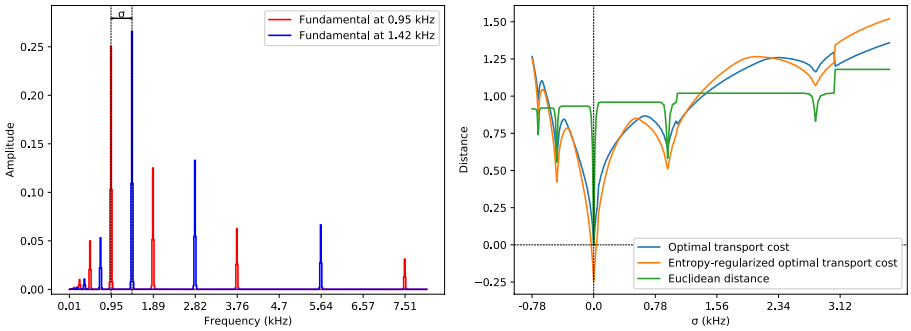

FIGURE 1.3: Figure taken from [Rolet et al., 2018]. Left: A reference normalized spectrogram and another normalized spectrogram (blue) obtained by translation of the support, followed by normalization. Right: Value of the OT cost, entropy-regularized OT cost and Euclidean distance between the reference spectrogram and its translated version for several frequency shift σ.

way to compare histograms or probability distributions, it also computes the optimal way to move one distribution to another through the concept of *optimal maps* or *optimal couplings*. Optimal maps have numerous applications, such as Bayesian inference [Moselhy and Marzouk, 2012, Marzouk et al., 2016], data assimilation [Reich, 2011], image registration [Haker et al., 2004], color transfer [Pitié et al., 2007] or shape matching [Su et al., 2015]. One interesting use of optimal maps in machine learning is *domain adaptation* (DA). In a typical domain adaptation setting, one has a source labeled dataset $\mathcal{S} = \{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_N, \mathbf{y}_N)\}$ and a target unlabeled dataset $\mathcal{T} = \{\mathbf{z}_1, \cdots, \mathbf{z}_N\}$. Both data sets features usually represent the same kind of data, such as text, sound or images, but marginal distributions of the features $\mathbf{x}$ and $\mathbf{z}$ may be very different due to different method for gathering each data set. For instance, a source data set may be composed of images objects in natural scenes while the target data set may be made of images of objects on a white background. In that case, a classifier learned on the source data set may perform poorly on the target set.

The idea of the *optimal transport domain adaptation* (OTDA) framework introduced by [Courty et al., 2014] is to use the optimal map obtained by solving the optimal transport problem in order to move the samples (features) of the source data set near the features of the target data set. Learning a classifier on the mapped source samples is likely to perform better, as illustrated by Fig. 1.4. This approach has been successful for providing state-of-the-art domain adaptation performance.

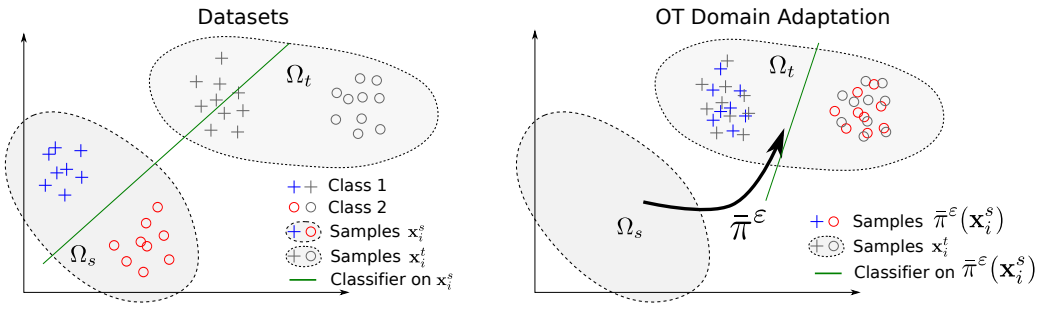

FIGURE 1.4: Illustration of the *optimal transport domain adaptation* method to learn a classifier on a target data set from a labeled source data set. The mapping function $\bar{\pi}^\varepsilon$ is computed as the barycentric projection of a regularized optimal plan π^ε.

1.4 Contributions

Along this first chapter, several concepts were introduced such as *transport maps*, *couplings*, *optimal transport*. These concepts find applications in numerous machine learning tasks as suggested by the recent literature [Marzouk et al., 2016, Genevay et al., 2016, Kolouri et al., 2017, Peyré et al., 2017, Arjovsky et al., 2017, Courty et al., 2017a, 2018]. So far, there are still many open challenges to use such concepts in a machine learning framework. One is computational: as we have seen, the computation of (optimal) transport maps, optimal transport couplings or the optimal transport objective for continuous measures or measures supported on a large support still lacks of tractable and efficient algorithms. Since it is important in machine learning to be able to scale to large data sets, algorithms for computing large-scale (optimal) transport are much needed. A second challenge is more theoretical and relates to how one can leverage the well-known mathematical theory of the Wasserstein space of probability measures [Villani, 2003, Ambrosio et al., 2006, Villani, 2008, Santambrogio, 2015] to build useful learning algorithms which can shed lights on some data set of histograms or probability measures. In the present thesis, which gathers the work done by my co-authors and I over the last four years, some of these challenges are addressed:

- In Chapter 2, we propose to obtain a *cartogram* using a transport map computed with a fast flow-based approach. Cartograms are visually appealing geographic map projections where each region is distorted in order to make its area proportional to some given number such as its population or GNP per capita. The computation of a transport map is a natural approach to obtain a *contiguous* cartogram since a transport map between the initial 2-dimensional density of the considered quantity and its average density provides a mapping which fulfills the definition of a cartogram. The proposed algorithm is faster than previous state-of-the-art methods by one order of magnitude, allowing to obtain cartograms in a matter of seconds on a regular laptop. We also show that cartograms provide a compelling visualization of geographic data beyond simple visual entertainment.

- In Chapter 3 we investigate the computation of regularized optimal transport in the large-scale or continuous setting. The convex regularization of the primal problem, with either an entropic or squared L^2 norm regularization term is equivalent to the relaxation of the dual problem, which we can solve with stochastic gradient methods in the same fashion as [Genevay et al., 2016]. We

show that using a full dual formulation approach rather than a semi-dual approach proposed by Genevay et al. [2016] in the discrete case provides a more simple algorithm which scales better with the size of the input measures. Whenever one or both input measures are continuous, we propose to parameterize the dual variables as deep neural networks, which provides a much more tractable algorithm than the kernel-based dual algorithm also proposed by Genevay et al. [2016]. The main novelty presented in this work lies in the stochastic computation of the barycentric projection of a regularized optimal plan, enabling to learn an approximation of an optimal map. This is to my knowledge the first algorithm able to provide optimal map approximations for probability measures which are continuous or supported on a large number of locations. Moreover, the parameterization of the barycentric projection as a deep neural network provides a map which generalizes to samples outside the support of the source measure, i.e. its domain of definition is not restricted to the support of the source measure. The power of the proposed algorithms is shown by performing two tasks which could not be done with previously proposed computational methods: firt we use the *optimal transport domain adaptation* (OTDA) method proposed by [Courty et al., 2014] on large-scale data sets, and second we propose a novel way to obtain a generative model by considering the learned approximate optimal map between a Gaussian measure and a target set as the generator of a generative model. On the theoretical side, we are able to prove some consistency theorems regarding the convergence of entropy-regularized optimal plans, and their barycentric projection, to the non-regularized optimal plan and Monge map respectively between the underlying probability measures, when the support of the empirical measures grows and the regularization amplitude decreases to 0. Although these consistency theorems have no useful practical results, they provide a theoretical justification to our proposed method: in the limit, the barycentric projection of a regularized plan indeed converges to an optimal map. Further investigations about convergence rates and probabilistic bounds of regularized optimal plans, as well as generalization to the squared L^2 regularization would be of great interest, but the current literature about this complex topic remains relatively scarce.

- In Chapter 4, we investigate how to use the geometry of the one-dimensional 2-Wasserstein space, i.e. the space of probability measures supported on the real line and equipped with the 2-Wasserstein metric, in order to compute principal geodesics of a data set of histograms or probability densities. The 2-Wasserstein space has similar concepts to those of Riemannian geometry: tangent vectors, tangent spaces, exponential / log maps and geodesics, allowing to generalize standard Euclidean PCA. Contrary to the more general 2-Wasserstein space of probability measures supported on $\mathbb{R}^d$ with $d > 1$, the one-dimensional 2-Wasserstein space is a *flat* space: it is isometric to its tangent space at any continuous probability measure, therefore allowing to cast the geodesic PCA problem as an optimization problem whose objective is simple to compute. The main difficulty lies in the constraint of optimizing over the set of geodesics or geodesic surfaces. We show that this involves imposing a constraint on the divergence of the directional tangent vector of a given geodesic. We rely on a proximal-based forward-backward algorithm in order to solve this challenging optimization problem with guarantees of convergence. We show on some synthetic or real data sets that principal geodesics

are able to capture more meaningful variations of some data sets of probability measures than their Euclidean principal component counterparts. The applicability of this work is limited by the fact that the proposed algorithms are specific to probability measures supported on the real line. In many cases, especially in machine learning, probability measures are supported on possibly high-dimensional spaces. Hence, we investigate in the following chapter how to design an algorithm which has no restriction on the dimensionality of the support of the probability measures.

- In Chapter 5, we extend the Wasserstein geodesic PCA approach introduced in Chapter 4 to the case of discrete probability measures supported on an arbitrary Hilbert space. As in Chapter 4, we rely on Riemannian geometry concepts such as Fréchet mean, tangent vectors and geodesics. Contrary to the one-dimensional 2-Wasserstein space, the isometry property does not hold and Wasserstein distances appear explicitly in the objective function. Moreover, the constraint on the divergence of the tangent vectors used in Chapter 4 does not guarantee geodesicity, and we propose instead to use the barycentric projection of an optimal plan as a projection step on the set of optimal maps. Since this is not a projection in the L^2 sense, the proposed projected gradient descend is not guaranteed to converge, although convergence has been observed in all our experiments.

Chapter 2

A Fast Flow-based Algorithm for producing Density-equalizing Map Projections

2.1 Introduction

A simple way of displaying statistical data in a diagram is the "area principle" where each part of the diagram should have an area in proportion to the number it represents [de Veaux et al., 2016]. Bar charts and pie charts are two well-known members of this type of diagram. For example, if our data are the electors that voted for the US president in December 2016, we can categorize the electors by US state. Every bar in the top half of Fig. 2.1 corresponds to a state that sent at least one Republican elector to the Electoral College. In the bottom half, the bars show the states with Democratic electors. The colors chosen for the bars are the traditional red for Republicans and blue for Democrats. Because the bar chart satisfies the area principle, the election is won by the color that occupies more area, which is evidently red in this example.[1]

Although a bar chart is often a suitable visualization tool, it does not reveal the spatial pattern behind the data. The bar chart of Fig. 2.1 lacks the information where the states are located: neighboring bars do not necessarily correspond to states that are geographic neighbors. If we want to visualize how the states fit together in real space, we need a different approach. The common alternative is to show a map such as Fig. 2.2A, where we use an Albers equal-area conic projection for the contiguous United States to produce their familiar geographic outline. We add Alaska and Hawaii, suitably rescaled, below the map of the contiguous United States. Each state is filled with either red or blue depending on the party affiliation of the electors.[2]

The map in Fig. 2.2A accurately shows the relative area and position of each state. However, it does not obey the area principle. For example, Montana (abbreviated by MT in Fig. 2.2A) covers more than 2000 times the area of Washington DC, but both regions have the same number of electors. On aggregate, Republican electors won 74% of the US area in square kilometers, but had only 57% of the vote share in the Electoral College. So, Fig. 2.2A has the opposite problem of Fig. 2.1 where we satisfied the statistical area principle, but conveyed no information about the

[1] Because of peculiarities in the US electoral system, the Electoral College is not an exact representation of the proportion of votes cast by the US population at large. The Republican candidate Donald Trump was elected as US president despite losing the popular vote to the Democratic candidate Hillary Clinton. We show a cartogram of the popular vote distribution in Fig. 2.7 of the Appendix.

[2] The only exception is Maine which applies the "congressional district method": although the majority in Maine voted for the Democratic candidate Hillary Clinton, the Republican candidate Donald Trump still gained one electoral vote for winning the 2nd congressional district (abbreviated as ME2 in Fig. 2.2A.)

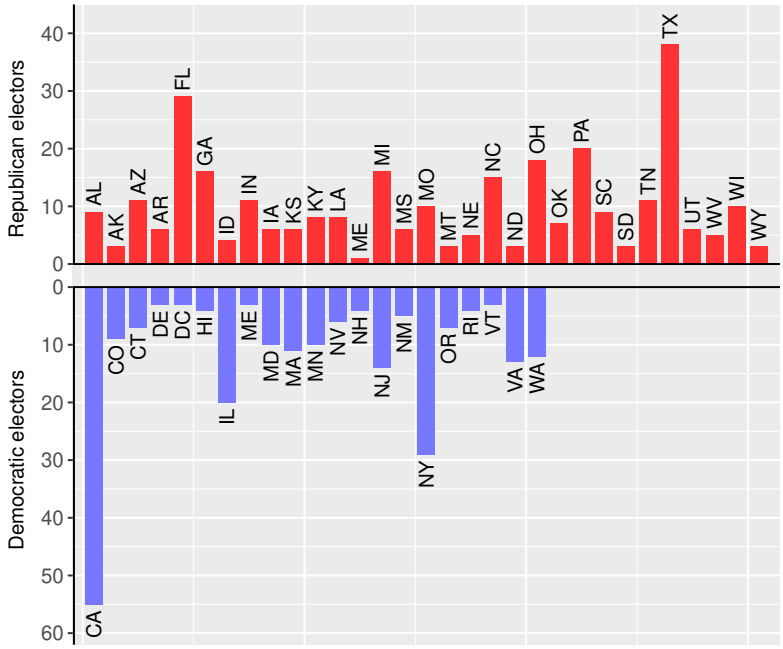

FIGURE 2.1: A bar chart of the Electoral College vote for the US president in 2016. This diagram satisfies the area principle: the area of each bar is proportional to the number of electors. However, from this bar chart it is not clear where states are located geographically.

states' locations. One might suspect that showing the locations and simultaneously satisfying the area principle are as impossible as squaring the circle. Fortunately, however, there is a visualization method, known as a *cartogram*, that can tackle this challenge [Tobler, 2004, Nusrat and Kobourov, 2016].

After a brief review and classification of cartograms, we introduce a technique that produces cartograms of a quality comparable to the most popular technique currently in use: the diffusion cartogram [Gastner and Newman, 2004]. The technique we propose solves a completely different set of equations so that the computation can finish within a fraction of the previously needed time. We benchmark our algorithm with data from the USA, India, China, and the London borough of Kensington and Chelsea to demonstrate that our method accurately satisfies the area principle and generates visually pleasing cartograms.

2.2 Classification of Cartogram Methods

In a cartogram, regions are deformed such that their areas are proportional to statistical data such as population, votes in an election, or gross domestic product. An example, showing the Electoral College on a cartogram, is the diagram in Fig. 2.2B which we adapted from Wikipedia [Ma, 2012]. Similar cartograms have been shown in the news media [Gamio, 2016, Parlapiano, 2016]. Here each elector is represented by a small square. The squares are then positioned with two objectives in mind. First, the shapes on the cartogram should resemble those on the map in Fig. 2.2A. Second, the set of neighboring states in Fig. 2.2A and Fig. 2.2B should be the same. Satisfying both objectives is not trivial. A careful comparison with Fig. 2.2A shows that, for example, Arizona (AZ) and Texas (TX) incorrectly appear as neighbors in

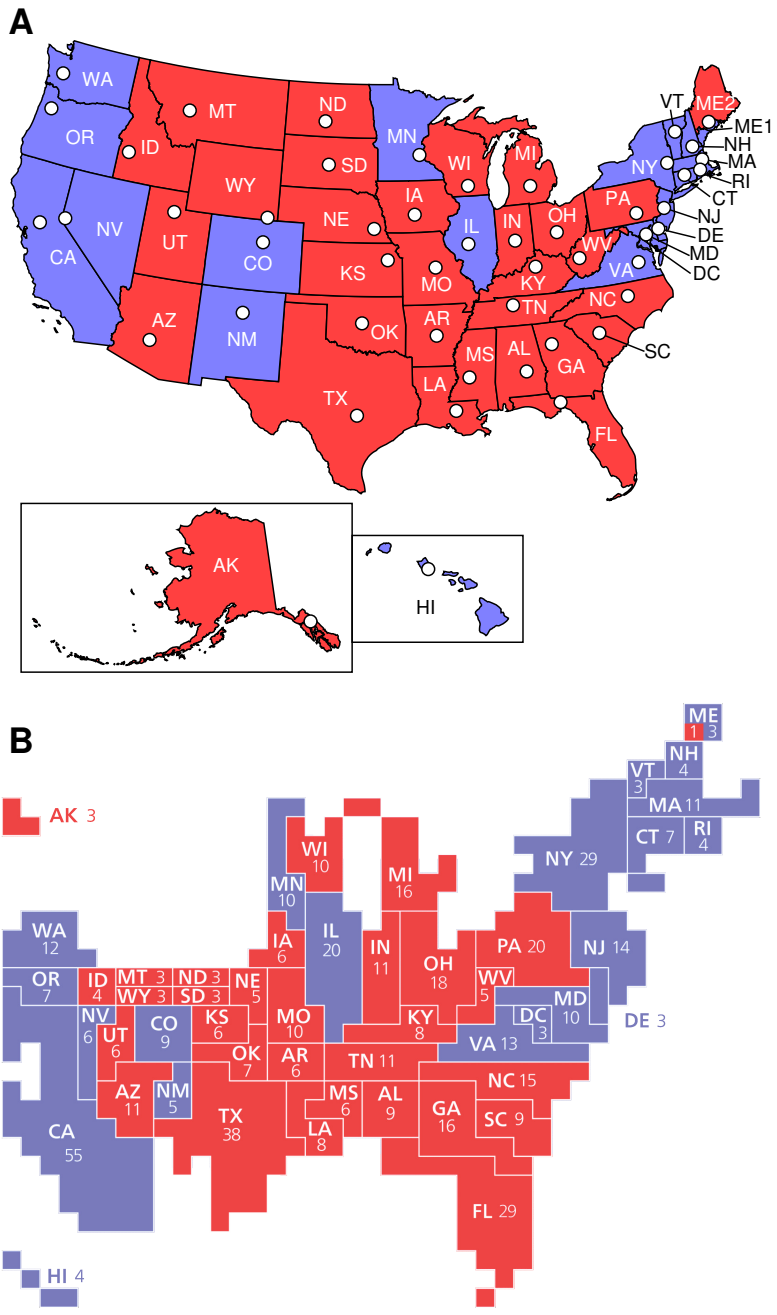

FIGURE 2.2: (A) A conventional map projection (here an Albers projection) clearly shows the location of each state, but violates the area principle: states that occupy a large area do not necessarily have a large number of electors. (B) A cartogram of the 2016 Electoral College [adapted from Wikipedia [Ma, 2012]] satisfies the area principle. Each elector is represented by a small square at the approximate location of the elector's home state. Cartograms such as these are popular in the media, but are not map projections in a strict sense: there is no continuous mathematical function that transforms coordinates of longitude and latitude to coordinates on the cartogram. For example, in (B) it is not possible to identify the location of the state capitals (indicated by white circles in panel A).

Fig. 2.2B. On the other hand, the geographic neighbors Colorado (CO) and Nebraska (NE) have been separated in Fig. 2.2B to make space for other states in the vicinity.

For certain applications, it is perfectly acceptable that neighboring states are split apart. So long as the areas of the states are proportional to the number of electors, such representations are called noncontiguous cartograms [Olson, 1976]. Dorling's circular cartograms are good examples of noncontiguous cartograms that, while not strictly maintaining the topology, indicate where the represented regions are located [Dorling, 1996]. Contiguous cartograms, by contrast, not only rescale the regions, but also keep the topology intact (i.e., neighbors on the map are neighbors on the cartogram and vice versa).

The methods that have been proposed for generating contiguous cartograms fall into two distinct categories. The first group consists of algorithms that operate only on the boundaries of regions [Dougenik et al., 1985, Merrill et al., 1992, House and Kocmoud, 1998, Keim et al., 2004, 2005, Inoue and Shimizu, 2006, Kämper et al., 2013, Cano et al., 2015, Daya Sagar, 2014]. Each region is represented by one or multiple polygons. The input to these algorithms are a finite number of polygon corners $(x_1, y_1), \ldots, (x_n, y_n)$. Here (x_i, y_i) is a projection of the longitude and latitude, usually obtained from a conventional projection (e.g., plate carrée or an equal-area projection). The algorithm generates transformed polygon coordinates $f(x_1, y_1), \ldots, f(x_n, y_n)$. For the first group of algorithms, these n points are in fact the only output and, hence, we refer to them as "boundaries-only" algorithms. In other words, boundaries-only methods do not transform points that are in the interior of a polygon. For example, on a US state cartogram (such as Fig. 2.2B) we would not be able to uniquely locate a state capital such as Austin, TX, because it is far from any state border. One might symbolically place all capitals at the centroid of the corresponding polygon, but some centroids might be outside the polygon if it is concave or contains holes (e.g., lakes or enclaves). The situation is even more complicated if we want to represent multiple distinct points or lines (e.g., rivers or roads) inside a state as distinct objects on a boundaries-only cartogram.

The second group of contiguous cartogram algorithms approaches the problem from a different point of view by producing a continuous transformation f for the entire continuous set of longitudes and latitudes on the input map, including coordinates that are not on a boundary [Tobler, 1973, Cauvin and Schneider, 1989, Gusein-Zade and Tikunov, 1993, Edelsbrunner and Waupotitsch, 1997, Gastner and Newman, 2004, Sun, 2013]. We refer to this group as "all-coordinates" algorithms. Generating the map projection f for all longitudes and latitudes can be computationally more demanding than only shifting the boundary coordinates. In fact, for applications where only the boundaries are of interest – as is the case for the US election map – the boundaries-only algorithms can give adequate results. However, the run time of these discrete algorithms typically increases steeply with the number of corners. As a result, they often rely on coarse-grained input to gain speed, for example by removing Michigan's Upper Peninsula from the US map [Keim et al., 2004, 2005, Cano et al., 2015]. If we wish to show data that are resolved at a scale much finer than the polygons to be displayed [e.g., graticules for a fine, spatially regular grid [Hennig, 2013] or individual addresses], the all-coordinates algorithms usually outpace their boundaries-only counterparts.

In the present work we describe an all-coordinates algorithm that only needs a few seconds to produce the complete projection f for realistic input. Knowing f will allow us to show the positions of all US state capitals with respect to the states' boundaries (Fig. 2.3B) and the coordinates of individual death cases in London (Fig. 2.4B and C).

2.3 Previous All-Coordinates Methods to Produce a Cartogram Projection

For the sake of concreteness, let us assume that we want to make a cartogram whose areas are proportional to the population. We define the population density as the function $\rho(x, y)$ such that a small rectangular area element with the corners $(x \pm dx/2, y \pm dy/2)$ contains the population $\rho(x, y)\, dx\, dy$. Some data allow us to model $\rho(x, y)$ with variations on fine spatial scales. (Our application below to the mortality statistics of Kensington and Chelsea belongs to this category.) In other cases, it is more natural to model $\rho(x, y)$ as a piecewise constant function. For example, California's 55 electors can be represented by a constant density in this state equal to the number of electors divided by the state's geographic area.

An accurate cartogram must project the rectangle $(x \pm dx/2, y \pm dy/2)$ onto a quadrilateral $f(x \pm dx/2, y \pm dy/2)$ in such a way that the area of the quadrilateral is proportional to $\rho\, dx\, dy$. In other words, we are looking for a two-dimensional function $f = (f_x, f_y)$ such that $\rho(x, y)\, dx\, dy = \bar{\rho}\, df_x\, df_y$ where $\bar{\rho}$ depends neither on x nor y. Such a transformation f is called a density-equalizing projection. Taking the limits $dx \to 0$ and $dy \to 0$ and assuming that f is differentiable, we obtain the condition [Tobler, 1973, Gastner and Newman, 2004],

$$\frac{\partial f_x}{\partial x}\frac{\partial f_y}{\partial y} - \frac{\partial f_x}{\partial y}\frac{\partial f_y}{\partial x} = \frac{\rho(x, y)}{\bar{\rho}}\,, \tag{2.3.1}$$

which is called a prescribed Jacobian equation [Dacorogna and Moser, 1990, Avinyó et al., 2003]. For convenience, we choose the constant $\bar{\rho}$ to be the spatially averaged density so that the total mapped area is preserved.

Equation 2.3.1 alone does not uniquely specify f because it is only one single equation for the two unknowns f_x and f_y [Gusein-Zade and Tikunov, 1993]. As a consequence, there are infinitely many different strategies to obtain a density-equalizing projection f. In practice, however, only a few methods are computationally efficient, produce attractive graphics and are independent of the choice of coordinate axes. Most of the methods that have been proposed in the literature are based on physical analogies. A common metaphor is to view the undistorted input map as a rubber sheet. Forces or stresses act on the rubber sheet such that the points move toward equilibrium positions that satisfy Eq. 2.3.1 [Cauvin and Schneider, 1989, Sun, 2013]. Although such mechanical metaphors make intuitive sense, there is no direct physical connection between force and area. Therefore, it is not immediately obvious how the forces should be chosen as functions of $\rho(x, y)$ to ensure that Eq. 2.3.1 is valid. Some methods treat the term "force" in a less literal sense so that the area constraints are more explicitly part of the equations [Dougenik et al., 1985, House and Kocmoud, 1998]. However, these algorithms must take special care to avoid topological errors (e.g., regions that are flipped or boundaries that intersect themselves) during the relaxation of the forces. Another method, based on neural networks, starts by placing sample points on a regular grid [Henriques et al., 2009]. During the training of the network, the samples are attracted toward regions of high density to mimic the population distribution. Their final positions define a mapping which can produce a cartogram by considering its inverse. However, a large number of sample points is necessary to produce smooth boundaries.

An alternative physical metaphor is to view the process that generates the cartogram as the flow of a fluid. In this analogy, we think of the map as a Petri dish

covered with a thin layer of water. In an experiment, we would model the population density $\rho(x, y)$ by injecting small particles with spatially varying concentrations into the water layer. The particles then diffuse across the entire Petri dish. In the long run, the probability density function of finding a particle becomes a constant everywhere inside the dish. We can make a cartogram by translating this simple physical model of density equalization into a geographic map projection.

The most familiar process that equilibrates the density is Brownian motion. On a macroscopic scale, the Fokker-Planck equation that describes Brownian motion is Fick's second law $\partial\rho_t/\partial t = D\nabla^2\rho_t$. Here t stands for time, D is the diffusivity and ∇^2 is the Laplace differential operator. This equation, known as the diffusion or heat equation, is at the heart of the "diffusion cartogram" method [Gastner and Newman, 2004, Avinyó et al., 2003]. An example of a diffusion cartogram is Fig. 2.3A where we show the US Electoral College results. The diffusion algorithm guarantees that, unlike in Fig. 2.2B, each state keeps its neighbors while still reaching the target areas to any desired level of accuracy. The diffusion cartogram distorts the shapes of the states, which is inevitable for any contiguous cartogram method. The shapes are, however, still recognizable; this is one of the reasons why diffusion cartograms have become popular in the past decade [Nusrat and Kobourov, 2016]. Another reason is that, despite the apparent complexity of the equations, they can be computed relatively efficiently.

However, Fickian diffusion is only one of many types of fluid dynamic rules that make particle densities equal everywhere. As we argue now, there is an alternative that is computationally more efficient while producing cartograms of comparable quality.

2.4 Flow-based Cartogram with Linear Equalization

Following the flow-based approach described in Sec. 1.2.1, we can build a flow-based cartogram by seeking a family of population densities (ρ_t) and a family of velocity fields $(\mathbf{v}_t)$ such that density approaches its mean in the long run $\lim_{t\to\infty}\rho(x, y, t) = \bar{\rho}$ for all x and y, and such that the continuity equation is verified.

$$\frac{\partial\rho_t}{\partial t} + \nabla\cdot(\rho\mathbf{v}_t) = 0. \tag{2.4.1}$$

That is, the particles must flow in such a way that all initial differences in their density are completely leveled out over time. If we know $\mathbf{v}(x, y, t)$ for all x, y, and t, we can compute the position $\mathbf{r}(t)$ for a point that is initially at $\mathbf{r}(0)$,

$$\mathbf{r}(t) = \mathbf{r}(0) + \int_0^t \mathbf{v}_t(\mathbf{r}(t'), t')dt'. \tag{2.4.2}$$

The projection f is the function that shifts $\mathbf{r}(0)$ to $\lim_{t\to\infty}\mathbf{r}(t)$. The justification that f defines a density-equalizing map follows from [Ambrosio et al., 2006, Proposition 8.1.8]. See also the Appendix (section 2) where we provide an informal justification of why f is density-equalizing.

We can satisfy the continuity equation while simultaneously demanding Fick's law $\mathbf{v} = -D(\nabla\rho_t)/\rho_t$. Substituting Fick's law into Eq. 2.4.1 shows that the evolution of ρ is then governed by the heat equation $\partial\rho_t/\partial t = D\nabla^2\rho_t$. This is the key motivation behind the diffusion cartogram method [Gastner and Newman, 2004], but Fickian diffusion is only one special case among a large class of processes in which ρ relaxes to its mean density while satisfying the continuity equation for some velocity

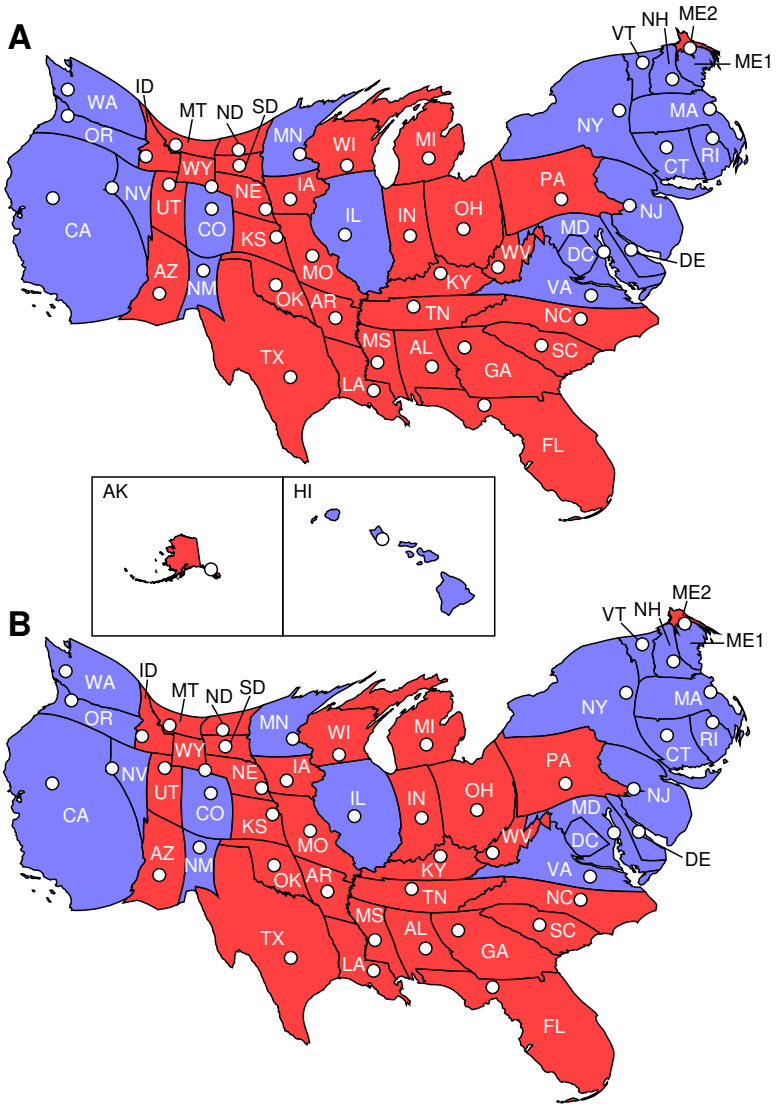

FIGURE 2.3: The 2016 US Electoral College vote represented on cartograms generated with (A) the diffusion algorithm of [Gastner and Newman, 2004] and (B) the alternative flow-based algorithm based on Eq. 2.4.3–2.4.6. The insets for Hawaii and Alaska apply to both (A) and (B) as these regions' areas match both cartograms. All areas differ by <1% from their target values (i.e., the proportion of votes in the Electoral College). Cartograms (A) and (B) differ in detail, but appear remarkably similar considering that generating (B) needs only 2.5% of the time required by the diffusion algorithm. The white circles indicate the positions of the state capitals.

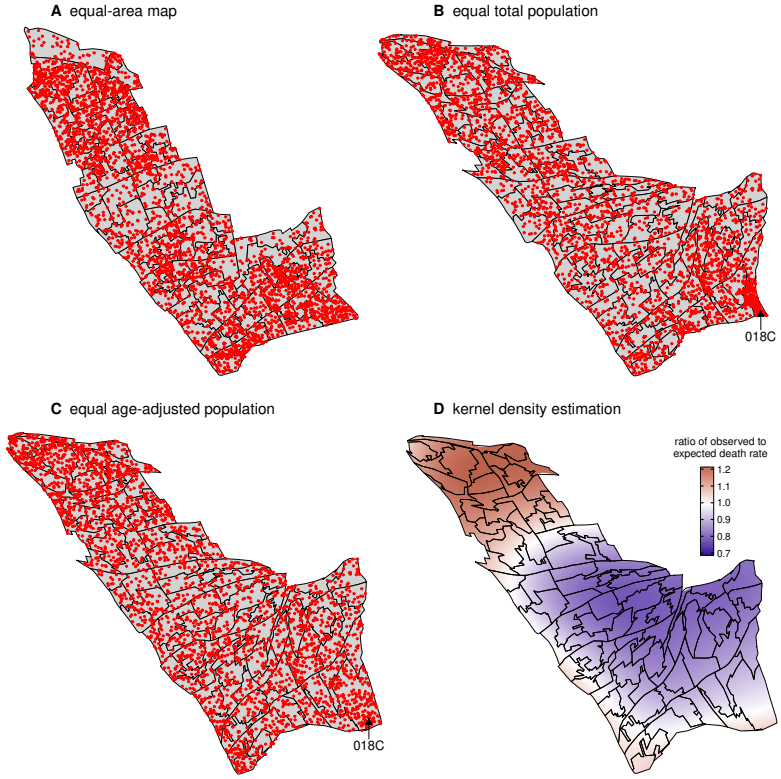

FIGURE 2.4: Maps with scatter plots of death cases in Kensington and Chelsea between 2011 and 2014 on (A) an equal-area map and on cartograms equalizing (B) the total population in each Lower Layer Super Output Area (LSOA) and (C) age-adjusted population (i.e., the expected number of deaths given the age and gender composition of the LSOA). Cartogram (B) reveals a high per-capita mortality in LSOA 018C in the southeast of the borough caused by a nursing home located inside this polygon. When accounting for the heterogeneous age distribution across the borough in (C), LSOA 018C has approximately the expected number of death cases. In other LSOAs, however, the expected and observed numbers differ. A kernel density estimate in panel (D) indicates an increasing trend in the age-adjusted death rate from the southeast to the northwest.

field $\mathbf{v}$. One advantage of Fickian diffusion is that the corresponding flow is guaranteed to be free of vortices that could cause severe local distortions in f. However, Fickian diffusion is not unique in this respect (see section 2.9.2 of the Appendix) so that one is left wondering whether other vortex-free, mass-conserving processes might also be suitable for generating cartograms. As we now argue, if we replace the heat equation by a linear equalization of the density toward the mean,

$$\rho(x,y,t) = \begin{cases} (1-t)\,\rho_0(x,y) + t\bar{\rho} & \text{if } t \le 1, \\ \bar{\rho} & \text{if } t > 1, \end{cases} \tag{2.4.3}$$

we can indeed compute f significantly faster. It has been shown that there exists a velocity field $\mathbf{v}$ for Eq. 2.4.3 so that the resulting transformation f satisfies Eq. 2.3.1 [Dacorogna and Moser, 1990]. We derive the concrete formulas for $\mathbf{v}$ in the Appendix and only give a brief summary here.

After an affine transformation of all coordinates, we place the mapped area inside a rectangular box with bounding coordinates $x_{\min} = 0$, $x_{\max} = L_x$, $y_{\min} = 0$, $y_{\max} = L_y$. (For later convenience, we choose L_x and L_y to be integers.) If we demand that there is no flow through the edges of the box, the velocity for $t \le 1$ can be expressed in terms of sine and cosine Fourier transforms,

$$v_x(x,y,t) = -\frac{L_y}{\pi\rho(x,y,t)} \sum_{m=1}^{\infty}\sum_{n=0}^{\infty} \left[\frac{m}{m^2L_y^2 + n^2L_x^2}\tilde{\rho}_{mn} \times \sin\left(\frac{m\pi x}{L_x}\right)\cos\left(\frac{n\pi y}{L_y}\right)\right], \tag{2.4.4}$$

$$v_y(x,y,t) = -\frac{L_x}{\pi\rho(x,y,t)} \sum_{m=0}^{\infty}\sum_{n=1}^{\infty} \left[\frac{n}{m^2L_y^2 + n^2L_x^2}\tilde{\rho}_{mn} \times \cos\left(\frac{m\pi x}{L_x}\right)\sin\left(\frac{n\pi y}{L_y}\right)\right] \tag{2.4.5}$$

with

$$\tilde{\rho}_{mn} = \frac{4}{(\delta_{m0}+1)(\delta_{n0}+1)} \times \int_0^{L_x}\int_0^{L_y} \rho(x',y',0)\cos\left(\frac{m\pi x'}{L_x}\right)\cos\left(\frac{n\pi y'}{L_y}\right)dx'dy'. \tag{2.4.6}$$

Here $\delta_{00} = 1$ and $\delta_{m0} = 0$ if $m \neq 0$. For $t > 1$, we simply obtain $v_x(x,y,t) = v_y(x,y,t) = 0$.

Equations 2.4.4–2.4.6 look superficially similar to the corresponding equations in the diffusion-based cartogram [Gastner and Newman, 2004], but there are two important differences. First, neither the sums in Eq. 2.4.4, 2.4.5 nor the integral in Eq. 2.4.6 depend on t so that the Fourier transforms need to be computed only once at the beginning of the calculation. Second, after we have computed the Fourier transforms, here we only require quick arithmetic operations: addition, subtraction, multiplication, and division. For a diffusion cartogram, by contrast, we must repeatedly calculate time-dependent Fourier transforms and evaluate the exponential function during the integration of Eq. 2.4.2 (see section 2.9.2 of the Appendix). The speed of computing the exponential function depends on details of the implementation and hardware, but is in general much slower than addition, subtraction, multiplication, or division [Brent, 1975].

These mathematical differences alone already cut the time needed per integration step by more than half. Another simplification compared with the diffusion cartogram is that we need to integrate Eq. 2.4.2 only until $t = 1$ instead of $t = \infty$. The benefit is that we no longer need to check whether the improper integral over the velocity has sufficiently converged. Most importantly, however, the integrals from different starting points $\mathbf{r}(0)$ can be performed in parallel as we now explain.

We overlay the map with an $L_x \times L_y$ square grid. For these $L_x L_y$ coordinates, we compute the sums and integrals in Eq. 2.4.4–2.4.6 at the start of the calculation with the fast Fourier transform algorithm [Frigo and Johnson, 2005]. We have found that the time needed for this one-time procedure is a negligible fraction of the total run time. After storing the $L_x L_y$ Fourier transforms in memory, we obtain $\mathbf{v}(\mathbf{r}, t)$ at each grid point $\mathbf{r}$ with basic arithmetic. Subsequently, we find the integrand in Eq. 2.4.2 for non-grid positions $\mathbf{r}$ by interpolating between the grid points. We numerically approximate the integral using a predictor-corrector method that automatically adapts the size of the next time step. During each step, we distribute the integration of the $L_x L_y$ distinct integrands to different processing units. In practice, given the wide availability of multi-core processors nowadays, this parallelization enormously boosts the speed of the calculation.

2.5 Benchmarking the Algorithm with Data for the USA, India, and China

We have implemented the algorithm based on Eq. 2.4.3–2.4.6 as a C program. In this section, we illustrate its performance with three case studies: the 2016 vote in the US Electoral College (Fig. 2.3B), the distribution of India's gross domestic product (GDP) by state (Fig. 2.5), and mainland China's and Taiwan's GDP by province (Fig. 2.6).

In each case, we first project the longitudes and latitudes of the territorial borders with an Albers equal-area conic projection onto a flat two-dimensional space. As described above, we embed the resulting map (Fig. 2.2A, 2.5A and 2.6A, respectively) inside an $L_x \times L_y$ rectangle whose edges act as reflecting boundaries for the flow. The rectangular box should, on one hand, be chosen large enough so that the cartogram is independent of the boundary conditions. On the other hand, it should not be so large that we spend the bulk of the run time on computing the projection f far from the region of interest. As a compromise, we have chosen the side length equal to 1.5 times the maximum of the countries' north-south and east-west extent. (These rectangular boxes are larger than the frames shown in Fig. 2.5 and Fig. 2.6 whose purpose is purely to visually separate the different panels in the figure.) The space between the country and the edges of the box is filled with the mean density $\bar{\rho}$. Other choices are conceivable and may improve shape preservation (e.g., by more faithfully retaining the outer boundaries of the map), but they would result in more complex computer code.

For the discrete Fourier transforms, we divide the large rectangular box into a grid of $L_x \times L_y$ smaller squares (in our examples $L_x = L_y = 512$, but the number can be adjusted if necessary) whose sizes are just fine enough to discern the smallest geographic regions on each map: Washington, DC in the USA; Daman and Diu in India (abbreviated by DD in Fig. 2.5); and Macao in China (MO in Fig. 2.6). Officially, these regions have neither the status of a state nor a province: Washington DC is a district, Daman and Diu a union territory, and Macao a Special Administrative

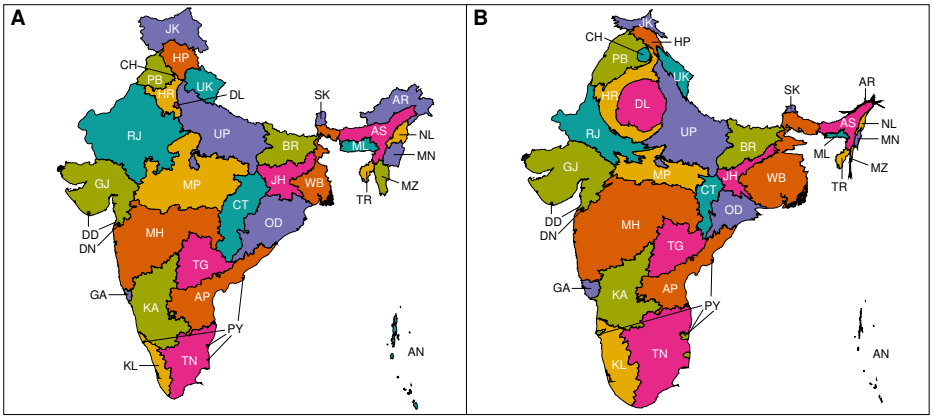

FIGURE 2.5: The states and union territories of India on (A) an equal-area map, (B) a cartogram where the area of each region is proportional to GDP (data from Statistics Times [Statistics Times, 2017]). The two largest states by area, Rajasthan (RJ) and Madhya Pradesh (MP), shrink on the cartogram because they only rank 7th and 10th in GDP, respectively. Maharashtra (MH), the state with the highest GDP, slightly grows on the cartogram. Even more striking is the increase of Delhi (DL): although small in area, the capital city has a higher GDP than many larger states. The opposite happens for Arunachal Pradesh (AR) and several other northeastern states because they rank low in GDP. Our algorithm only needs 2.6 seconds to construct the cartogram. AN, Andaman and Nicobar Islands; AP, Andhra Pradesh; AS, Assam; BR, Bihar; CH, Chandigarh; CT, Chhattisgarh; DN, Dadra and Nagar Haveli; DD, Daman and Diu; GA, Goa; GJ, Gujarat; HR, Haryana; HP, Himachal Pradesh; JK, Jammu and Kashmir; JH, Jharkhand; KA, Karnataka; KL, Kerala; MN, Manipur; ML, Meghalaya; MZ, Mizoram; NL, Nagaland; OD, Odisha; PY, Puducherry; PB, Punjab; RJ, Rajasthan; SK, Sikkim; TN, Tamil Nadu; TG, Telangana; TR, Tripura; UP, Uttar Pradesh; UK, Uttarakhand; WB, West Bengal.

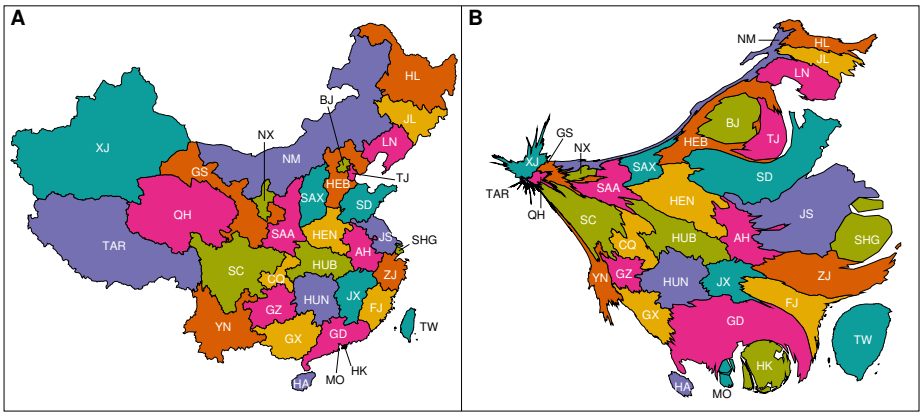

FIGURE 2.6: Provincial-level administrative divisions of mainland China and Taiwan on (A) an equal-area map, (B) a cartogram where areas are proportional to GDP (data from Wikipedia [Wikipedia, 2017]). Some coastal cities such as Shanghai (SHG) and Hong Kong (HK) increase remarkably on the cartogram. By contrast, western states such as Xinjiang (XJ) and the Tibet Autonomous Region (TAR) shrink dramatically. Despite the substantial deformations, our algorithm only needs 2.7 seconds to construct the cartogram. AH, Anhui; BJ, Beijing; CQ, Chongqing; FJ, Fujian; GS, Gansu; GD, Guangdong; GX, Guangxi; GZ, Guizhou; HA, Hainan; HEB, Hebei; HL, Heilongjiang; HEN, Henan; HUB, Hubei; HUN, Hunan; NM, Inner Mongolia; JS, Jiangsu; JX, Jiangxi; JL, Jilin; LN, Liaoning; MO, Macao; NX, Ningxia; QH, Qinghai; SAA, Shaanxi; SD, Shandong; SAX Shanxi; SC, Sichuan; TW, Taiwan; TJ, Tianjin; YN, Yunnan; ZJ, Zhejiang.

Region. We still include these regions on the cartograms because they are typically included on maps showing the states and provinces of their respective countries.[3]

When numerically integrating Eq. 2.4.2, the choice of time steps determines how accurately we estimate $\mathbf{r}(1)$. One possible strategy for achieving a highly accurate cartogram is to take a large number of small steps. After some experiments, we have decided to use a different strategy that achieves quicker run times and ultimately also comes arbitrarily close to a perfectly density-equalizing map. We use only a moderate number of adaptive time steps (≈ 100 in a typical run; the exact number is determined at runtime) during the initial integration. We expedite the convergence by applying a Gaussian blur of moderate width to the initial density prior to starting the integration. After one round of integration, the areas do not yet perfectly match their targets. For example, Washington DC still needs to grow by a factor ≈ 50. The key feature is to use the output of the first integration as input to another round of integration, which then usually comes closer to the objective areas. By repeating the integration sufficiently often, we have in all test cases observed that we can reach the objective areas with arbitrary precision. For the contiguous 48 states of the USA, we perform five iterations. Afterward, even for the extreme case of Washington DC, the smallest region in land area, the cartogram area differs by only 0.31% from the objective area. For India we iterate the integration twelve times and for China six times. The maximum differences between target and objective area are then 0.72% for the Andaman and Nicobar Islands (AN in Fig. 2.5B) and 0.83% for Tibet (TAR in Fig 2.6B), respectively. These differences are certainly so small that they cannot be

[3] We exclude the island territory of Lakshadweep from the maps of India because it is so small that it is neither visible on an equal-area map nor on a GDP cartogram.

detected by eye. We generally set a maximum relative area error of $< 1\%$, defined as

$$\text{relative area error} = \frac{\text{target area} - \text{objective area}}{\text{objective area}},$$

as stopping criterion for the algorithm.

This level of accuracy is all the more remarkable when considering the speed of our implementation. On a Dell Precision® T7810 workstation with a 12-core Intel® Xeon® E5-2680V3 processor and an Ubuntu 16.04.2 operating system, we need 1.5 seconds for the US Electoral College cartogram (Fig. 2.3B), 2.6 seconds for the India GDP cartogram (Fig. 2.5B), and 2.7 seconds for the China GDP cartogram (Fig. 2.6B). Compared with the diffusion algorithm, which needs 59.5 seconds to generate the US cartogram (Fig. 2.3A) with equal accuracy, this is a speedup by roughly a factor 40. Among other all-coordinates cartogram algorithms, only the rubbersheet method Carto3F [Sun, 2013] can achieve comparable speed, but not for all types of input. For a cartogram of Chinese provinces, Carto3F needs 8 minutes of computer time. Our fast flow-based method achieves smaller area errors in a fraction of this time.

2.6 Benchmarking with Data for Mortality in Kensington and Chelsea (London) 2011–2014

As noted above, cartogram algorithms that generate the complete density-equalizing projection f are particularly advantageous when displaying demographic data that are individual points on a map. We now demonstrate how our algorithm can be applied to such input and how we can use it to compare different statistical models. The data also serve as another benchmark for the speed of our method. Our example involves the locations of all 3197 death cases in the London borough of Kensington and Chelsea between the years 2011 and 2014. The database from the UK's Office for National Statistics (ONS) [Office for National Statistics, 2016] lists the number of deaths in each of London's 4835 Lower Layer Super Output Areas (LSOAs). A total of 103 LSOAs are located in Kensington and Chelsea. We show the density of death cases in this borough on an equal-area map in Fig. 2.4A. Each death corresponds to one point on the map placed at a random position inside the LSOA where it occurred.

The point pattern on the equal-area map is spatially heterogeneous with two bands of high density, one in the south and another in the north, separated by a band of lower density in the middle. However, it remains unclear from the equal-area map whether the differences in the spatial distribution of death cases are caused by differences in per-capita mortality or by a heterogeneous population density. We can distinguish between these two effects by projecting the death cases to a cartogram where each LSOA area is proportional to the number of inhabitants (Fig. 2.4B).

The most striking feature on this cartogram is the high per-capita mortality in the southeast corner of the borough. The reason for the high number of death cases in the LSOA with the ONS code "Kensington and Chelsea 018C" is a large proportion of elderly, most likely because of the St. Wilfrid's nursing home located in this LSOA. Because mortality increases markedly as a person becomes elderly, total population is too crude a measure to predict death rates. We now show how to improve the prediction by using each LSOA's age-adjusted mortality as the basis of a cartogram instead of the simple per-capita mortality displayed in Fig. 2.4B.

Data from the ONS [Office for National Statistics, 2015, 2016] include population size and death cases in the following age groups for each LSOA: 0 years old, 1-4, 5-9, ..., 85-89, and $\geq$ 90 years old, with each age group divided into men and women. For each of these 40 demographic subgroups, we can compute its total mortality in western central London (i.e., Kensington and Chelsea as well as the adjacent boroughs Brent, Westminster, Wandsworth, Hammersmith and Fulham). We denote by p_j the size of the population that lives in this part of London and belongs, because of its gender and age, to the demographic group j. If there were d_j deaths in this subpopulation, its region-wide per-capita mortality is $m_j = d_j/p_j$. The expected number of deaths in the i-th LSOA is thus $e_i = \sum_j p_{ij} m_j$, where p_{ij} is the population that lives in LSOA i and belongs to the demographic group j. This approach is known in the public health literature as age-adjustment [Lilienfeld and Stolley, 1994]. Unlike the unadjusted population size $\sum_j p_{ij}$, the expected value e_i makes a fair comparison between, for example, an LSOA mostly inhabited by a younger population and an LSOA with a large proportion of elderly inhabitants such as 018C.

In Fig. 2.4C, we show a cartogram with LSOA areas proportional to e_i. On this cartogram, the density of points in 018C is near the average in the borough, visualizing that age is indeed an important predictor for local death rates. Across the borough, however, differences between death rates still remain despite age-adjustment. We can quantify the deviation from spatial homogeneity, for example, with the Hopkins statistic H [Hopkins and Skellam, 1954], which is a number between 0 and 1. If a point pattern is caused by a homogeneous Poisson process (i.e., deaths are independent and equally likely everywhere), then the expected value of H equals 0.5. The more clustered the points are, the larger H is. We find $H = 0.524$ (95% confidence interval $[0.518, 0.530]$) in Fig. 2.4C, indicating that the data are inconsistent with a homogeneous Poisson process.

We show a kernel density estimate of the underlying probability distribution in Fig. 2.4D. We use a bivariate normal kernel with a bandwidth chosen according to [Venables and Ripley, 2002]. The figure reveals a minimum in the age-adjusted death rate in the east of the borough and a maximum in the north. Previous studies have argued that indicators of health (e.g., life expectancy) in different parts of London are positively correlated to average household income [Dorling, 2013]. A choropleth map of deprivation in Kensington and Chelsea [The Economist, 2017] does indeed follow a strikingly similar regional pattern as the death rate in Fig. 2.4D.

The flow-based method of Eq. 2.4.3–2.4.6 calculates the cartograms in Fig. 2.4B and C in 1.6 and 1.9 seconds respectively. To avoid boundary effects in Fig. 2.4D, we also include data for Kensington and Chelsea's neighboring boroughs when computing the cartograms and the kernel density estimate. The equivalent calculations with the diffusion-based method take 69.9 and 99.5 seconds respectively.

2.7 Measures of Distortion

Our algorithm is not only accurate and fast, but also generates cartograms whose visual appearance is on par with previous methods. In Fig. 2.3 we directly compare the diffusion cartogram of the USA (panel A) with the faster method based on Eq. 2.4.3–2.4.6 (panel B). The border between Illinois (IL) and Indiana (IN) is straighter in Fig. 2.3A than in Fig. 2.3B and thus more similar to the input map (Fig. 2.2A). On the other hand, the border between New Mexico (NM) and Colorado (CO) is straighter and Oklahoma's (OK) panhandle less bent in Fig. 2.3B. Overall, however, the differences between both cartograms are only subtle.

Because visual appearance is not a fully satisfactory criterion, we now turn to quantitative measures of distortion. One way to compare the local distortion of different projections is by analyzing the Tissot indicatrix that is constructed as follows. Suppose we draw an infinitesimal circle at the coordinates (x, y) on the input map. Locally, the effect of the projection f is to deform the circle into an ellipse, called the Tissot indicatrix of f at (x, y). Figure 2.8 in the Appendix shows concrete examples of Tissot indicatrices for our benchmarking examples. We denote the semi-major and -minor axes of the Tissot indicatrix by $a(x, y)$ and $b(x, y)$ respectively. Two measures of the local distortion error are [Papadopoulos, 2017]

$$e(x, y) = \ln\left(\frac{a(x, y)}{b(x, y)}\right)$$

and [Snyder, 1987]

$$\tilde{e}(x, y) = 2 \arcsin\left(\frac{a(x, y) - b(x, y)}{a(x, y) + b(x, y)}\right) .$$

For a conformal (i.e., angle-preserving) map, we would have $a = b$ for all (x, y) and thus $e = \tilde{e} = 0$. This scenario would be ideal, but, as we review in the Appendix, except in a few special cases there cannot be a conformal density-equalizing projection [Gusein-Zade and Tikunov, 1993]. As a global measure for the deviation of a cartogram from conformality, we can use for example either the spatially averaged or the maximum local distortion error,

$$e_a = \frac{1}{|\Omega|} \int_\Omega e(x, y) dx\, dy, \qquad e_\infty = \sup_{(x,y)\in\Omega} e(x, y),$$

where Ω is the spatial domain of the input map. In our comparison of the diffusion and fast flow-based algorithm in Table 2.1, we choose Ω to be the rectangular $L_x \times L_y$ bounding box that contains the area to be mapped as described above. By replacing e with $\tilde{e}$, we obtain similar measures $\tilde{e}_a$ and $\tilde{e}_\infty$.

TABLE 2.1: Measures of distortion applied to the diffusion algorithm and the flow-based algorithm using Eq. 2.4.3–2.4.6. Smaller values are highlighted in bold.

Map	Algorithm	e_a	e_∞	$\tilde{e}_a$	$\tilde{e}_\infty$	α	δ	θ	run time (seconds)
USA	diffusion	**0.278**	**6.85**	**0.273**	**3.01**	**2.01**	**17.1**	**0.0388**	59.5
	fast flow-based	0.285	7.06	0.280	3.02	2.04	17.4	0.0435	**1.5**
India	diffusion	**0.190**	3.95	**0.185**	2.59	2.45	**39.0**	**0.0281**	113.0
	fast flow-based	0.191	**3.18**	0.187	**2.34**	**2.40**	39.7	0.0290	**2.6**
mainland China and Taiwan	diffusion	0.590	**5.07**	0.553	**2.83**	2.33	**18.6**	**0.0849**	178.5
	fast flow-based	**0.570**	8.16	**0.530**	3.07	**2.19**	20.6	0.103	**2.7**
Kensington & Chelsea (age adj.)	diffusion	**0.161**	**6.86**	**0.154**	**3.01**	**2.03**	**22.1**	**0.0589**	99.5
	fast flow-based	0.163	7.08	0.156	3.03	2.20	24.1	0.0615	**1.9**

When computing e and $\tilde{e}$, we need to know f at each coordinate (x, y) so that these measures can only be applied to all-coordinates cartograms. Measures that aim to quantify the distortions also for other types of cartograms must instead rely on the polygons defining each region. In Table 2.1, we include three such measures from [Alam et al., 2015]: the average aspect ratio α, the Hamming distance δ and the relative position error θ. We provide details of their definition in the Appendix.

Briefly, the aspect ratio of a region is the ratio of the larger to the smaller side length of the bounding rectangle with minimum area, minimized over all possible rotations with respect to the coordinate axes. The Hamming distance between two polygons is the area lying within exactly one of them [Skiena, 2008]. For the measurement in Table 2.1, we rescale each polygon on the input map and the corresponding polygon on the cartogram so that they have equal area. We then calculate the minimum Hamming distance between these two polygons by shifting one polygon with respect to the other. We define δ as the sum of the minimum Hamming distances, where the summation is over all corresponding pairs of polygons. For the relative position error, we compute the angle between the line connecting the centroids of two polygons on the input map and the line that connects the centroids of the corresponding two polygons on the cartogram. We obtain θ by averaging over all pairs of polygons [Heilmann et al., 2004].

Most measures listed in Table 2.1 exhibit only small relative differences in the range of a few percent between the diffusion and fast flow-based method. Diffusion performs a little better in the majority of examples and measures, but there are also cases where the fast flow-based method produces a smaller error. Considering the vastly different run times, the fast flow-based method is the better solution as a general-purpose algorithm for interactive applets.

2.8 Conclusion

The scientific value of cartograms can go far beyond providing mere entertainment, shock, or amusement. As "isodemographic maps" they have been used for mapping diseases and mortality for several decades [Selvin et al., 1984, Dorling, 2005, Wieland et al., 2007] in order to improve health services [Lovett et al., 2014]. Our analysis of data for mortality in Kensington and Chelsea from 2011 to 2014 shows how cartograms allow to quickly grasp regional disparities after dismissing one non-relevant factor, here the the expected number of deaths given the age and gender of each region. Arguably, the technical challenge of computing the map projection has so far prevented more widespread use. The flow-based algorithm presented in this chapter allows to compute cartograms in a matter of seconds, while the popular diffusion-based algorithm [Gastner and Newman, 2004] needs at least several minutes on a regular desktop computer. We accompany this study with C code available at `https://github.com/Flow-Based-Cartograms/go_cart` to alleviate some of the challenge. The code optionally produces the graticule of the inverse transformation so that features found on the cartogram can be identified in the original domain. We reconstruct the original positions by first approximating f as a piecewise linear function and then computing its inverse.

2.9 Appendix

2.9.1 Cartogram of the popular vote in the 2016 US presidential election

US presidential elections are indirect: voters do not directly elect the president, but instead choose electors for their state who represent a presidential candidate in the Electoral College. The candidate with most votes in the Electoral College becomes the next president. 48 out of 50 states and Washington DC apply a winner-takes-all rule: the presidential candidate with the largest number of votes cast by the population in the state wins all of the state's electoral votes. The only exceptions are Maine and Nebraska. These two states apply the congressional district method: besides two electors for the state's aggregate winner, each congressional district chooses one elector for the candidate with most votes in this district.

The composition of the Electoral College does not need to be an accurate representation of the nationwide popular vote. The predominant winner-takes-all rule gives an advantage to a candidate who wins many states with narrow margins even if the opponent may have won more votes in the population as a whole. Furthermore, the number of votes in the Electoral College is not strictly proportional to state populations. There is a small bias in favor of less populated states by guaranteeing every state a minimum of three electors.

Historically, the winner of the nationwide popular vote has usually also won the Electoral College. However, the 2016 election was one of the exceptions. Hillary Clinton gained 48.2% of the popular vote, Donald Trump only 46.1%. Nevertheless, Trump won the Electoral College by 304 to 227 votes.

We visualize the popular vote on a cartogram (Fig. 2.7) by making each state's area proportional to the number of combined votes cast for Trump or Clinton in this state. We indicate the result with a color between blue (100% for Clinton) and red (100% for Trump). The shade of purple indicates how votes were split in each state. Cartograms with the same color scheme have been shown for previous US presidential elections [Gastner et al., 2005].

2.9.2 Motivating the equations used by the algorithm

Flow-based density-equalizing projections

Suppose we are given a population density $\rho_0(\mathbf{r})$ for every point $\mathbf{r} = (x, y)$ in a rectangle defined by $0 \le x \le L_x$ and $0 \le y \le L_y$. Our objective is to map the rectangle onto itself with a density-equalizing projection f. That is, assuming f is differentiable, it must satisfy

$$\det(\nabla f(\mathbf{r})) = \frac{\rho_0(\mathbf{r})}{\bar{\rho}} \tag{2.9.1}$$

for every point $\mathbf{r}$ in the rectangle. The left-hand side is the Jacobian determinant

$$\det(\nabla f(\mathbf{r})) = \frac{\partial f_x}{\partial x}\frac{\partial f_y}{\partial y} - \frac{\partial f_x}{\partial y}\frac{\partial f_y}{\partial x}$$

and the denominator in Eq. 2.9.1 is the spatially averaged density

$$\bar{\rho} = \frac{1}{L_x L_y}\int_0^{L_x}\int_0^{L_y} \rho_0(x, y)\, dx\, dy.$$

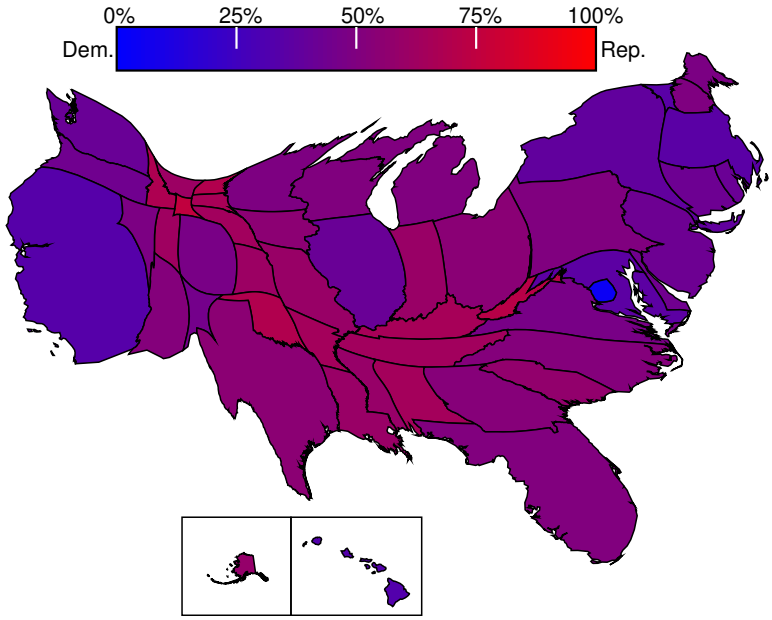

FIGURE 2.7: The popular vote in the 2016 US presidential election on a cartogram made with the fast flow-based algorithm described in the main text.

Loosely speaking, $\det(\nabla f(\mathbf{r}))$ is the factor by which a small area element near $\mathbf{r}$ is rescaled after applying the transformation f. The general form of Eq. 2.9.1 is called a "prescribed Jacobian equation".

The idea behind flow-based methods to find a solution f is to define a sequence of densities $\rho(x, y, t)$, where the nonnegative variable t represents time. We start from the given population density,

$$\rho(x, y, 0) = \rho_0(x, y), \tag{2.9.2}$$

and demand that ρ approaches in the long run the spatially averaged density,

$$\lim_{t\to\infty} \rho(x, y, t) = \bar{\rho}\,. \tag{2.9.3}$$

For constructing a flow-based cartogram, we also need a two-dimensional velocity field $\mathbf{v} = (v_x, v_y)$ for all x, y, and t. We define the map projection f_t of a point that is initially at $\mathbf{r} = (x, y)$ by

$$f_t(\mathbf{r}) = \mathbf{r} + \int_0^t \mathbf{v}(f_{t'}(\mathbf{r}))dt'\,. \tag{2.9.4}$$

We now argue that in the limit of infinite time, f_∞ is a density-equalizing projection (i.e., it satisfies Eq. 2.9.1) if the combination of ρ and $\mathbf{v}$ satisfies the continuity equation

$$\frac{\partial\rho}{\partial t} = -\nabla\cdot\mathbf{J}. \tag{2.9.5}$$

Here

$$\mathbf{J} = \rho\mathbf{v}$$

is the flux (i.e., the population that flows per unit time through a line of unit length perpendicular to $\mathbf{J}$) and $\nabla\cdot\mathbf{J} = \frac{\partial J_x}{\partial x} + \frac{\partial J_y}{\partial y}$ is the so-called divergence of $\mathbf{J}$.

An intuitive explanation why Eq. 2.9.2–2.9.5 imply Eq. 2.9.1 for f_∞ is as follows. Suppose a small, simply connected region $\mathcal{R}$ with area A contains the point $\mathbf{r}$. Because of the definition of the population density ρ_0, the initial population contained inside $\mathcal{R}$ is approximately equal to $A\rho_0(\mathbf{r})$. After the boundary has drifted with the flow and reached its final position, $\mathcal{R}$ has been mapped to a new region $\mathcal{S}$ with area $\approx A \cdot \det(\nabla \mathbf{T}_\infty(\mathbf{r}))$. As a consequence of Eq. 2.9.3, the population contained in $\mathcal{S}$ is approximately $\bar{\rho}A \cdot \det(\nabla \mathbf{T}_\infty(\mathbf{r}))$. The continuity equation 2.9.5 guarantees that the population inside any closed boundary is preserved while the boundary is drifting with the velocity field [Anderson, 1991]. Therefore, $A\rho_0(\mathbf{r}) = \bar{\rho}A \cdot \det(\nabla \mathbf{T}_\infty(\mathbf{r}))$. After canceling the common factor A on both sides and comparing with Eq. 2.9.1, we conclude that f_∞ is indeed a density-equalizing projection.

The general solution for vortex-free flow

Equations 2.9.2–2.9.5 are, under mild assumptions, sufficient to ensure that f_∞ is a density-equalizing projection. However, the equations have multiple solutions and many of them are in practice unsuitable for producing cartograms. In particular, solutions with vortices in the flux field $\mathbf{J}$ create severe local distortions in the vicinity of each vortex. We therefore add one more demand to Eq. 2.9.2–2.9.5,

$$\frac{\partial J_x}{\partial y} = \frac{\partial J_y}{\partial x}\,, \tag{2.9.6}$$

which guarantees that there are no vortices [Anderson, 1991].

Can we construct concrete pairs of a density $\rho(x,y,t)$ and a velocity $\mathbf{v}(x,y,t)$ that satisfy Eq. 2.9.2–2.9.6? Let us assume that $\rho(x,y,t)$ is a piecewise continuous function. At all points (x,y) where $\rho(x,y,t)$ is continuous, the cosine Fourier series of ρ converges pointwise to ρ. Thus, at these points we have

$$\rho(x,y,t) = \frac{1}{L_x L_y} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \tilde{\rho}_{mn} f_{mn}(t) \cos\left(\frac{m\pi x}{L_x}\right) \cos\left(\frac{n\pi y}{L_y}\right), \tag{2.9.7}$$

where

$$\tilde{\rho}_{mn} = \frac{4}{(\delta_{m0}+1)(\delta_{n0}+1)} \times \int_0^{L_x} \int_0^{L_y} \rho(x',y',0) \cos\left(\frac{m\pi x'}{L_x}\right) \cos\left(\frac{n\pi y'}{L_y}\right) dx'dy'$$

is the backward cosine Fourier transform of the initial density, δ_{m0} is the Kronecker symbol

$$\delta_{m0} = \begin{cases} 1 & \text{if } m = 0, \\ 0 & \text{otherwise,} \end{cases}$$

and $f_{mn}(t)$ is a function that must be consistent with the constraints expressed by Eq. 2.9.2–2.9.6.

The functions $\cos\left(\frac{m\pi x}{L_x}\right)$ with $m = 0, 1, \ldots$ are mutually orthogonal so that $f_{mn}(t)$ on the right-hand side of Eq. 2.9.7 is uniquely determined by $\rho(x,y,t)$ on the left-hand side. From this observation and Eq. 2.9.2, it follows that

$$f_{mn}(0) = 1 \quad \text{for all } m \text{ and } n. \tag{2.9.8}$$

Because of $\tilde{\rho}_{00} = \bar{\rho} L_x L_y$ and Eq. 2.9.3, we must have

$$\lim_{t\to\infty} f_{mn}(t) = \begin{cases} 1 & \text{if } m = n = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2.9.9}$$

To interpret the remaining constraints (i.e., Eq. 2.9.5 and Eq. 2.9.6) we must specify the boundary conditions of the flux $\mathbf{J}$. We assume that there is no flow through the edges of the rectangular box $[0, L_x] \times [0, L_y]$. Then it must be possible to express the x- and y-coordinates of the two-dimensional function $\mathbf{J}$ in terms of the following mixed sine and cosine Fourier transforms at all points (x, y) where $\mathbf{J}$ is continuous,

$$J_x(x,y,t) = \frac{1}{L_x L_y} \sum_{m=1}^{\infty} \sum_{n=0}^{\infty} \tilde{J}_{x,mn}(t) \sin\left(\frac{m\pi x}{L_x}\right) \cos\left(\frac{n\pi y}{L_y}\right), \tag{2.9.10}$$

$$J_y(x,y,t) = \frac{1}{L_x L_y} \sum_{m=0}^{\infty} \sum_{n=1}^{\infty} \tilde{J}_{y,mn}(t) \cos\left(\frac{m\pi x}{L_x}\right) \sin\left(\frac{n\pi y}{L_y}\right). \tag{2.9.11}$$

We insert Eq. 2.9.7, 2.9.10, and 2.9.11 into Eq. 2.9.5, interchange differentiation and summation, and finally compare each term in the series on the left- and right-hand side. The result is

$$\tilde{\rho}_{mn} f'_{mn}(t) = -\frac{m\pi}{L_x} \tilde{J}_{x,mn}(t) - \frac{n\pi}{L_y} \tilde{J}_{y,mn}(t). \tag{2.9.12}$$

For $m = n = 0$, the right-hand side is 0 so that $f'_{00}(t) = 0$. From this result and Eq. 2.9.8, we can deduce that

$$f_{00}(t) = 1 \quad \text{for all } t. \tag{2.9.13}$$

Similarly, we obtain, after inserting Eq. 2.9.10 and 2.9.11 into Eq. 2.9.6,

$$\frac{n}{L_y} \tilde{J}_{x,mn}(t) = \frac{m}{L_x} \tilde{J}_{y,mn}(t). \tag{2.9.14}$$

Combining Eq. 2.9.12 and Eq. 2.9.14, we can solve for the Fourier coefficients of the flux,

$$\tilde{J}_{x,mn}(t) = -\frac{m L_x L_y^2}{\pi(m^2 L_y^2 + n^2 L_x^2)} \tilde{\rho}_{mn} f'_{mn}(t), \tag{2.9.15}$$

$$\tilde{J}_{y,mn}(t) = -\frac{n L_x^2 L_y}{\pi(m^2 L_y^2 + n^2 L_x^2)} \tilde{\rho}_{mn} f'_{mn}(t). \tag{2.9.16}$$

In summary, a flow-based density-equalizing projection is vortex-free if and only if $f_{mn}(t)$ satisfies Eq. 2.9.8, 2.9.9, 2.9.13 and the Fourier coefficients of the flux obey Eq. 2.9.15 and 2.9.16.

Equations 4–7 in the main text as a special density-equalizing projection with vortex-free flow

There are many possible choices of f_{mn} consistent with Eq. 2.9.8, 2.9.9 and 2.9.13. The diffusion-based method of [Gastner and Newman, 2004] corresponds to the choice

$$f_{mn,\mathrm{diff}}(t) = \exp\left[-\left(\frac{m^2}{L_x^2}+\frac{n^2}{L_y^2}\right)t\right]. \tag{2.9.17}$$

According to Eq. 2.9.15 and 2.9.16, the Fourier coefficients of the flux are then given by

$$\tilde{J}_{x,mn,\mathrm{diff}}(t) = \frac{m}{\pi L_x}\tilde{\rho}_{mn}\exp\left[-\left(\frac{m^2}{L_x^2}+\frac{n^2}{L_y^2}\right)t\right], \tag{2.9.18}$$

$$\tilde{J}_{y,mn,\mathrm{diff}}(t) = \frac{n}{\pi L_y}\tilde{\rho}_{mn}\exp\left[-\left(\frac{m^2}{L_x^2}+\frac{n^2}{L_y^2}\right)t\right]. \tag{2.9.19}$$

It is computationally disadvantageous that t appears in the argument of the exponential function in Eq. 2.9.17–2.9.19. Whenever the numerical integration of Eq. 2.9.4 must advance the time t by a small increment, the Fourier coefficients, including the exponential function, must be computed again. Although modern computers can evaluate the exponential function relatively quickly, it is still slower than the four basic arithmetic operations (i.e., addition, subtraction, multiplication, and division). Even more time-consuming than the exponential function are the backward Fourier transforms to $\rho(x,y,t)$ and $\mathbf{J}(x,y,t)$, which we need in order to evaluate $\mathbf{v}$ appearing in Eq. 2.9.4.

The alternative approach that we explore is based on the choice

$$f_{mn}(t) = \begin{cases} 1 & \text{if } m=n=0, \\ 1-t & \text{if } (m,n)\neq(0,0) \text{ and } 0\le t\le 1, \\ 0 & \text{otherwise.}\end{cases} \tag{2.9.20}$$

instead of Eq. 2.9.17. Performing the backward transform in Eq. 2.9.7 shows that the density is

$$\rho(x,y,t) = \begin{cases} (1-t)\,\rho(x,y,0)+t\bar{\rho} & \text{if } 0\le t\le 1, \\ \bar{\rho} & \text{if } t>1.\end{cases} \tag{2.9.21}$$

Although the physical interpretation of the resulting flow is now less intuitive than for the diffusion-based method, the mathematical literature has explored solutions of the prescribed Jacobian equation 2.9.1 based on Eq. 2.9.21 ([Dacorogna and Moser, 1990, Moser, 1965]).

The Fourier coefficients of the flux follow from Eq. 2.9.15 and 2.9.16,

$$\tilde{J}_{x,mn}(t) = \begin{cases} \frac{mL_xL_y^2}{\pi(m^2L_y^2+n^2L_x^2)}\tilde{\rho}_{mn} & \text{if } 0\le t\le 1, \\ 0 & \text{otherwise.}\end{cases} \tag{2.9.22}$$

$$\tilde{J}_{y,mn}(t) = \begin{cases} \frac{nL_x^2L_y}{\pi(m^2L_y^2+n^2L_x^2)}\tilde{\rho}_{mn} & \text{if } 0\le t\le 1, \\ 0 & \text{otherwise.}\end{cases} \tag{2.9.23}$$

Upon inserting Eq. 2.9.22 and 2.9.23 into Eq. 2.9.10 and 2.9.11, we obtain the flux

$$J_x(x,y,t) = -\frac{L_y}{\pi}\sum_{m=1}^{\infty}\sum_{n=0}^{\infty}\left[\frac{m}{m^2L_y^2+n^2L_x^2}\tilde{\rho}_{mn} \times \sin\left(\frac{m\pi x}{L_x}\right)\cos\left(\frac{n\pi y}{L_y}\right)\right], \tag{2.9.24}$$

$$J_y(x,y,t) = -\frac{L_x}{\pi}\sum_{m=0}^{\infty}\sum_{n=1}^{\infty}\left[\frac{n}{m^2L_y^2+n^2L_x^2}\tilde{\rho}_{mn} \times \cos\left(\frac{m\pi x}{L_x}\right)\sin\left(\frac{n\pi y}{L_y}\right)\right] \tag{2.9.25}$$

for $0 \leq t \leq 1$. For $t > 1$, we simply get $J_x = J_y = 0$. When we divide J_x and J_y by the density ρ in Eq. 2.9.21, we obtain equations 5 and 6 in the main text.

There are multiple advantages when choosing Eq. 2.9.20 instead of Eq. 2.9.17.

- As we have just derived from Eq. 2.9.20, the flux is zero after $t = 1$. It follows that $f_1 = f_\infty$. Hence, there is no need to take the limit $t \to \infty$ when we perform the integral in Eq. 2.9.4. In practice, we no longer need to apply heuristics to test whether the integrand at time t is small enough to terminate the numerical integration. Instead, we integrate until the fixed upper integration limit $t = 1$, which is easier to implement.

- Unlike in the diffusion-based method, we can calculate the density $\rho(x,y,t)$ in Eq. 2.9.21 without Fourier transforms.

- In the diffusion-based method, the Fourier coefficients of the flux (Eq. 2.9.18 and 2.9.19) are time-dependent. Therefore, at every new time step during the numerical integration of Eq. 2.9.4, we must carry out a new backward Fourier transform. By contrast, the right-hand sides of Eq. 2.9.24 and 2.9.25 do not depend on t. It suffices to perform the summations once at the start of the algorithm, most efficiently with the fast Fourier transform technique [Frigo and Johnson, 2005]. If we store the result in memory, we do not need any more Fourier transforms at all during the integration.

- After computing the sums in Eq. 2.9.24 and 2.9.25 at the beginning of the code, we only need addition, subtraction, multiplication, and division. In particular, we never need to evaluate the exponential function that appears in Eq. 2.9.17 of the diffusion-based method.

The overall effect is remarkably fast computer code. For typical runs, we find that a serial implementation of the algorithm based on Eq. 2.9.20 only takes around 18% of the time needed for the diffusion-based method. By parallelizing the integrator, it is possible to speed up the code even further. With a 12-core processor, we were able to reduce the time needed by the new algorithm to only around 3% of the run-time for the diffusion-based code.

2.9.3 Tissot ellipses and angular-distortion metrics

In cartography, the Tissot indicatrix is a visual and numerical concept to analyze the distortions generated by a map projection. Introduced by Nicolas Auguste Tissot in the nineteenth century, the Tissot indicatrix has become an important tool, especially

when characterizing projections of the Earth's (nearly) spherical surface onto a two-dimensional plane. Our proposed framework is different: we are transforming a two-dimensional map (the cartogram input) to another two-dimensional map (the cartogram). Still, we can use Tissot indicatrices to measure the magnitude of the distortions produced by different cartogram algorithms.

Tissot ellipses

Consider an infinitesimally small circle centered at (x, y) on the input map. Locally, a smooth map projection f is approximately equal to the affine transformation

$$f(x + \delta_x, y + \delta_y) \approx f(x, y) + \nabla f(x, y) \begin{pmatrix} \delta_x \\ \delta_y \end{pmatrix} , \tag{2.9.26}$$

so long as δ_x and δ_y are sufficiently small. Here ∇f is the Jacobian matrix. It can be shown that any affine transformation applied to a circle results in an ellipse. The Tissot indicatrix of (x, y) under the projection f is the ellipse generated by the affine transformation on the right-hand side of Eq. 2.9.26 when applied to the infinitesimal circle at (x, y). In the left-hand column of Fig. 2.8, we show several circles placed at regularly spaced locations on the input maps of our benchmarking examples (USA by electors, India and China by GDP). We use a finite radius to make the circles visible. In the middle and right-hand columns, we show the corresponding Tissot indicatrices centered at locations $f(x, y)$ on diffusion and fast-flow based cartograms, respectively.

Angular-distortion metrics

It is desirable for a density-equalizing projection f to preserve shapes as much as possible so that each area is easily recognizable by the reader of the cartogram. One way to interpret shape preservation is to demand that angles remain locally unchanged by the transformation f. This property is referred to as *conformality*. Equivalently, a conformal transformation must satisfy both Cauchy-Riemann equations

$$\frac{\partial f_x}{\partial x} = \frac{\partial f_y}{\partial y}, \quad \frac{\partial f_x}{\partial y} = -\frac{\partial f_y}{\partial x} .$$

Together with the prescribed Jacobian equation (1) in the main text, a conformal density-equalizing projection would have to satisfy three equations. In general, two functions f_x and f_y cannot satisfy three independent constraints so that a perfectly conformal density-equalizing solution is infeasible [Gusein-Zade and Tikunov, 1993]. Yet, a visually pleasing cartogram should deviate from conformality as little as possible. Although this study focuses on building a fast cartogram algorithm rather than achieving small conformality error, we compute several conformality metrics to verify that our proposed algorithm produces cartograms that are in this respect as good as the state-of-the-art diffusion algorithm.

Angular distortion metrics can be derived from the properties of Tissot ellipses. Consider the Tissot ellipse that is the image of the unit-radius circle centered at (x, y) after applying the affine transformation of Eq. 2.9.26. We denote the length of the ellipse's semi-major and semi-minor axis by $a(x, y)$ and $b(x, y)$, respectively. In the case of a conformal projection f, we would have $a(x, y) = b(x, y)$. That is, the Tissot ellipse would be a circle whose radius can be smaller or bigger than 1. Hence, we

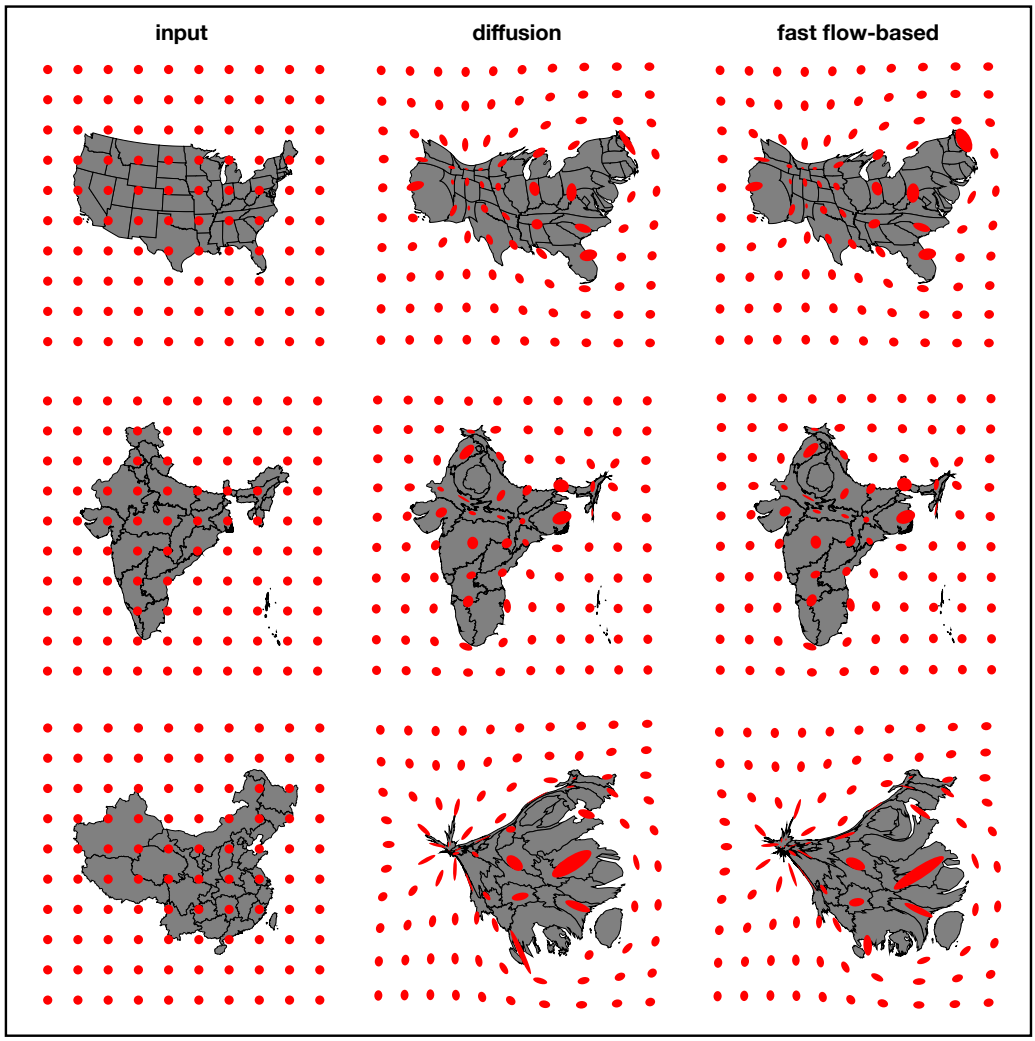

FIGURE 2.8: Tissot indicatrices obtained for the diffusion-based algorithm (middle column) and the fast flow-based algorithm proposed in the main text (right column). The unprojected circles are displayed in the left column. The cartograms for the USA (top row), India (middle row), mainland China and Taiwan (bottom row) are based on the same data as Fig. 2.3, 2.5 and 2.6.

can define a measure of the angle distortion at (x, y) by

$$e(x,y) = \ln\left(\frac{a(x,y)}{b(x,y)}\right), \tag{2.9.27}$$

as described for example in [Papadopoulos, 2017]. We choose the average

$$e_a = \frac{1}{|\Omega|}\int_\Omega \ln\left(\frac{a(x,y)}{b(x,y)}\right) dx\, dy,$$

and the largest value

$$e_\infty = \sup_{\mathbf{x}\in\Omega} \ln\left(\frac{a(x,y)}{b(x,y)}\right)$$

as two global measures for the distortion error, where Ω is the total area of the cartogram. Here we choose Ω as the $L_x \times L_y$ bounding rectangle described in the section "Benchmarking the algorithm with data for the USA, India, and China" in the main text. Another angular-distortion metric can be computed from the local maximum angular-value change (see derivation in [Snyder, 1987]),

$$\tilde{e}(x,y) = 2\arcsin\left(\frac{a(x,y) - b(x,y)}{a(x,y) + b(x,y)}\right), \tag{2.9.28}$$

which also provides two global angular-distortion metrics $\tilde{e}_a$ and $\tilde{e}_\infty$. We display these errors for both the diffusion-based and our new proposed algorithm in Table 1 of the main text.

2.9.4 Polygon-level distortions

The metrics e and $\tilde{e}$ defined in Eq. 2.9.27 and 2.9.28 are local: they can be computed only for *"all-coordinates"* cartograms for which we know the transformation f at every location (x, y). In order to obtain metrics that are well-defined for general cartograms, one has to measure distortions at the level of polygons instead of all coordinates. Such metrics have been introduced in several previous articles [Alam et al., 2015, Keim et al., 2003, Heilmann et al., 2004]. According to some metrics, the fast flow-based algorithm defined in the main text and several other contiguous methods, including the diffusion cartogram, are already optimal. For example, both the diffusion and fast flow-based algorithm succeed in rescaling the regions to their objective areas and preserve the adjacency between polygons. Three metrics that meaningfully compare the diffusion and fast-flow based algorithm are (1) the average aspect ratio α, (2) the total Hamming distance δ, and (3) the relative position error θ. We describe them below and display the results for each cartogram in Table 1 of the main text.

Average aspect ratio

Cartograms in which polygons become thin and elongated are difficult to read. It is also difficult to place labels inside such polygons. The aspect ratio of a polygon quantifies how stretched it appears. For the i-th polygon on the cartogram, we define $l_i(\phi)$ and $s_i(\phi)$ as the longer and shorter side length, respectively, of the bounding rectangle whose edges are at angles ϕ and $\phi + 90^\circ$ with respect to the x-axis.

We define $\phi_{\min,i}$ as the angle at which the bounding rectangle for the i-th polygon has the minimum area,

$$\phi_{\min,i} = \arg\min_{\phi\in[0^\circ,90^\circ]}[l_i(\phi)\cdot s_i(\phi)].$$

The aspect ratio of the i-th polygon is the ratio of the larger to the smaller side length of the bounding rectangle of minimum area for that polygon,

$$\alpha_i = \frac{l_i(\phi_{\min,i})}{s_i(\phi_{\min,i})}.$$

If there are p polygons on the cartogram, we define α as the mean aspect ratio,

$$\alpha = \frac{1}{p}\sum_{i=1}^{p}\alpha_i\ .$$

Total Hamming distance

The Hamming distance h measures the difference in the shapes of two polygons. It is computed by superimposing one polygon on top of another and measuring the fraction of area that lies in only one, but not both polygons,

$$h = \frac{\text{area in exactly one polygon}}{\text{sum of areas of individual polygons}}\ .$$

In our application, one of the polygons is from the input map, the other is the corresponding polygon from the cartogram. We rescale the cartogram polygon so that it has the same area as the polygon before the cartogram projection. Otherwise we would unfairly penalize cartograms that correctly changed the polygon areas to their objective values. To make the measure translation invariant, we define δ_i as the minimum Hamming distance of all possible translations of the rescaled i-th cartogram polygon with respect to the i-th unprojected polygon. The total Hamming distance δ is obtained by summing the Hamming distances of all polygons.

Relative position error

We can quantify changes in the relative position of two polygons i and j between input map and cartogram by measuring the angle ϕ_{ij} between the lines connecting the centroids before and after the projection. If $\mathbf{c}_i$, $\mathbf{c}_j$ are the centroids on the input map and $\mathbf{d}_i$, $\mathbf{d}_j$ on the cartogram, then

$$\phi_{ij} = \arccos\left(\frac{(\mathbf{c}_i-\mathbf{c}_j)\cdot(\mathbf{d}_i-\mathbf{d}_j)}{|\mathbf{c}_i-\mathbf{c}_j|\cdot|\mathbf{d}_i-\mathbf{d}_j|}\right)\ .$$

We define the relative position error θ as the average of ϕ_{ij} over all possible pairs of polygons. We also divide by π,

$$\theta = \frac{2}{p(p-1)\pi}\sum_{i=1}^{p-1}\sum_{j=i+1}^{p}\phi_{ij},$$

so that $\theta\in[0,1]$ [Heilmann et al., 2004].

Chapter 3

Large-Scale Optimal Transport and Mapping Estimations

3.1 Introduction

Mapping one distribution to another Given two random variables X and Y taking values in $\mathcal{X}$ and $\mathcal{Y}$ respectively, the problem of finding a transport map, i.e. a map f such that $f(X)$ and Y have the same distribution, denoted $f(X) \sim Y$ henceforth, finds applications in many areas, as outlined in the introductory chapter 1. For instance, in domain adaptation, given a source dataset and a target dataset with different distributions, the use of a mapping to align the source and target distributions is a natural formulation [Gopalan et al., 2011] since theory has shown that generalization depends on the similarity between the two distributions [Ben-David et al., 2010]. Current state-of-the-art methods for computing generative models such as generative adversarial networks [Goodfellow et al., 2014], generative moments matching networks [Li et al., 2015] or variational auto encoders [Kingma and Welling, 2013] also rely on finding f such that $f(X) \sim Y$. In this setting, one often chose variable X, referred to as the latent variable, as a continuous random variable from which it is easy to sample, and Y is a discrete distribution of real data, e.g. the ImageNet dataset. By learning a map f, sampling from the generative model boils down to drawing a sample from X and then applying f to that sample. The previous chapter also showed that such maps can produce visually appealing geographic map projections.

Mapping with optimality Among the potentially many maps f verifying $f(X) \sim Y$, the map which minimizes the total transportation cost of transporting the mass from X to Y w.r.t. some ground metric is referred to as a *Monge map*, or as an *optimal map*. Such *optimal maps* can be useful in numerous applications such as color transfer [Ferradans et al., 2014], shape matching [Su et al., 2015], image registration [Haker et al., 2004], data assimilation [Reich, 2011, 2013], or Bayesian inference [Moselhy and Marzouk, 2012]. In small dimension and for some specific costs, multi-scale approaches [Mérigot, 2011] or dynamic formulations [Evans and Gangbo, 1999, Benamou and Brenier, 2000, Papadakis et al., 2014, Solomon et al., 2014] can be used to compute optimal maps, but these approaches become intractable in higher dimension as they are based on space discretization. Furthermore, maps veryfiying $f(X) \sim Y$ might not exist, for instance when X is a constant but not Y. Still, one would like to find optimal maps between distributions at least approximately. The modern approach to OT relaxes the Monge problem by optimizing over plans, i.e. distributions over the product space $\mathcal{X} \times \mathcal{Y}$, rather than maps, casting the OT problem as a linear program which is always feasible and easier to solve. However, even with specialized algorithms such as the network simplex, solving that linear program takes $O(n^3 \log n)$ time, where n is the size of the discrete distribution (measure)

support.

Large-scale OT Recently, Cuturi [2013] showed that introducing entropic regularization into the OT problem turns its dual into an easier optimization problem which can be solved using the Sinkhorn algorithm. However, the Sinkhorn algorithm does not scale well to measures supported on a large number of samples, since each of its iterations has an $\mathcal{O}(n^2)$ complexity. In addition, the Sinkhorn algorithm cannot handle continuous probability measures. To address these issues, two recent works proposed to optimize variations of the dual OT problem through stochastic gradient methods. Genevay et al. [2016] proposed to optimize a "semi-dual" objective function. However, their approach still requires $\mathcal{O}(n)$ operations per iteration and hence only scales moderately w.r.t. the size of the input measures. Arjovsky et al. [2017] proposed a formulation that is specific to the so-called 1-Wasserstein distance (unregularized OT using the Euclidean distance as a cost function). This formulation has a simpler dual form with a single variable which can be parameterized as a neural network. This approach scales better to very large datasets and handles continuous measures, enabling the use of OT as a loss for learning a generative model. However, a drawback of that formulation is that the dual variable has to satisfy the non-trivial constraint of being a Lipschitz function. As a workaround, Arjovsky et al. [2017] proposed to use weight clipping between updates of the neural network parameters. However, this makes unclear whether the learned generative model is truly optimized in an OT sense. Besides these limitations, these works only focus on the computation of the OT objective and do not address the problem of finding an optimal map between two distributions.

Contributions In this chapter, We present a novel two-step approach for learning an *optimal map* f that satisfies $f(X) \sim Y$. First, we compute an optimal transport plan, which can be thought as a one-to-many map between the two distributions. To that end, we propose a new simple dual stochastic gradient algorithm for solving regularized OT which scales well with the size of the input measures. We provide numerical evidence that our approach converges faster than semi-dual approaches considered in [Genevay et al., 2016]. Second, we learn an *optimal map* (also referred to as a *Monge map*) as a neural network by approximating the barycentric projection of the OT plan obtained in the first step. Parameterization of this map with a neural network allows efficient learning and provides generalization outside the support of the input measure. Fig. 3.1 provides a 2D example showing the computed map between a Gaussian measure and a discrete measure and the resulting density estimation. On the theoretical side, we prove the convergence of regularized optimal plans (resp. barycentric projections of regularized optimal plans) to the optimal plan (resp. Monge map) between the underlying continuous measures from which data are sampled. We demonstrate our approach on domain adaptation and generative modeling.

Notations: We denote $\mathcal{X}$ and $\mathcal{Y}$ some complete metric spaces. In most applications, these are Euclidean spaces. We denote random variables such as X or Y as capital letters. We use $X \sim Y$ to say that X and Y have the same distribution, and also $X \sim \mu$ to say that X is distributed according to the probability measure μ. $\text{Supp}(\mu)$ refers to the support of μ, a subset of $\mathcal{X}$, which is also the set of values which $X \sim \mu$ can take. Given $X \sim \mu$ and a map f defined on $\text{Supp}(\mu)$, $f\#\mu$ is the probability distribution of $f(X)$. We say that a measure is continuous when it admits a density w.r.t. the Lebesgues measure. We denote id the identity map.

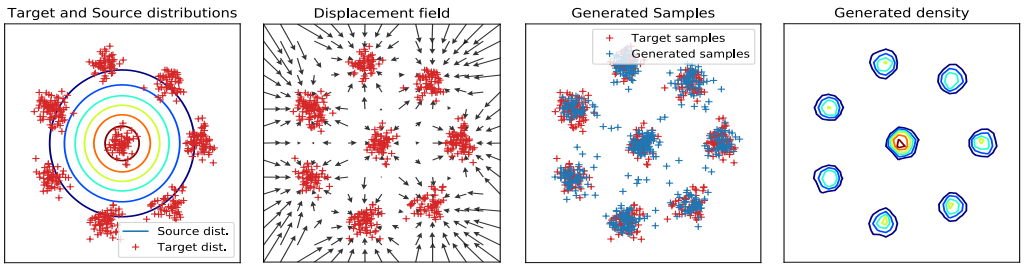

FIGURE 3.1: Example of estimated optimal map between a Gaussian distribution (colored level sets) and a multi-modal discrete measure (red +). (Left) Continuous source and discrete target distributions. (Center left) displacement field of the estimated optimal map: each arrow is proportional to $f(\mathbf{x}_i) - \mathbf{x}_i$ where $(\mathbf{x}_i)$ is a uniform discrete grid. (Center right) Generated samples obtained by sampling from the source distribution and applying our estimated optimal map f. (Right) Level sets of the resulting density (approximated as a 2D histogram over 10^6 samples).

3.2 Reminder on Optimal Transport

The Monge Problem Consider a cost function $c : (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y} \mapsto c(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^+$, and two random variables $X \sim \mu$ and $Y \sim \nu$ taking values in $\mathcal{X}$ and $\mathcal{Y}$ respectively. The Monge problem [Monge, 1781] consists in finding a map $f : \mathcal{X} \to \mathcal{Y}$ which transports the mass from μ to ν while minimizing the mass transportation cost,

$$\inf_f \mathbb{E}_{X\sim\mu}\left[c(X, f(X))\right] \quad \text{subject to } f(X) \sim Y. \tag{3.2.1}$$

Monge originally considered the cost $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$, but in the present chapter we refer to the Monge problem as Problem (3.2.1) for any cost c. When μ is a discrete measure, a map f satisfying the constraint may not exist: if μ is supported on a single point, no such map exists as soon as ν is not supported on a single point. In that case, the Monge problem is not feasible. However, when $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$, μ admits a density and c is the squared Euclidean distance, an important result by Brenier [1991] states that the Monge problem is feasible and that the infinum of Problem (3.2.1) is attained. The existence and uniqueness of *Monge maps*, also referred to as *optimal maps*, was later generalized to more general costs (e.g. strictly convex and super-linear) by several authors. With the notable exception of the Gaussian to Gaussian case which has a closed form affine solution, computation of *Monge maps* remains an open problem for measures supported on high-dimensional spaces.

Kantorovich Relaxation In order to make Problem (3.2.1) always feasible, Kantorovich [1942] relaxed the Monge problem by casting Problem (3.2.1) into a minimization over couplings $(X, Y) \sim \pi$ rather than the set of maps, where π should have marginals equal to μ and ν,

$$\inf_\pi \mathbb{E}_{(X,Y)\sim\pi}\left[c(X, Y)\right] \quad \text{subject to } X \sim \mu,\ Y \sim \nu. \tag{3.2.2}$$

Concretely, this relaxation allows mass at a given point $x \in \text{Supp}(\mu)$ to be transported to several locations $\mathbf{y} \in \text{Supp}(\nu)$, while the Monge problem would send the whole mass at $\mathbf{x}$ to a unique location $f(\mathbf{x})$. This relaxed formulation is a linear program, which can be solved by specialized algorithms such as the network simplex when considering discrete measures. However, current implementations of

this algorithm have a super-cubic complexity in the size of the support of μ and ν, preventing wider use of OT in large-scale settings.

Regularized OT OT regularization was introduced by Cuturi [2013] in order to speed up the computation of OT. Regularization is achieved by adding a negative-entropy penalty R (defined in Eq. (3.3.2)) to the primal variable π of Problem (3.2.2),

$$\inf_{\pi} \mathbb{E}_{(X,Y)\sim\pi}\left[c(X,Y)\right] + \varepsilon R(\pi) \text{ subject to } X \sim \mu,\ Y \sim \nu. \tag{3.2.3}$$

Besides efficient computation through the Sinkhorn algorithm, regularization also makes the OT distance differentiable everywhere w.r.t. the weights of the input measures [Blondel et al., 2018], whereas OT is differentiable only almost everywhere. We also consider the L^2 regularization introduced by Dessein et al. [2016], whose computation is found to be more stable since there is no exponential term causing overflow. As highlighted by Blondel et al. [2018], adding an entropy or squared L^2 norm regularization term to the primal problem (3.2.3) makes the dual problem an unconstrained maximization problem. We use this dual formulation in the next section to propose an efficient stochastic gradient algorithm.

3.3 Large-Scale Regularized Optimal Transport

By considering the dual of the regularized OT problem, we first show that stochastic gradient ascent can be used to maximize the resulting concave objective. A closed form for the primal solution π of Problem (3.2.3) can then be obtained by using first-order optimality conditions.

3.3.1 Dual stochastic approach

OT dual Let $X \sim \mu$ and $Y \sim \nu$. The Kantorovich duality provides the following dual of the OT problem (3.2.2),

$$\sup_{u\in L^1(d\mu,\mathcal{X}),\ v\in L^1(d\nu,\mathcal{Y})} \mathbb{E}_{(X,Y)\sim\mu\times\nu}\left[u(X)+v(Y)\right], \tag{3.3.1}$$

$$\text{subject to } u(\mathbf{x})+v(\mathbf{y}) \leqslant c(\mathbf{x},\mathbf{y}) \text{ for } (\mu\times\nu)-\text{almost all } (\mathbf{x},\mathbf{y}).$$

This dual formulation suggests that stochastic gradient methods can be used to maximize the objective of Problem (3.3.1) by sampling batches from the independant coupling $\mu\times\nu$. However there is no easy way to fulfill the constraint on u and v along gradient iterations. This motivates considering regularized optimal transport.

Regularized OT dual The hard constraint in Eq. (3.3.1) can be relaxed by regularizing the primal problem (3.2.2) with a strictly convex regularizer R as detailed in [Blondel et al., 2018]. In this chapter we consider both entropy regularization R_e used in [Cuturi, 2013, Genevay et al., 2016] and L^2 regularization R_{L^2},

$$R_e(\pi) \overset{\text{def.}}{=} \int_{\mathcal{X}\times\mathcal{Y}} \left(\ln\left(\frac{d\pi(\mathbf{x},\mathbf{y})}{d\mu(\mathbf{x})d\nu(\mathbf{y})}\right)-1\right) d\pi(\mathbf{x},\mathbf{y}),\quad R_{L^2}(\pi) \overset{\text{def.}}{=} \int_{\mathcal{X}\times\mathcal{Y}} \left(\frac{d\pi(\mathbf{x},\mathbf{y})}{d\mu(\mathbf{x})d\nu(\mathbf{y})}\right)^2 d\mu(\mathbf{x})d\nu(\mathbf{y}). \tag{3.3.2}$$

where $\frac{d\pi(\mathbf{x},\mathbf{y})}{d\mu(\mathbf{x})d\nu(\mathbf{y})}$ is the density, i.e. the Radon-Nikodym derivative, of π w.r.t. $\mu\times\nu$. This density is always well-defined since it is simple to prove by contraposition that for $\pi \in \Pi(\mu,\nu)$, $\mu\times\nu(B) = 0$ implies that $\pi(B) = 0$ for any measurable set B in $\mathcal{X}\times\mathcal{Y}$. When μ and ν are discrete, and so is π, the integrals are replaced by

sums. The dual of the regularized OT problems can be obtained through the Fenchel-Rockafellar's duality theorem,

$$\sup_{u,v} \mathbb{E}_{(X,Y)\sim\mu\times\nu}\left[u(X)+v(Y)+F_\varepsilon(u(X),v(Y))\right], \tag{3.3.3}$$

$$\text{where } F_\varepsilon(u(\mathbf{x}),v(\mathbf{y})) = \begin{cases} -\varepsilon e^{\frac{1}{\varepsilon}(u(\mathbf{x})+v(\mathbf{y})-c(\mathbf{x},\mathbf{y}))} & \text{(entropy reg.)} \\ -\frac{1}{4\varepsilon}(u(\mathbf{x})+v(\mathbf{y})-c(\mathbf{x},\mathbf{y}))_+^2 & (L^2 \text{ reg.}) \end{cases}. \tag{3.3.4}$$

Compared to Problem (3.3.1), the constraint $u(\mathbf{x})+v(\mathbf{y}) \leqslant c(\mathbf{x},\mathbf{y})$ has been relaxed and is now enforced smoothly through a penalty term $F_\varepsilon(u(\mathbf{x}),v(\mathbf{y}))$ which is concave w.r.t. (u,v). Although we derive formula and perform experiments w.r.t. entropy and L^2 regularizations, any strictly convex regularizer which is decomposable, i.e. which can be written as $R(\pi)=\sum_{ij} R_{ij}(\pi_{ij})$ (in the discrete case), gives rise to a dual problem of the form Eq. (3.3.3), and the proposed algorithms can be adapted.

Primal-Dual relationship In order to recover the solution π^ε of the regularized primal problem (3.2.3), we can use the first-order optimality conditions of the Fenchel-Rockafellar's duality theorem,

$$d\pi^\varepsilon(\mathbf{x},\mathbf{y}) = H_\varepsilon(\mathbf{x},\mathbf{y})d\mu(\mathbf{x})d\nu(\mathbf{y}) \text{ where } H_\varepsilon(\mathbf{x},\mathbf{y}) = \begin{cases} e^{\frac{u(\mathbf{x})}{\varepsilon}} e^{-\frac{c(\mathbf{x},\mathbf{y})}{\varepsilon}} e^{\frac{v(\mathbf{y})}{\varepsilon}} & \text{(entropy reg.)} \\ \frac{1}{2\varepsilon}\left(u(\mathbf{x})+v(\mathbf{y})-c(\mathbf{x},\mathbf{y})\right)_+ & (L^2 \text{ reg.}) \end{cases}. \tag{3.3.5}$$

Algorithm The relaxed dual (3.3.3) is an unconstrained concave problem which can be maximized through stochastic gradient methods by sampling batches from $\mu\times\nu$. When μ is discrete, i.e. $\mu=\sum_{i=1}^n a_i\delta_{\mathbf{x}_i}$, the dual variable u is a n-dimensional vector over which we carry the optimization, where $u(\mathbf{x}_i) \overset{\text{def.}}{=} u_i$. When μ has a density, u is a function on $\mathcal{X}$ which has to be parameterized in order to carry optimization. We thus consider deep neural networks for their ability to approximate general functions. Genevay et al. [2016] used the same stochastic dual maximization approach to compute the regularized OT objective in the continuous-continuous setting. The difference lies in their pamaterization of the dual variables as kernel expansions, while we decide to use deep neural networks. Using a neural network for parameterizing a continuous dual variable was done also by Arjovsky et al. [2017]. The same discussion also stands for the second dual variable v. Our stochastic gradient algorithm is detailed in Alg. 1.

Convergence rates and computational cost comparison. We first discuss convergence rates in the discrete-discrete setting (i.e. both measures are discrete), where the problem is convex, while parameterization of dual variables as neural networks in the semi-discrete or continuous-continuous settings make the problem non-convex. Because the dual (3.3.3) is not strongly convex, full-gradient descent converges at a rate of $\mathcal{O}(1/k)$, where k is the iteration number. SGD with a decreasing step size converges at the inferior rate of $\mathcal{O}(1/\sqrt{k})$ [Nemirovski et al., 2009], but with a $\mathcal{O}(1)$ cost per iteration. The two rates can be interpolated when using mini-batches, at the cost of $\mathcal{O}(p^2)$ per iteration, where p is the mini-batch size. In contrast, Genevay et al. [2016] considered a semi-dual objective of the form $\mathbb{E}_{X\sim\mu}[u(X)+G_\varepsilon(u(X))]$, with a cost per iteration which is now $\mathcal{O}(n)$ due to the computation of the gradient of G_ε. Because that objective is not strongly convex either, SGD converges at the same $O(1/\sqrt{k})$ rate, up to problem-specific constants. As noted by Genevay et al. [2016], this rate can be improved to $\mathcal{O}(1/k)$ while maintaining the same iteration cost, by using stochastic average gradient (SAG) method [Schmidt et al., 2017]. However,

Algorithm 1 Stochastic OT computation

Inputs: input measures μ, ν; cost function c; batch size p; regularization ε; learning rate γ.
Discrete case: $\mu = \sum_i a_i \delta_{\mathbf{x}_i}$ and u is a finite vector: $u(\mathbf{x}_i) \overset{\text{def.}}{=} u_i$ (similarly for ν and v)
Continuous case: μ is a continuous measure and u is a neural network (similarly for ν and v)
∇ indicates the gradient w.r.t. the parameters
while not converged **do**
  sample a batch $(\mathbf{x}_1, \cdots, \mathbf{x}_p)$ from μ
  sample a batch $(\mathbf{y}_1, \cdots, \mathbf{y}_p)$ from ν
  update $(u, v) \leftarrow (u + \gamma \sum_{ij} \nabla u(\mathbf{x}_i) + \partial_u F_\varepsilon(u(\mathbf{x}_i), v(\mathbf{y}_j)) \nabla u(\mathbf{x}_i), v \leftarrow v + \gamma \sum_{ij} \nabla v(\mathbf{y}_j) + \partial_v F_\varepsilon(u(\mathbf{x}_i), v(\mathbf{y}_j)) \nabla v(\mathbf{y}_j)$
end while

SAG requires to store past stochastic gradients, which can be problematic in a large-scale setting.

In the semi-discrete setting (i.e. one measure is discrete and the other is continuous), SGD on the semi-dual objective proposed by Genevay et al. [2016] also converges at a rate of $\mathcal{O}(1/\sqrt{k})$, whereas we only know that Alg. 1 converges to a stationary point in this non-convex case.

In the continuous-continuous setting (i.e. both measures are continuous), Genevay et al. [2016] proposed to represent the dual variables as kernel expansions. A disadvantage of their approach, however, is the $\mathcal{O}(k^2)$ cost per iteration. In contrast, our approach represents dual variables as neural networks. While non-convex, our approach preserves a $\mathcal{O}(p^2)$ cost per iteration. This parameterization with neural networks was also used by Arjovsky et al. [2017] who maximized the 1-Wasserstein dual-objective function $\mathbb{E}_{(X,Y)\sim\mu\times\nu}[u(X) - u(Y)]$. Their algorithm is hence very similar to ours, with the same complexity $\mathcal{O}(p^2)$ per iteration. The main difference is that they had to constrain u to be a Lipschitz function and hence relied of weight clipping in-between gradient updates. The proposed algorithm is capable of computing the regularized OT objective and optimal plans between empirical measures supported on arbitrary large numbers of samples. In statistical machine learning, one aims at estimating the underlying continuous distribution from which empirical observations have been sampled. In the context of optimal transport, one would like to approximate the true (non-regularized) optimal plan between the underlying measures. The next section states theoretical guarantees regarding this problem.

3.3.2 Convergence of regularized OT plans

Consider discrete probability measures $\mu_n = \sum_{i=1}^n a_i \delta_{\mathbf{x}_i} \in P(\mathcal{X})$ and $\nu_n = \sum_{j=1}^n b_j \delta_{\mathbf{y}_j} \in P(\mathcal{Y})$. Analysis of entropy-regularized linear programs [Cominetti and San Martín, 1994] shows that the solution π_n^ε of the entropy-regularized problem (3.2.3) converges exponentially fast to a solution π_n of the non-regularized OT problem (3.2.2). Also, a result about stability of optimal transport [Villani, 2008][Theorem 5.20] states that, if $\mu_n \to \mu$ and $\nu_n \to \nu$ weakly, then a sequence (π_n) of optimal transport plans between μ_n and ν_n converges weakly to a solution π of the OT problem between μ and ν. We can thus write,

$$\lim_{n\to\infty} \lim_{\varepsilon\to 0} \pi_n^\varepsilon = \pi. \tag{3.3.6}$$

A more refined result consists in establishing the weak convergence of π_n^ε to π when (n, ε) jointly converge to $(\infty, 0)$, at least for ε decreasing fast enough. This is the result of the following theorem which states a stability property of entropy-regularized plans (proof in the Appendix).

Theorem 2. *Let $\mu \in P(\mathcal{X})$ and $\nu \in P(\mathcal{Y})$ where $\mathcal{X}$ and $\mathcal{Y}$ are complete metric spaces. Let $\mu_n = \sum_{i=1}^n a_i \delta_{\mathbf{x}_i}$ and $\nu_n = \sum_{j=1}^n b_j \delta_{\mathbf{y}_j}$ be discrete probability measures which converge weakly to μ and ν respectively, and let (ε_n) a sequence of non-negative real numbers converging to 0 sufficiently fast. Assume the cost c is continuous on $\mathcal{X} \times \mathcal{Y}$ and finite. Let $\pi_n^{\varepsilon_n}$ the solution of the entropy-regularized OT problem (3.2.3) between μ_n and ν_n. Then, up to extraction of a subsequence, $(\pi_n^{\varepsilon_n})$ converges weakly to the solution π of the OT problem (3.2.2) between μ and ν,*

$$\pi_n^{\varepsilon_n} \to \pi \quad \textit{weakly}. \tag{3.3.7}$$

Keeping the analogy with statistical machine learning, this result is an analog to the universal consistency property of a learning method. In most applications, we consider empirical measures and n is fixed, so that regularization, besides enabling dual stochastic approach, may also help learn the optimal plan between the underlying continuous measures.

So far, we have derived an algorithm for computing the regularized OT objective and regularized optimal plans regardless of μ and ν being discrete or continuous. The OT objective has been used successfully as a loss in machine learning [Montavon et al., 2016, Frogner et al., 2015, Rolet et al., 2016, 2018, Arjovsky et al., 2017, Courty et al., 2017a], whereas the use of optimal plans has straightforward applications in logistics, as well as economy [Kantorovich, 1942, Carlier, 2012] or computer graphics [Bonneel et al., 2011]. In numerous applications however, we often need mappings rather than joint distributions. This is all the more motivated since Brenier [1991] proved that when the source measure is continuous, the optimal transport plan is actually induced by a map. Assuming that available data samples are sampled from some underlying continuous distributions, finding the Monge map between these continuous measures rather than a discrete optimal plan between discrete measures is essential in machine learning applications. Hence in the next section, we investigate how to recover an optimal map, i.e. find an approximate solution to the Monge problem (3.2.1), from regularized optimal plans.

3.4 Optimal Mapping Estimations

A map can be obtained from a solution to the OT problem (3.2.2) or regularized OT problem (3.2.3) through the computation of its barycentric projection. Indeed, a solution π of Problem (3.2.2) or (3.2.3) between a source measure μ and a target measure ν is, identifying the plan π with its density w.r.t. a reference measure, a function $\pi : (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}^+$ which can be seen as a weighted one-to-many map, i.e. π sends $\mathbf{x}$ to each location $\mathbf{y} \in \text{Supp}(\nu)$ where $\pi(\mathbf{x}, \mathbf{y}) > 0$. A map can then be obtained by simply averaging over these $\mathbf{y}$ according to the weights $\pi(\mathbf{x}, \mathbf{y})$.

Definition 3. *(Barycentric projection) Let π be a solution of the OT problem (3.2.2) or regularized OT problem (3.2.3). The barycentric projection $\bar{\pi}$ w.r.t. a convex cost $d : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ is defined as,*

$$\bar{\pi}(\mathbf{x}) = \arg\min_{\mathbf{z}} \mathbb{E}_{Y \sim \pi(\cdot|\mathbf{x})} \left[d(\mathbf{z}, Y) \right]. \tag{3.4.1}$$

Algorithm 2 Optimal map learning with SGD

Inputs: input measures μ, ν; cost function c; dual optimal variables u and v; map f_θ parameterized as a deep NN; batch size n; learning rate γ.
while not converged **do**
  sample a batch $(\mathbf{x}_1, \cdots, x_n)$ from μ
  sample a batch $(\mathbf{y}_1, \cdots, y_n)$ from ν
  update $\theta \leftarrow \theta - \gamma \sum_{ij} H_\varepsilon(\mathbf{x}_i, \mathbf{y}_j) \nabla_\theta d(y_j, f_\theta(\mathbf{x}_i))$
end while

In the special case $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$, Eq. (3.4.1) has the closed form solution $\bar{\pi}(\mathbf{x}) = \mathbb{E}_{Y \sim \pi(\cdot|\mathbf{x})}[Y]$. In a discrete setting, with $\mathbf{y} = (\mathbf{y}_1, \cdots, \mathbf{y}_n)$ and a the weights of μ, we have hence $\bar{\pi} = \frac{\pi \mathbf{y}^t}{a}$. Moreover, for the specific squared Euclidean cost $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$, the barycentric projection $\bar{\pi}$ w.r.t. that cost (i.e. $d(\mathbf{z}, \mathbf{y}) = \|\mathbf{z} - \mathbf{y}\|_2^2$) is an *optimal map* [Ambrosio et al., 2006][Theorem 12.4.4], i.e. $\bar{\pi}$ is a solution to the Monge problem (3.2.1) between the source measure μ and the target measure $\bar{\pi}\#\mu$. Hence the barycentric projection w.r.t. the squared Euclidean cost is often used as a simple way to recover optimal maps from optimal transport plans [Reich, 2013, Wang et al., 2013, Ferradans et al., 2014, Seguy and Cuturi, 2015].

Formula (3.4.1) provides a pointwise value of the barycentric projection. When μ is discrete, this means that we only have mapping estimations for a finite number of points. In order to define a map which is defined everywhere, we parameterize the barycentric projection as a deep neural network. We show in the next paragraph how to efficiently learn its parameters.

3.4.1 Optimal map learning

An estimation f of the barycentric projection of a regularized plan π^ε which is defined even outside the support of μ can be obtained by learning a deep neural network which minimizes the following objective w.r.t. the parameters θ,

$$\begin{aligned} \mathbb{E}_{X \sim \mu}\left[\mathbb{E}_{Y \sim \pi^\varepsilon(\cdot|X)}\left[d(Y, f_\theta(X))\right]\right] &= \mathbb{E}_{(X,Y) \sim \pi^\varepsilon}\left[d(Y, f_\theta(X))\right] \\ &= \mathbb{E}_{(X,Y) \sim \mu \times \nu}\left[d(Y, f_\theta(X)) H_\varepsilon(X, Y)\right]. \end{aligned} \tag{3.4.2}$$

When $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$, the last term in Eq. (3.4.2) is simply a weighted sum of squared errors, with possibly an infinite number of terms whenever μ or ν are continuous. We propose to minimize the objective (3.4.2) by stochastic gradient descent, which provides the simple Algorithm 2. The OT problem being symmetric, we can also compute the opposite barycentric projection g w.r.t. a cost $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ by minimizing $\mathbb{E}_{(X,Y) \sim \mu \times \nu}\left[d(g(Y), X) H_\varepsilon(X, Y)\right]$.

However, unless the plan π is induced by a map, the averaging process results in having the image of the source measure by $\bar{\pi}$ only approximately equal to the target measure ν. Still, when the size of discrete measure is large and the regularization is small, we show in the next paragraph that 1) the barycentric projection of a regularized OT plan is close to the Monge map between the underlying continuous measures (Theorem 3) and 2) the image of the source measure by this barycentric projection should be close to the target measure ν (Corollary 1).

3.4.2 Theoretical guarantees

As stated earlier, when $\mathcal{X} = \mathcal{Y}$ and $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$, Brenier [1991] proved that when the source measure μ is continuous, there exists a solution to the Monge problem (3.2.1). This result was generalized to more general cost functions, see [Villani, 2008][Corollary 9.3] for details. In that case, the plan π between μ and ν is written as $(\mathrm{id}, f)\#\mu$ where f is the Monge map. Now considering discrete measures μ_n and ν_n which converge to μ (continuous) and ν respectively, we have proved in Theorem 2 that π_n^ε converges weakly to $\pi = (\mathrm{id}, f)\#\mu$ when $(n, \varepsilon) \to (\infty, 0)$. The next theorem, proved in the Appendix, shows that the barycentric projection $\bar{\pi}_n^\varepsilon$ also converges weakly to the true Monge map between μ and ν, justifying our approach.

Theorem 3. *Let μ be a continuous probability measure on $\mathbb{R}^d$, and ν an arbitrary probability measure on $\mathbb{R}^d$ and c a cost function satisfying [Villani, 2008][Corollary 9.3]. Let $\mu_n = \frac{1}{n}\sum_{i=1}^n \delta_{\mathbf{x}_i}$ and $\nu_n = \frac{1}{n}\sum_{j=1}^n \delta_{\mathbf{y}_j}$ converging weakly to μ and ν respectively. Assume that the OT solution π_n of Problem (3.2.2) between μ_n and ν_n is unique for all n. Let (ε_n) a sequence of non-negative real numbers converging sufficiently fast to 0 and $\bar{\pi}_n^{\varepsilon_n}$ the barycentric projection w.r.t. the convex cost $d = c$ of the solution $\pi_n^{\varepsilon_n}$ of the entropy-regularized OT problem (3.2.3). Then, up to extraction of a subsequence,*

$$(id, \bar{\pi}_n^{\varepsilon_n})\#\mu_n \to (id, f)\#\mu \quad weakly, \tag{3.4.3}$$

where f is the solution of the Monge problem (3.2.1) between μ and ν.

This theorem shows that our estimated barycentric projection is close to an optimal map between the underlying continuous measures for n big and ε small. The following corollary confirms the intuition that the image of the source measure by this map converges to the underlying target measure.

Corollary 1. *With the same assumptions as above, $\bar{\pi}_n^{\varepsilon_n}\#\mu_n \to \nu$ weakly.*

In terms of random variables, the last equation states that if $X_n \sim \mu_n$ and $Y \sim \nu$, then $\bar{\pi}_n^{\varepsilon_n}(X_n)$ converges in distribution to Y.

These theoretical results show that our estimated Monge map can thus be used to perform domain adaptation by mapping a source dataset to a target dataset, as well as perform generative modeling by mapping a continuous measure to a target discrete dataset. We demontrate this in the following section.

3.5 Numerical Experiments

3.5.1 Dual vs semi-dual speed comparisons

We start by evaluating the training time of our dual stochastic algorithm 1 against a stochastic semi-dual approach similar to [Genevay et al., 2016]. In the semi-dual approach, one of the dual variable is eliminated and is computed in closed form. However, this computation has $\mathcal{O}(n)$ complexity where n is the size of the target measure ν. We compute the regularized OT objective with both methods on a spectral transfer problem, which is related to the color transfer problem [Reinhard et al., 2001, Pitié et al., 2007], but where images are multispectral, *i.e.* they share a finer sampling of the light wavelength. We take two 500×500 images from the CAVE dataset [Yasuma et al., 2010] that have 31 spectral bands. As such, the optimal transport problem is computed on two empirical distributions of 250000 samples in $\mathbb{R}^{31}$ on which we consider the squared Euclidean ground cost c. The timing evolution of train losses are

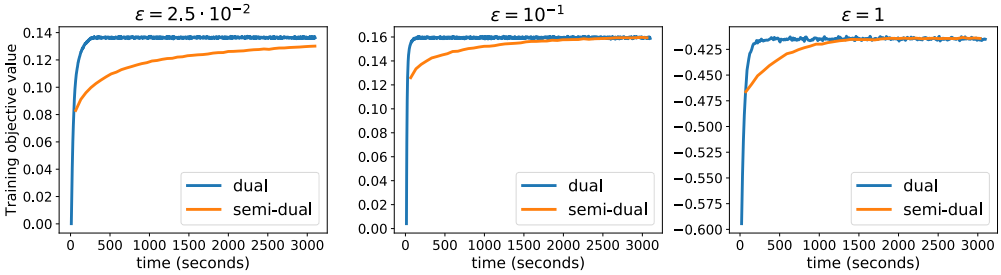

FIGURE 3.2: Convergence plots of the the Stochastic Dual Algorithm 1 against a stochastic semi-dual implementation (adapted from [Genevay et al., 2016]: we use SGD instead of SAG), for several entropy-regularization values. Learning rates are $\{5., 20., 20.\}$ and batch sizes $\{1024, 500, 100\}$ respectively and are taken the same for the dual and semi-dual methods.

reported in Figure 3.2 for three different regularization values $\varepsilon = \{0.025, 0.1, 1.\}$. In the three cases, one can observe that convergence of our proposed dual algorithm is much faster.

3.5.2 Large scale domain adaptation

We apply here our computation framework on an unsupervised domain adaptation (DA) task, for which optimal transport has shown to perform well on small scale datasets [Courty et al., 2017b, Perrot et al., 2016, Courty et al., 2014]. This restriction is mainly due to the fact that those works only consider the primal formulation of the OT problem. Our goal here is not to compete with the state-of-the-art methods in domain adaptation but to assess that our formulation allows to scale optimal transport based domain adaptation (OTDA) to large datasets. OTDA is illustrated in Fig. 3.3 and follows two steps: 1) learn an optimal map between the source and target distribution, 2) map the source samples and train a classifier on them in the target domain. Our formulation also allows to use any differentiable ground cost c while [Courty et al., 2017b] was limited to the squared Euclidean distance.

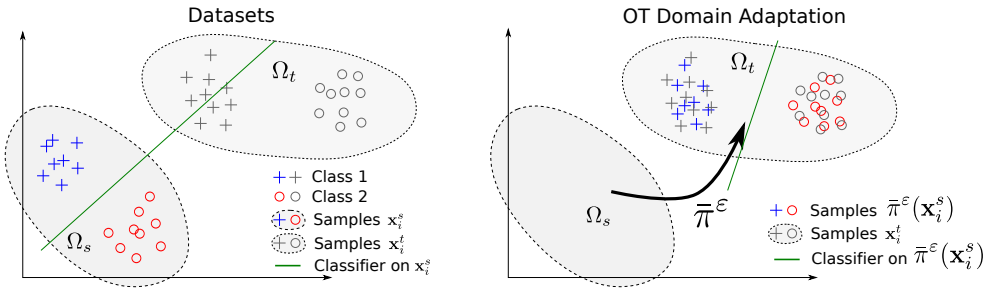

FIGURE 3.3: Illustration of the OT Domain Adaptation method adapted from [Courty et al., 2017b]. Source samples are mapped to the target set through the barycentric projection $\bar{\pi}^\varepsilon$. A classifier is then learned on the mapped source samples.

Datasets We consider the three cross-domain digit image datasets MNIST [Lecun et al., 1998], USPS, and SVHN [Netzer et al., 2011], which have 10 classes each. For the adaptation between MNIST and USPS, we use 60000 samples in the MNIST domain and 9298 samples in USPS domain. MNIST images are resized to the same resolution as USPS ones (16×16). For the adaptation between SVHN and MNIST,

TABLE 3.1: Results (accuracy in %) on domain adaptation among MNIST, USPS and SVHN datasets with entropy (R_e) and L2 (R_{L^2}) regularizations. *Source only* refers to 1-NN classification between source and target samples without adaptation.

Method	MNIST→ USPS	USPS→ MNIST	SVHN → MNIST
Source only	73.47	36.97	54.33
Bar. proj. OT	57.75	52.46	intractable
Bar. proj. OT with R_e	68.75	57.35	intractable
Bar. proj. Alg. 1 with R_e	68.84	57.55	58.87
Bar. proj. Alg. 1 with R_{L^2}	67.8	57.47	60.56
Monge map Alg. 1+2 with R_e	**77.92**	60.02	61.11
Monge map Alg. 1+2 with R_{L^2}	72.61	**60.50**	**62.88**

we use 73212 samples in the SVHN domain and 60000 samples in the MNIST domain. MNIST images are zero-padded to reach the same resolution as SVHN (32×32) and extended to three channels to match SVHN image sizes. The labels in the target domain are withheld during the adaptation. In the experiment, we consider the adaptation in three directions: MNIST → USPS, USPS → MNIST, and SVHN → MNIST.

Methods and experimental setup Our goal is to demonstrate the potential of the proposed method in large-scale settings. Adaptation performance is evaluated using a 1-nearest neighbor (1-NN) classifier, since it has the advantage of being parameter free and allows better assessment of the quality of the adapted representation, as discussed in [Courty et al., 2017b]. In all experiments, we consider the 1-NN classification as a baseline, where labeled neighbors are searched in the source domain and the accuracy is computed on target data. We compare our approach to previous OTDA methods where an optimal map is obtained through the discrete barycentric projection of either an optimal plan (computed with the network simplex algorithm[1]) or an entropy-regularized optimal plan (computed with the Sinkhorn algorithm [Cuturi, 2013]), whenever their computation is tractable. Note that these methods do not provide out-of-sample mapping. In all experiments, the ground cost c is the squared Euclidean distance and the barycentric projection is computed w.r.t. that cost. We learn the Monge map of our proposed approach with either entropy or L2 regularizations. Regarding the adaptation between SVHN and MNIST, we extract deep features by learning a modified LeNet architecture on the source data and extracting the 100-dimensional features output by the top hidden layer. Adaptation is performed on those features. We report for all the methods the best accuracy over the hyperparameters on the target dataset. While this setting is unrealistic in a practical DA application, it is widely used in the DA community [Long et al., 2013] and our goal is here to investigate the relative performances of large-scale OTDA in a fair setting.

Hyper-parameters and learning rate The value for the regularization parameter is set in $\{5, 2, 0.9, 0.5, 0.1, 0.05, 0.01\}$. Adam optimizer with batch size 1000 is used to optimize the network. The learning rate is varied in $\{2, 0.9, 0.1, 0.01, 0.001, 0.0001\}$. The learned Monge map f in Alg. 2 is parameterized as a neural network with two fully-connected hidden layers ($d \to 200 \to 500 \to d$) and ReLU activations, and the weights are optimized using the Adam optimizer with learning rate equal to 10^{-4} and batch size equal to 1000. For the Sinkhorn algorithm, regularization value is chosen from $\{0.01, 0.1, 0.5, 0.9, 2.0, 5.0, 10.0\}$.

[1] http://liris.cnrs.fr/~nbonneel/FastTransport/

Results Results are reported in Table 3.1. In all cases, our proposed approach outperforms previous OTDA algorithms. On MNIST→USPS, previous OTDA methods perform worse than using directly source labels, whereas our method leads to successful adaptation results with 20% and 10% accuracy points over OT and regularized OT methods respectively. On USPS→MNIST, all three algorithms lead to successful adaptation results, but our method achieves the highest adaptation results. Finally, on the challenging large-scale adaptation task SVHN→MNIST, only our method is able to handle the whole datasets, and outperforms the source only results.

Comparing the results between the barycentric projection and estimated Monge map illustrates that learning a parametric mapping provides some kind of regularization, and improves the performance.

3.5.3 Generative optimal transport (GOT)

Approach Corollary 1 shows that when the support of the discrete measures μ and ν is large and the regularization ε is small, then we have approximately $\bar{\pi}^{\varepsilon}\#\mu = \nu$. This observation motivates the use of our Monge map estimation as a generator between an arbitrary continuous measure μ and a discrete measure ν representing the discrete distribution of some dataset. We can thus obtain a generative model by first computing regularized OT through Alg. 1 between a Gaussian measure μ and a discrete dataset ν and then compute our generator with Alg. 2. This requires to have a cost function between the latent variable $X \sim \mu$ and the discrete variable $Y \sim \nu$. The property we gain compared to other generative models is that our generator is, at least approximately, an *optimal map* w.r.t. this cost. In our case, the Gaussian is taken with the same dimensionality as the discrete data and the squared Euclidean distance is used as ground cost c.

Permutation-invariant MNIST We preprocess MNIST data by rescaling grayscale values in $[-1, 1]$. We run Alg. 1 and Alg. 2 where μ is a Gaussian whose mean and covariance are taken equal to the empirical mean and covariance of the preprocessed MNIST dataset; we have observed that this makes the learning easier. The target discrete measure ν is the preprocessed MNIST dataset. Permutation invariance means that we consider each grayscale 28×28 images as a 784-dimensional vector and do not rely on convolutional architectures. In Alg. 1 the dual potential u is parameterized as a $(d \to 1024 \to 1024 \to 1)$ fully-connected NN with ReLU activations for each hidden layer, and the L^2 regularization is considered as it produced experimentally less blurring. The barycentric projection f of Alg. 2 is parameterized as a $(d \to 1024 \to 1024 \to d)$ fully-connected NN with ReLU activation for each hidden layer and a tanh activation on the output layer. We display some generated samples in Fig. 3.4.

3.6 Conclusion

We proposed two original algorithms that allow for *i)* large-scale computation of regularized optimal transport *ii)* learning an optimal map that moves one probability distribution onto another (the so-called *Monge map*). To our knowledge, our approach introduces the first tractable algorithms for computing both the regularized OT objective and optimal maps in large-scale or continuous settings. We believe that these two contributions enable a wider use of optimal transport strategies in machine learning applications. Notably, we have shown how it can be used in an

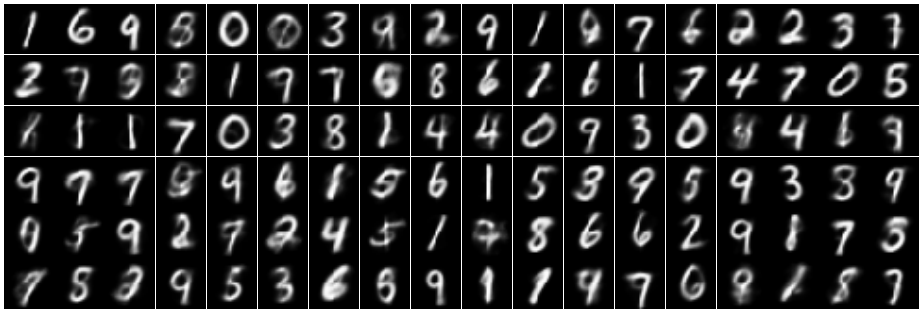

FIGURE 3.4: Samples generated by our optimal generator learned through Algorithms 1 and 2.

unsupervised domain adaptation setting, or in generative modeling, where a Monge map acts directly as a generator. Our consistency results show that our approach is theoretically well-grounded. An interesting direction for future work is to investigate the corresponding convergence rates of the empirical regularized optimal plans. We believe this is a very complex problem since technical proofs regarding convergence rates of the empirical OT objective used e.g. in [Sriperumbudur et al., 2012, Boissard et al., 2014, Fournier and Guillin, 2015] do not extend simply to the optimal transport plans.

3.7 Appendix

3.7.1 Proofs

Proof of Theorem 2.

Proof. Let π_n be the solution of the OT problem (3.2.2) between μ_n and ν_n which has maximum entropy (i.e. the maximizer of $\sum_{ij} \pi_{ij} \ln(\pi_{ij})$ over the set of optimal plans). A result about stability of optimal transport [Villani, 2008][Theorem 5.20] states that, up to extraction of a subsequence, π_n converges weakly to a solution π of the OT problem between μ and ν (regardless of π_n being the solution with maximum entropy or not). We still write (π_n) this subsequence, as well as $(\pi_n^{\varepsilon_n})$ the corresponding subsequence.
Let $g \in \mathcal{C}_b(\mathcal{X} \times \mathcal{Y})$ a bounded continuous function on $\mathcal{X} \times \mathcal{Y}$. We have,

$$\int_{\mathcal{X}\times\mathcal{Y}} g d\pi_n^{\varepsilon_n} - \int_{\mathcal{X}\times\mathcal{Y}} g d\pi = \left(\int_{\mathcal{X}\times\mathcal{Y}} g d\pi_n^{\varepsilon_n} - \int_{\mathcal{X}\times\mathcal{Y}} g d\pi_n\right) + \left(\int_{\mathcal{X}\times\mathcal{Y}} g d\pi_n - \int_{\mathcal{X}\times\mathcal{Y}} g d\pi\right) \tag{3.7.1}$$

The second term in the right-hand side converges to 0 as a result of the previously mentioned stability of optimal transport [Villani, 2008][Theorem 5.20]. We now show the convergence of the first term to 0 when $\varepsilon_n \to 0$ sufficiently fast. We have

$$\begin{aligned}\left|\int_{\mathcal{X}\times\mathcal{Y}} g d\pi_n^{\varepsilon_n} - \int_{\mathcal{X}\times\mathcal{Y}} g d\pi_n\right| &= \left|\sum_{i=1,n\ j=1,n} g(\mathbf{x}_i, \mathbf{y}_j)\pi_n^{\varepsilon_n}(\mathbf{x}_i, \mathbf{y}_j) - \sum_{i=1,n\ j=1,n} g(\mathbf{x}_i, \mathbf{y}_j)\pi_n(\mathbf{x}_i, \mathbf{y}_j)\right| \\ &\leqslant M_g \sum_{ij} \left|\pi_n^{\varepsilon_n}(\mathbf{x}_i, \mathbf{y}_j) - \pi_n(\mathbf{x}_i, \mathbf{y}_j)\right| \\ &= M_g \|\pi_n^{\varepsilon_n} - \pi_n\|_{\mathbb{R}^{n\times n},1}\end{aligned} \tag{3.7.2}$$

where M_g is an upper-bound of g. Since π_n is the optimal plan with maximum entropy, a convergence result by Cominetti and San Martín [1994] shows that there exists positive constants (w.r.t. ε_n) M_{c_n,μ_n,ν_n} and $\lambda_{c_n,\mu_n,\nu_n}$ such that

$$\|\pi_n^{\varepsilon_n} - \pi_n\|_{\mathbb{R}^{n\times n},1} \leqslant M_{c_n,\mu_n,\nu_n} e^{-\frac{\lambda_{c_n,\mu_n,\nu_n}}{\varepsilon_n}} \tag{3.7.3}$$

where $c_n = (c(\mathbf{x}_1, \mathbf{y}_1), \cdots, c(\mathbf{x}_n, \mathbf{y}_n))$. The subscript indices indicate the dependences of each constant. Hence, we see that choosing any (ε_n) such that the right-hand side of Eq. (3.7.3) tends to 0 provides the results. In particular, we can take

$$\varepsilon_n = \frac{\lambda_{c_n,\mu_n,\nu_n}}{\ln(nM_{c_n,\mu_n,\nu_n})} \tag{3.7.4}$$

which suffices to have the convergence of (3.7.2) to 0 for any bounded continuous function $g \in \mathcal{C}_b(\mathcal{X} \times \mathcal{Y})$. This proves the weak convergence of $\pi_n^{\varepsilon_n}$ to π. □

Proof of Theorem 3.

Proof. First, note that the existence of a Monge map between μ and ν follows from the absolute continuity of μ and the assumptions on the cost functions c [Villani, 2008][Corollary 9.3].
Let $g \in \mathcal{C}_l(\mathbb{R}^d \times \mathbb{R}^d)$ a Lipschitz function on $\mathbb{R}^d \times \mathbb{R}^d$. Let π_n be the unique (by

assumption) solution of the OT problem between μ_n and ν_n. We have

$$\int_{\mathbb{R}^d\times\mathbb{R}^d} gd(\mathrm{id},\bar{\pi}_n^{\varepsilon_n})\#\mu_n - \int_{\mathbb{R}^d\times\mathbb{R}^d} gd(\mathrm{id},f)\#\mu = \left(\int_{\mathbb{R}^d\times\mathbb{R}^d} gd(\mathrm{id},\bar{\pi}_n^{\varepsilon_n})\#\mu_n - \int_{\mathbb{R}^d\times\mathbb{R}^d} gd(\mathrm{id},\bar{\pi}_n)\#\mu_n\right) + \left(\int_{\mathbb{R}^d\times\mathbb{R}^d} gd(\mathrm{id},\bar{\pi}_n)\#\mu_n - \int_{\mathbb{R}^d\times\mathbb{R}^d} gd(\mathrm{id},f)\#\mu\right) \tag{3.7.5}$$

Since μ_n and ν_n are uniform discrete probability measures supported on the same number of points, we know by [Birkhoff, 1946] that the optimal transport π_n is actually an optimal assignment T_n, so that we have $\pi_n = (\mathrm{id}, T_n)\#\mu_n$. This also implies $\bar{\pi}_n = T_n$ so that $(\mathrm{id},\bar{\pi}_n)\#\mu_n = (\mathrm{id}, T_n)\#\mu_n$. Hence, the second term in the right-hand side of (3.7.5) converges to 0 as a result of the stability of optimal transport [Villani, 2008][Theorem 5.20]. Now, we show that the first term also converges to 0 for ε_n converging sufficiently fast to 0. By definition of the pushforward operator,

$$\int_{\mathbb{R}^d\times\mathbb{R}^d} gd(\mathrm{id},\bar{\pi}_n^{\varepsilon_n})\#\mu_n - \int_{\mathbb{R}^d\times\mathbb{R}^d} gd(\mathrm{id},\bar{\pi}_n)\#\mu_n = \int_{\mathbb{R}^d} g(\mathbf{x},\bar{\pi}_n^{\varepsilon_n}(\mathbf{x})d\mu_n(\mathbf{x}) - \int_{\mathbb{R}^d} g(\mathbf{x},T_n(\mathbf{x}))d\mu_n(\mathbf{x}) \tag{3.7.6}$$

and we can bound

$$\begin{aligned}\left|\int_{\mathbb{R}^d} g(\mathbf{x},\bar{\pi}_n^{\varepsilon_n}(\mathbf{x}))d\mu_n(\mathbf{x}) - \int_{\mathbb{R}^d} g(\mathbf{x},T_n(\mathbf{x}))d\mu_n(\mathbf{x})\right| &= \left|\frac{1}{n}\sum_{i=1}^n g(\mathbf{x}_i,\bar{\pi}_n^{\varepsilon_n}(\mathbf{x}_i)) - \frac{1}{n}\sum_{i=1}^n g(\mathbf{x}_i,T_n(\mathbf{x}_i))\right| \\ &\leqslant \sum_i K_g\|\bar{\pi}_n^{\varepsilon_n}(\mathbf{x}_i) - T_n(\mathbf{x}_i)\|_{\mathbb{R}^d,2} \\ &= nK_g\|\pi_n^{\varepsilon_n}Y_n - \pi_n Y_n\|_{\mathbb{R}^{n\times n},2} \\ &\leqslant nK_g\|\pi_n^{\varepsilon_n} - \pi_n\|^{1/2}_{\mathbb{R}^{n\times n},2}\|Y_n\|^{1/2}_{\mathbb{R}^{n\times d},2}\end{aligned} \tag{3.7.7}$$

where $Y_n = (\mathbf{y}_1,\cdots,y_n)^t$ and K_g is the Lipschitz constant of g. The first inequality follows from g being Lipschitz. The next equality follows from the discrete closed form of the barycentric projection. The last inequality is obtained through the Cauchy-Schwartz inequality. We can now follow the same reasoning as done in the previous proof. Since π_n is the optimal plan with maximum entropy, a convergence result by Cominetti and San Martín [1994] shows that there exists positive constants (w.r.t. ε_n) M_{c_n,μ_n,ν_n} and $\lambda_{c_n,\mu_n,\nu_n}$ such that

$$\|\pi_n^{\varepsilon_n} - \pi_n\|^{1/2}_{\mathbb{R}^{n\times n},2} \leqslant M_{c_n,\mu_n,\nu_n}e^{-\frac{\lambda_{c_n,\mu_n,\nu_n}}{\varepsilon_n}} \tag{3.7.8}$$

where $c_n = (c(\mathbf{x}_1,\mathbf{y}_1),\cdots,c(\mathbf{x}_n,\mathbf{y}_n))$. The subscript indices indicate the dependences of each constant. Hence, we see that choosing any (ε_n) such that (3.7.8) tends to 0 provides the results. In particular, we can take

$$\varepsilon_n = \frac{\lambda_{c_n,\mu_n,\nu_n}}{\ln(n^2\|Y_n\|^{1/2}_{\mathbb{R}^{n\times d},2}M_{c_n,\mu_n,\nu_n})} \tag{3.7.9}$$

which suffices to have the convergence of (3.7.2) to 0 for Lipschitz function $g \in \mathcal{C}_l(\mathbb{R}^d\times\mathbb{R}^d)$. By the Portmanteau theorem, this proves the weak convergence of $(\mathrm{id},\bar{\pi}_n^{\varepsilon_n})\#\mu_n$ to $(\mathrm{id},f)\#\mu$. □

Proof of Corollary 1.

Proof. Let $h \in \mathcal{C}_b(\mathbb{R}^d)$ a bounded continuous function. Let $g \in \mathcal{C}_b(\mathbb{R}^d \times \mathbb{R}^d)$ defined as $g : (\mathbf{x}, \mathbf{y}) \mapsto h(\mathbf{y})$. We have,

$$\int_{\mathbb{R}^d} h d\bar{\pi}_n^{\varepsilon_n} \# \mu_n - \int_{\mathbb{R}^d} h d f \# \mu = \int_{\mathbb{R}^d \times \mathbb{R}^d} g d(\mathrm{id}, \bar{\pi}_n^{\varepsilon_n}) \# \mu_n - \int_{\mathbb{R}^d \times \mathbb{R}^d} g d(\mathrm{id}, f) \# \mu \quad (3.7.10)$$

which converges to 0 by Theorem (3). Since $f \# \mu = \nu$, this proves the corollary. □

Chapter 4

Principal Geodesic Analysis in the Wasserstein Space: The Real Line

4.1 Introduction

Most data sets describe multivariate data, namely vectors of relevant features that can be modeled as random elements sampled from an unknown distribution. In that setting, Principal Component Analysis (PCA) is certainly the simplest and most widely used approach to reduce the dimension of such data sets. We consider in this chapter the statistical analysis of data sets whose elements are histograms supported on the real line, while extensions to the general case of probability measures supported on the d-dimensional Euclidean space is addressed in the next chapter. Just as with PCA, our main goal in that setting is to compute the principal modes of variation of histograms around their mean element and therefore facilitate the visualization of such data sets. However, since the number, size or locations of significant bins in the histograms of interest may vary from one histogram to another, using standard PCA on histograms (with respect to the Euclidean metric) is bound to fail (see for instance Figure 4.1).

In this study, we propose to use the 2-Wasserstein metric [Villani, 2003, §7.1] to measure the distance between histograms, and to compute their modes of variation accordingly. In our approach, histograms are seen as piecewise constant probability density functions (pdf) supported on a given interval Ω of the real line. In this setting, the variability in a set of histograms can be analyzed via the notion of Geodesic PCA (GPCA) of probability measures in the Wasserstein space $W_2(\Omega)$ admitting these histograms as pdf. That approach has been recently proposed in the statistics and machine learning literature in [Bigot et al., 2015] for probability measures on the real line, and in [Seguy and Cuturi, 2015, Wang et al., 2013] for discrete probability measures on $\mathbb{R}^d$ (the approach of [Seguy and Cuturi, 2015] is the topic of the next chapter). However, implementing GPCA remains a challenging computational task even in the simplest case of pdf's supported on $\mathbb{R}$. The purpose of this chapter is to provide a fast algorithm to perform GPCA of probability measures supported on the real line, and to compare its performances with log-PCA, namely standard PCA in the tangent space at the Wasserstein barycenter of the data [Fletcher et al., 2004, Petersen and Müller, 2016].

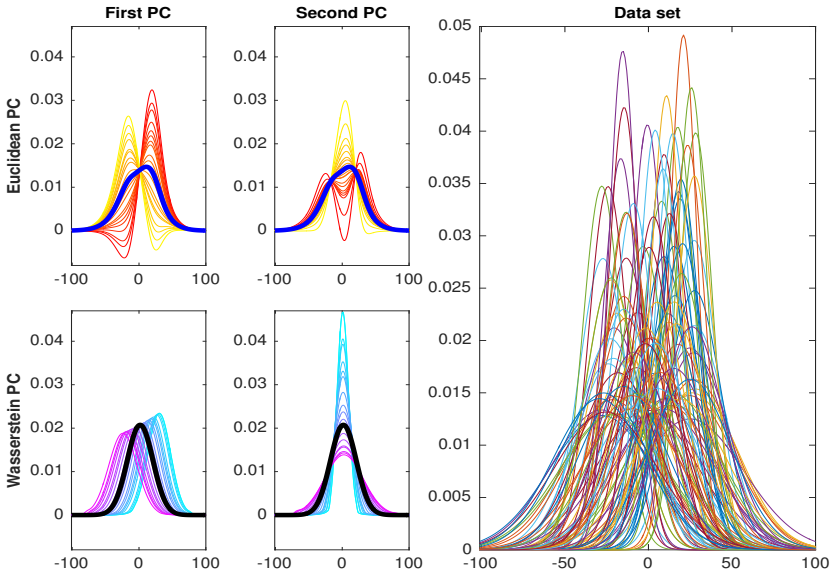

FIGURE 4.1: Synthetic example. (Right) A data set of $n = 100$ Gaussian histograms randomly translated and scaled. (Top-left) Standard PCA of this data set with respect to the Euclidean metric. The Euclidean barycenter of the data set is depicted in blue. (Bottom-left) Geodesic PCA with respect to the Wasserstein metric using the iterative geodesic algorithm (4.4.1). The black curve represents the density of the Wasserstein barycenter. Colors encode the progression of the pdf of principal geodesic components in $W_2(\Omega)$.

4.1.1 Related results

Foundations of Geodesic PCA in the Wasserstein space The space of probability measures (with finite second moment) endowed with the 2-Wasserstein distance is not a Hilbert space. Therefore, standard PCA, which involves computing a covariance matrix, cannot be applied directly to compute principal mode of variations in a Wasserstein sense. Nevertheless, a meaningful notion of PCA can still be defined by relying on the pseudo-Riemannian structure of the Wasserstein space, which was extensively studied in [Ambrosio et al., 2006] and [Ambrosio et al., 2006]. Following this principle, a framework for GPCA of probability measures supported on a interval $\Omega \subset \mathbb{R}$ was introduced in [Bigot et al., 2015]. GPCA is defined as the problem of estimating a principal geodesic subspace (of a given dimension) which maximizes the variance of the projection of the data to that subspace. In that approach the base point of that subspace is the Wasserstein barycenter of the data as introduced in [Agueh and Carlier, 2011], which is also known as a Fréchet mean in the general context of metric spaces. Existence, consistency and a detailed characterization of GPCA in $W_2(\Omega)$ were studied in [Bigot et al., 2015]. In particular, the authors have shown that this approach is equivalent to map the data in the tangent space of $W_2(\Omega)$ at the Fréchet mean, and then to perform a PCA in this Hilbert space that is constrained to lie in a convex and closed subset of functions. Mapping the data to this tangent space is not difficult in the one-dimensional case as it amounts to computing a set of optimal maps between the data and their Wasserstein barycenter, for which a closed form is available using their quantile functions (see for example [Villani, 2003, §2.2]). To perform PCA on the mapped data, [Bigot et al., 2015] fell short of proposing an algorithm to minimize that problem, which has a non-convex

and non-differentiable objective function as well as involved constraints. Only a numerical approximation to the computation of GPCA was proposed in [Bigot et al., 2015], which amounts to applying log-PCA, namely a standard PCA of the data set mapped beforehand to the tangent space of $W_2(\Omega)$ at its Fréchet mean.

Previous work in the one-dimensional case PCA of histograms with respect to the Wasserstein metric has also been proposed in [Verde et al., 2015] in the context of symbolic data analysis. Their approach consists in computing a standard PCA in the Hilbert space $L^2([0,1])$ of the quantile functions associated to the histograms. Therefore, the algorithm in [Verde et al., 2015] corresponds to log-PCA of probability measures as suggested in [Bigot et al., 2015], but it does not solve the problem of convex-constrained PCA in a Hilbert space associated to an exact GPCA in $W_2(\Omega)$. A related problem, which can be referred to as geodesic regression (considered in [Fletcher, 2011, 2013] for data on a Riemannian manifold), has been considered by Jiang et al. in [Jiang et al., 2012] where the authors fit a single geodesic g_t to indexed histograms in order to model nonstationary time series. In the problem of finding principal geodesics, we do not assume that the data set is indexed.

PGA and log-PCA on Riemannian manifolds The method of GPCA proposed in [Bigot et al., 2015] clearly shares similarities with analogs of PCA for data belonging to a Riemannian manifold $\mathcal{M}$ of finite dimension. These methods, generally referred to as Principal Geodesic Analysis (PGA), extend the notion of classical PCA in Euclidean spaces for the purpose of analyzing data belonging to curved Riemannian manifolds (see e.g. [Fletcher et al., 2004, Sommer et al., 2010]). This generalization of PCA proceeds by replacing Euclidean concepts of vector means, lines and orthogonality by the more general notions in Riemannian manifolds of Fréchet mean, geodesics, and orthogonality in tangent spaces.

In [Fletcher et al., 2004], linearized PGA, which we refer to as log-PCA, is defined as follows. In a first step, data are mapped to the tangent space $T_{\bar{x}}\mathcal{M}$ at their Fréchet mean $\bar{x}$ by applying the logarithmic map $\log_{\bar{x}}$ to each data point. Then, in a second step, standard PCA in the Euclidean space $T_{\bar{x}}\mathcal{M}$ can be applied. This provides a family of orthonormal tangent vectors. Principal components of variation in $\mathcal{M}$ can then be defined by back-projection of these tangent vectors on $\mathcal{M}$ by using the exponential map at $\bar{x}$, that is known to parameterize geodesics at least locally. Log-PCA has low computational cost, but this comes at the expense of two simplifications and drawbacks:

(1) First, log-PCA amounts to substituting geodesic distances between data points by the linearized distance in $T_{\bar{x}}\mathcal{M}$, which may not always be a good approximation because of the curvature of $\mathcal{M}$, see e.g. [Sommer et al., 2010].

(2) Secondly, the exponential map at the Fréchet mean parameterizes geodesics only locally, which implies that principal components in $\mathcal{M}$ obtained with log-PCA may not be geodesic along the typical range of the data set.

Numerical approaches to GPCA and log-PCA in the Wasserstein space Computational methods have been introduced in [Wang et al., 2013] or [Seguy and Cuturi, 2015] (which we describe in the next chapter) to extend the concepts of PGA on Riemannian manifolds to that of the space $W_2(\mathbb{R}^d)$ of probability measures supported on $\mathbb{R}^d$ endowed with the Wasserstein metric. [Wang et al., 2013] propose to compute a notion of template measure (using k-means clustering) of a set of discrete

probability measures, and to consider then the optimal transport plans from that template measure to each measure in the data set. Computation of the barycentric projection of each optimal transport plan leads to a set of Monge maps over which a standard PCA can be applied, resulting in an orthonormal family of tangent vectors defined on the support of the template measure. Principal components of variation in $\mathbb{R}^d$ can then be obtained through the pushforward operator, namely by moving the mass along these tangent vectors. This approach, analog to log-PCA on Riemannian manifolds, suffers from the main drawbacks mentioned above: for $d > 1$, the linearized Wasserstein distance may be a crude approximation of the Wasserstein distance, and there is no guarantee that the computed tangent vectors parameterize geodesics of sufficient length to summarize most of the variability in the data set. Losing geodesicity means that the principal components are curves in $W_2(\mathbb{R}^d)$ along which the mass may not be transported optimally, which may significantly reduce the interpretability of these principal components.

4.1.2 Main contributions

We focus on computing an exact GPCA on probability measures supported on $\Omega \subset \mathbb{R}$. The Wasserstein space of probability measures supported on the real line has the advantage that the linearized Wasserstein distance in the tangent space is equal to the Wasserstein distance in the space $W_2(\Omega)$. The main challenge is thus to obtain principal curves which are geodesics along the range of the data set.

The first contribution of this chapter is to propose two fast algorithms for GPCA in $W_2(\Omega)$. The first algorithm finds iteratively geodesics such that the Wasserstein distance between the data set and the parameterized geodesic is minimized with respect to $W_2(\Omega)$. We show that the global minimum of our objective function for the first principal geodesic curve corresponds indeed to the solution of the exact GPCA problem defined in [Bigot et al., 2015]. While this algorithm is able to find iteratively orthogonal principal geodesics, there is no guarantee that several principal geodesics parameterize a surface which is also geodesic. This is the reason we also propose a second algorithm which computes all the principal geodesics at once by parameterizing a geodesic surface as a convex combination of optimal velocity fields and relaxing the orthogonality constraint between principal geodesics. Both algorithms are a variant of the proximal forward-backward algorithm. They converge to a stationary point of the objective function, as shown by recent results in nonconvex optimization based on proximal methods [Attouch et al., 2013, Ochs et al., 2014]. Our second contribution is a numerical comparison of log-PCA in $W_2(\Omega)$, as done in [Bigot et al., 2015] (for $d = 1$) or [Wang et al., 2013], with our approach which solves the exact Wasserstein GPCA problem.

In all our experiments, data are normalized in order to have a suitable representation as probability measures. We believe this preprocessing does not affect any useful properties of the histogram data sets considered in the present chapter, in the same way as centering or whitening are often used as a preprocessing step in many data-analysis tasks. Yet, if the total mass of a given histogram matters for some application, we could consider the use of unbalanced optimal transport [Chizat et al., 2015, Liero et al., 2018, Chizat et al., 2016a] which provides a distance between unnormalized measures. This generalization is out of the scope of this work and may be an interesting line of research in the future.

4.1.3 Structure of the chapter

In Section 4.2, we provide some background on GPCA in the Wasserstein space $W_2(\Omega)$, borrowing material from previous work in [Bigot et al., 2015]. Section 4.3 describes log-PCA in $W_2(\Omega)$, and some of its limitations are discussed. Section 4.4 contains the main results of this chapter, namely two algorithms for computing GPCA. In Section 4.5, we provide a comparison between GPCA and log-PCA using statistical analysis of real data sets of histograms. Various details on the implementation of the algorithms are deferred to technical Appendix.

4.2 Background on Geodesic PCA in the Wasserstein space

4.2.1 Definitions and notations

Let Ω be a (possibly unbounded) interval in $\mathbb{R}$. Let ν be a probability measure (also called distribution) over $(\Omega, \mathcal{B}(\Omega))$ where $\mathcal{B}(\Omega)$ is the σ-algebra of Borel subsets of Ω. For a mapping $T : \Omega \to \Omega$, the pushforward measure, also referred to as image measure, $T\#\nu$ is a probability measure on Ω defined by $(T\#\nu)(A) = \nu\{x \in \Omega | T(x) \in A\}$, for any $A \in \mathcal{B}(\Omega)$. The cumulative distribution function (cdf) and the (generalized) quantile function of ν are denoted respectively by F_ν and F_ν^-. The Wasserstein space $W_2(\Omega)$ is the set of probability measures with support included in Ω and having a finite second moment, that is endowed with the quadratic Wasserstein distance d_W defined by

$$d_W^2(\mu, \nu) := \int_0^1 (F_\mu^-(\alpha) - F_\nu^-(\alpha))^2 d\alpha, \ \mu, \nu \in W_2(\Omega). \tag{4.2.1}$$

We also denote by $W_2^{ac}(\Omega)$ the set of measures $\nu \in W_2(\Omega)$ that are absolutely continuous with respect to the Lebesgue measure dx on $\mathbb{R}$. If $\mu \in W_2^{ac}(\Omega)$ then $T^* = F_\nu^- \circ F_\mu$ will be referred to as the optimal mapping to push forward μ onto ν and in this case $d_W^2(\mu, \nu) = \int_\Omega (T^*(x) - x)^2 d\mu(x)$. For a detailed analysis of $W_2(\Omega)$ and its connection with optimal transport theory, we refer to [Villani, 2003].

4.2.2 The pseudo Riemannian structure of the Wasserstein space

In what follows, μ_r denotes a reference measure in $W_2^{ac}(\Omega)$, whose choice will be discussed later on. The space $W_2(\Omega)$ has a formal Riemannian structure described, for example, in [Ambrosio et al., 2006]. The tangent space at μ_r is defined as the Hilbert space $L^2_{\mu_r}(\Omega)$ of real-valued, μ_r-square-integrable functions on Ω, equipped with the inner product $\langle \cdot, \cdot \rangle_{\mu_r}$ defined by $\langle u, v \rangle_{\mu_r} = \int_\Omega u(x)v(x)d\mu_r(x), \ u, v \in L^2_{\mu_r}(\Omega)$, and associated norm $\| \cdot \|_{\mu_r}$. We define the exponential and the logarithmic maps at μ_r, as follows.

Definition 4.2.1. Let $\mathrm{id} : \Omega \to \Omega$ be the identity mapping. The exponential $\exp_{\mu_r} : L^2_{\mu_r}(\Omega) \to W_2(\mathbb{R})$ and logarithmic $\log_{\mu_r} : W_2(\Omega) \to L^2_{\mu_r}(\Omega)$ maps are defined respectively as

$$\exp_{\mu_r}(v) = (\mathrm{id} + v)\#\mu_r \quad \text{and} \quad \log_{\mu_r}(\nu) = F_\nu^- \circ F_{\mu_r} - \mathrm{id}. \tag{4.2.2}$$

Contrary to the setting of Riemannian manifolds, the "exponential map" $\exp_{\mu_r}$ defined above is not a local homeomorphism from a neighborhood of the origin in

the "tangent space" $L^2_{\mu_r}(\Omega)$ to the space $W_2(\Omega)$, see e.g. [Ambrosio et al., 2006]. Nevertheless, it is shown in [Bigot et al., 2015] that $\exp_{\mu_r}$ is an isometry when restricted to the following specific set of functions

$$V_{\mu_r}(\Omega) := \log_{\mu_r}(W_2(\Omega)) = \left\{ \log_{\mu_r}(\nu) \; ; \; \nu \in W_2(\Omega) \right\} \subset L^2_{\mu_r}(\Omega),$$

and that the following results hold (see [Bigot et al., 2015]).

Proposition 4.2.1. *The subspace $V_{\mu_r}(\Omega)$ satisfies the following properties :*

(P1) *the exponential map $\exp_{\mu_r}$ restricted to $V_{\mu_r}(\Omega)$ is an isometric homeomorphism, with inverse $\log_{\mu_r}$. We have hence $d_W(\nu,\eta) = ||\log_{\mu_r}(\nu) - \log_{\mu_r}(\eta)||_{L^2_{\mu_r}(\Omega)}$.*

(P2) *the set $V_{\mu_r}(\Omega) := \log_{\mu}(W_2(\Omega))$ is closed and convex in $L^2_{\mu_r}(\Omega)$.*

(P3) *the space $V_{\mu_r}(\Omega)$ is the set of functions $v \in L^2_{\mu_r}(\Omega)$ such that $T := \mathrm{id} + v$ is μ_r-almost everywhere non decreasing and that $T(x) \in \Omega$, for $x \in \Omega$.*

Moreover, it follows from [Bigot et al., 2015] that geodesics in $W_2(\Omega)$ are exactly the image under $\exp_{\mu_r}$ of straight lines in $V_{\mu_r}(\Omega)$. This property is stated in the following lemma.

Lemma 4.2.1. *Let $\gamma : [0,1] \to W_2(\Omega)$ be a curve and let $v_0 := \log_{\mu_r}(\gamma(0))$, $v_1 := \log_{\mu_r}(\gamma(1))$. Then $\gamma = (\gamma_t)_{t\in[0,1]}$ is a geodesic if and only if $\gamma_t = \exp_{\mu_r}((1-t)v_0 + tv_1)$, for all $t \in [0,1]$.*

4.2.3 GPCA for probability measures

Let $\nu_1, \ldots, \nu_n$ be a set of probability measures in $W_2^{ac}(\Omega)$. Assuming that each ν_i is absolutely continuous simplifies the following presentation, and it is in line with the purpose of statistical analysis of histograms. We define now the notion of (empirical) GPCA of this set of probability measures by following the approach in [Bigot et al., 2015]. The first step is to choose the reference measure μ_r. To this end, let us introduce the Wasserstein barycenter [Agueh and Carlier, 2011] or Fréchet mean of the ν_i's, that is defined as the probability measure $\bar{\mu}$,

$$\bar{\mu} = \mathrm{argmin}_{\mu\in W_2(\Omega)} \frac{1}{n}\sum_{i=1}^{n} d^2_W(\nu_i,\mu).$$

Note that it immediately follows from results in [Agueh and Carlier, 2011] that $\bar{\mu} \in W_2^{ac}(\Omega)$, and that its cdf satisfies

$$F^-_{\bar{\mu}} = \frac{1}{n}\sum_{i=1}^{n} F^-_{\nu_i}. \tag{4.2.3}$$

A typical choice for the reference measure is to take $\mu_r = \bar{\mu}$ which represents an average location in the data around which can be computed the principal sources of geodesic variability. To introduce the notion of a principal geodesic subspace of the measures $\nu_1, \ldots, \nu_n$, we need to introduce further notation and definitions. Let G be a subset of $W_2(\Omega)$. The distance between $\mu \in W_2(\Omega)$ and the set G is $d_W(\nu, G) = \inf_{\lambda\in G} d_W(\nu,\lambda)$, and the average distance between the data and G is taken as

$$\mathrm{D}_W(G) := \frac{1}{n}\sum_{i=1}^{n} d^2_W(\nu_i, G). \tag{4.2.4}$$

Definition 4.2.2. Let K be some positive integer. A subset $G \subset W_2(\Omega)$ is said to be a geodesic set of dimension $\dim(G) = K$ if $\log_{\mu_r}(G)$ is a convex set such that the dimension of the smallest affine subspace of $L^2_{\mu_r}(\Omega)$ containing $\log_{\mu_r}(G)$ is of dimension K.

The notion of principal geodesic subspace (PGS) with respect to the reference measure $\mu_r = \bar{\mu}$ can now be presented below.

Definition 4.2.3. Let $\mathrm{CL}(W)$ be the metric space of nonempty, closed subsets of $W_2(\Omega)$, endowed with the Hausdorff distance, and

$$\mathrm{CG}_{\bar{\mu},K}(W) = \{G \in \mathrm{CL}(W) \mid \bar{\mu} \in G,\ G \text{ is a geodesic set and } \dim(G) \leq K\},\ K \geq 1.$$

A principal geodesic subspace (PGS) of $\mu = (\nu_1, \ldots, \nu_n)$ of dimension K with respect to $\bar{\mu}$ is a set

$$G_K \in \underset{G \in \mathrm{CG}_{\bar{\mu},K}(W)}{\operatorname{argmin}} \mathrm{D}_W(G) = \underset{G \in \mathrm{CG}_{\bar{\mu},K}(W)}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} d^2_W(\nu_i, G). \tag{4.2.5}$$

When $K = 1$, searching for the first PGS of μ simply amounts to search for a geodesic curve $\gamma^{(1)}$ that is a solution of the following optimization problem:

$$\tilde{\gamma}^{(1)} := \underset{\gamma}{\operatorname{argmin}} \left\{ \frac{1}{n}\sum_{i=1}^{n} d^2_W(\nu_i, \gamma) \mid \gamma \text{ is a geodesic in } W_2(\Omega) \text{ passing through } \mu_r = \bar{\mu}. \right\}.$$

We remark that this definition of $\tilde{\gamma}^{(1)}$ as the first principal geodesic curve of variation in $W_2(\Omega)$ is consistent with the usual concept of PCA in a Hilbert space in which geodesic are straight lines.

For a given dimension k, the GPCA problem consists in finding a nonempty closed geodesic subset of dimension k which contains the reference measure μ_r and minimizes Eq. (4.2.4). We describe in the next section how we can parameterize such sets G.

4.2.4 Geodesic PCA parameterization

GPCA can be formulated as an optimization problem in the Hilbert space $L^2_{\bar{\mu}}(\Omega)$. To this end, let us define the functions $\omega_i = \log_{\bar{\mu}}(\nu_i)$ for $1 \leq i \leq n$ that corresponds to the data mapped in the tangent space. It can be easily checked that this set of functions is centered in the sense that $\frac{1}{n}\sum_{i=1}^{n} \omega_i = 0$. Note that, in a one-dimensional setting, computing ω_i (mapping of the data to the tangent space) is straightforward since the optimal maps $T^*_i = F^-_{\nu_i} \circ F_{\bar{\mu}}$ between the data and their Fréchet mean are available in a simple and closed form.

For $\mathcal{U} = \{u_1, \ldots, u_K\}$ a collection of $K \geq 1$ functions belonging to $L^2_{\bar{\mu}}(\Omega)$, we denote by $\mathrm{Sp}(\mathcal{U})$ the subspace spanned by $u_1, \ldots, u_K$. Defining $\Pi_{\mathrm{Sp}(\mathcal{U})}v$ as the projection of $v \in L^2_{\bar{\mu}}(\Omega)$ onto $\mathrm{Sp}(\mathcal{U})$, and $\Pi_{\mathrm{Sp}(\mathcal{U}) \cap V_{\bar{\mu}}(\Omega)}v$ as the projection of v onto the closed convex set $\mathrm{Sp}(\mathcal{U}) \cap V_{\bar{\mu}}(\Omega)$, then we have

Proposition 4.2.2. *Let $\omega_i = \log_{\bar{\mu}}(\nu_i)$ for $1 \leq i \leq n$, and $\mathcal{U}^* = \{u^*_1, \ldots, u^*_k\}$ be a minimizer of*

$$\frac{1}{n}\sum_{i=1}^{n} \|\omega_i - \Pi_{\mathrm{Sp}(\mathcal{U}) \cap V_{\bar{\mu}}(\Omega)}\omega_i\|^2_{\bar{\mu}}, \tag{4.2.6}$$

over orthonormal sets $\mathcal{U} = \{u_1, \dots, u_K\}$ *of functions in* $L^2_{\bar{\mu}}(\Omega)$ *of dimension* K *(namely such that* $\langle u_j, u_{j'} \rangle_{\bar{\mu}} = 0$ *if* $j \neq j'$ *and* $\|u_j\|_{\bar{\mu}} = 1$*). If we let*

$$G_{\mathcal{U}^*} := \exp_{\bar{\mu}}(\mathrm{Sp}(\mathcal{U}^*) \cap V_{\bar{\mu}}(\Omega)),$$

then $G_{\mathcal{U}^*}$ *is a principal geodesic subset (PGS) of dimension* k *of the measures* $\nu_1, \dots, \nu_n$*, meaning that* $G_{\mathcal{U}^*}$ *belongs to the set of minimizers of the optimization problem* (4.2.5).

Proof. For $v \in L^2_{\bar{\mu}}(\Omega)$ and a subset $C \in L^2_{\bar{\mu}}(\Omega)$, we define $d_{\bar{v}}(v, C) = \inf_{u \in C} \|v - u\|_{\bar{v}}$. Remark that $\sum_i \omega_i = 0$. Hence by Proposition 3.3 in [Bigot et al., 2015], if $\mathcal{U}^*$ minimizes

$$\frac{1}{n} \sum_{i=1}^n d^2_{\bar{v}}(\omega_i, \mathrm{Sp}(\mathcal{U}^*) \cap V_{\bar{v}}(\Omega)) = \frac{1}{n} \sum_{i=1}^n \|\omega_i - \Pi_{\mathrm{Sp}(\mathcal{U}^*) \cap V_{\bar{v}}(\Omega)} \omega_i\|^2_{\bar{\mu}},$$

then $\mathrm{Sp}(\mathcal{U}^*) \cap V_{\bar{v}}(\Omega) \in \mathrm{argmin}_C \frac{1}{n} \sum_{i=1}^n d^2_{\bar{v}}(\omega_i, C)$, where C is taken over all nonempty, closed, convex set of $V_{\bar{v}}(\Omega)$ such that $dim(C) \leq K$ and $0 \in C$. By Proposition 4.3 in [Bigot et al., 2015], and since $\log_{\bar{v}}(\bar{v}) = 0$, we can conclude that G^* is a geodesic subset of dimension K which minimizes (4.2.4). □

Thanks to Proposition 4.2.2, it follows that GPCA in $W_2(\Omega)$ corresponds to a mapping of the data into the Hilbert space $L^2_{\bar{\mu}}(\Omega)$ which is followed by a PCA in $L^2_{\bar{\mu}}(\Omega)$ that is constrained to lie in the convex and closed subset $V_{\bar{\mu}}(\Omega)$. This has to be interpreted as a geodesicity constraint coming from the definition of a PGS in $W_2(\Omega)$. Because this geodesicity constraint is nontrivial to implement, recent works about geodesic PCA in $W_2(\Omega)$ relied on a heuristic projection on the set of optimal maps [Seguy and Cuturi, 2015], or relaxed the geodesicity constraint by solving a linearized PGA [Wang et al., 2013, Bigot et al., 2015]. We describe the latter approach in the following section.

4.3 The log-PCA approach

For data in a Riemannian manifold, we recall that log-PCA consists in solving a linearized version of the PGA problem by mapping the whole data set to the tangent space at the Fréchet mean through the logarithmic map Fletcher et al. [2004]. This approach is computationally attractive since it boils down to computing a standard PCA. Wang et al. [2013] used this idea to define a linearized PGA in the Wasserstein space $W_2(\mathbb{R}^d)$, by defining the logarithmic map of a probability measure as the barycentric projection of an optimal transport plan with respect to a template measure. This approach has the two drawbacks (1) and (2) of log-PCA mentioned in Section 4.1.1. A third limitation inherent to the Wasserstein space is that when this template probability measure is discrete, the logarithmic map cannot be defined straightforwardly as there is no guarantee about the existence of an optimal map solution of the optimal transport problem. This is why the authors of Wang et al. [2013] had to compute the barycentric projection of each optimal transport plan, which is obtained by simply averaging the locations of the split mass defined by this plan. This averaging process is however lossy as distinct probability measures can have the same barycentric projection.

We consider as usual a subset $\Omega \subset \mathbb{R}$. In this setting, $W_2(\Omega)$ is a flat space as shown by the isometry property (P1) of Proposition 4.2.1. Moreover, if the Wasserstein barycenter $\bar{\mu}$ is assumed to be absolutely continuous, then Definition 4.2.1

shows that the logarithmic map at $\bar{\mu}$ is well defined everywhere. Under such an assumption, log-PCA in $W_2(\Omega)$ corresponds to the following steps:

1. compute the log-maps (see Definition 4.2.1) $\omega_i = \log_{\bar{\mu}}(\nu_i)$, $i = 1, \dots, n$,
2. perform the PCA of the projected data $\omega_1, \cdots, \omega_n$ in the Hilbert space $L^2_{\bar{\mu}}(\Omega)$ to obtain K orthogonal directions $\tilde{u}_1, \dots, \tilde{u}_K$ in $L^2_{\bar{\mu}}(\Omega)$ of principal variations,
3. recover a principal subspace of variation in $W_2(\Omega)$ with the exponential map $\exp_{\bar{\mu}}(\mathrm{Sp}(\tilde{\mathcal{U}}))$ of the principal eigenspace $\mathrm{Sp}(\tilde{\mathcal{U}})$ in $L^2_{\bar{\mu}}(\Omega)$ spanned by $\tilde{u}_1, \dots, \tilde{u}_K$.

For specific datasets, log-PCA in $W_2(\Omega)$ may be equivalent to GPCA, in the sense that the set $\exp_{\bar{\mu}}(\mathrm{Sp}(\tilde{\mathcal{U}}) \cap V_{\bar{\mu}}(\Omega))$ is a principal geodesic subset of dimension K of the measures $\nu_1, \dots, \nu_n$, as defined by (4.2.5). Informally, this case corresponds to the setting where the data are sufficiently concentrated around their Wasserstein barycenter $\bar{\mu}$ (we refer to Remark 3.5 in Bigot et al. [2015] for further details). However, carrying out a PCA in the tangent space of $W_2(\mathbb{R})$ at $\bar{\mu}$ is a relaxation of the convex-constrained GPCA problem (4.2.6), where the elements of the sought principal subspace do not need to be in $V_{\bar{\mu}}$. Indeed, standard PCA in the Hilbert space $L^2_{\bar{\mu}}(\Omega)$ amounts to finding $\tilde{\mathcal{U}} = \{\tilde{u}_1, \dots, \tilde{u}_K\}$ minimizing,

$$\frac{1}{n} \sum_{i=1}^{n} \|\omega_i - \Pi_{\mathrm{Sp}(\mathcal{U})} \omega_i\|^2_{\bar{\mu}}, , \tag{4.3.1}$$

over orthonormal sets $\mathcal{U} = \{u_1, \dots, u_k\}$ of functions in $L^2_{\bar{\mu}}(\Omega)$. It is worth noting that the three steps of log-PCA in $W_2(\Omega)$ are simple to implement and fast to compute, but that performing log-PCA or GPCA (4.2.6) in $W_2(\Omega)$ are not necessarily equivalent.

Log-PCA is generally used for two main purposes. The first one is to obtain a low dimensional representation of each data measure $\nu_i = \exp_{\bar{\mu}}(\omega_i)$ through the coefficients $\langle \omega_i, \tilde{u}_k \rangle_{L^2_{\bar{\mu}}}$. From this low dimensional representation, the measure $\nu_i \in W_2(\Omega)$ can be approximated through the exponential mapping $\exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(\mathcal{U})} \omega_i)$. The second one is to visualize each mode of variation in the dataset, by considering the evolution of the curve $t \mapsto \exp_{\bar{\mu}}(t\tilde{u}_k)$ for each $\tilde{u}_k \in \tilde{\mathcal{U}}$.

However, relaxing the convex-constrained GPCA problem (4.2.6) when using log-PCA results in several issues. Indeed, as shown in the following paragraphs, not taking into account this geodesicity constraint makes difficult the computation and interpretation of $\exp_{\bar{\mu}}(\mathrm{Sp}(\tilde{\mathcal{U}}))$ as a principal subspace of variation, which may limit its use for data analysis.

Numerical implementation of pushforward operators A first downside to the log-PCA approach is the difficulty of the numerical implementation of the pushforward operator in the exponential map $\exp_{\bar{\mu}}(v) = (\mathrm{id} + v)\#\bar{\mu}$ when the mapping $\mathrm{id} + v$ is not a strictly increasing function for a given vector $v \in \mathrm{Sp}(\tilde{\mathcal{U}})$. This can be shown with the following proposition, which provides a formula for computing the density of a pushforward operator.

Proposition 4.3.1. *(Density of the pushforward) Let $\mu \in W_2(\mathbb{R})$ be an absolutely continuous measure with density ρ (that is possibly supported on an interval $\Omega \subset \mathbb{R}$). Let $T : \mathbb{R} \to \mathbb{R}$ be a differentiable function such that $|T'(x)| > 0$ for almost every $x \in \mathbb{R}$, and*

define $\nu = T\#\mu$. Then, ν admits a density g given by,

$$g(y) = \sum_{x \in T^{-1}(y)} \frac{\rho(x)}{|T'(x)|}, \ y \in \mathbb{R}. \tag{4.3.2}$$

When T is injective, this simplifies to,

$$g(y) = \frac{\rho(T^{-1}(y))}{|T'(T^{-1}(y))|}. \tag{4.3.3}$$

Proof. Under the assumptions made on T, the coarea formula (which is a more general form of Fubini's theorem, see e.g. Krantz and Parks [2008] Corollary 5.2.6 or Evans and Gariepy [2015] Section 3.4.3) states that, for any measurable function $h : \mathbb{R} \to \mathbb{R}$, one has

$$\int_{\mathbb{R}} h(x)|T'(x)|dx = \int_{\mathbb{R}} \sum_{x \in T^{-1}(y)} h(x)dy. \tag{4.3.4}$$

Let B a Borel set and choose $h(x) = \frac{\rho(x)}{|T'|}\mathbf{1}_{T^{-1}(B)}, \ x$. Hence, using (4.3.4), one obtains that

$$\int_{T^{-1}(B)} \rho(x)dx = \int_{\mathbb{R}} \sum_{x \in T^{-1}(y)} \frac{\rho(x)}{|T'(x)|}\mathbf{1}_{T^{-1}(B)}(x)dy = \int_{B} \sum_{x \in T^{-1}(y)} \frac{\rho(x)}{|T'(x)|}dy.$$

The definition of the pushforward $\nu(B) = \mu(T^{-1}(B))$ then completes the proof. □

The numerical computation of formula (4.3.2) or (4.3.3) is not straightforward. When T is not injective, computation of the formula (4.3.2) must be done carefully by partitioning the domain of T in sets on which T is injective. Such a partitioning depends on the method of interpolation for estimating a continuous density ρ from a finite set of its values on a grid of reals. More importantly, when $T'(x)$ is very small, $\frac{\rho(x)}{|T'(x)|}$ may become very irregular and the density of $\nu = T\#\mu$ may exhibit large peaks, see Figure 4.2 for an illustrative example.

Pushforward of the barycenter outside the support Ω A second downside of log-PCA in $W_2(\Omega)$ is that the range of the mapping $\tilde{T}_i = \mathrm{id} + \Pi_{\mathrm{Sp}(\tilde{\mathcal{U}})}\omega_i$ may be larger than the interval Ω. This implies that the density of the pushforward of the Wasserstein barycenter $\bar{\mu}$ by this mapping, namely $\exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(\tilde{\mathcal{U}})}\omega_i)$, may have a support which is not included in Ω. This issue may be critical when trying to estimate the measure $\nu_i = \exp_{\bar{\mu}}(\omega_i)$ by its projected measure $\exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(\tilde{\mathcal{U}})}\omega_i)$. For example, in a dataset of histograms with bins necessarily containing only nonnegative reals, a projected distribution with positive mass on negative reals would be hard to interpret.

A higher Wasserstein reconstruction error Finally, relaxing the geodesicity constraint (4.2.6) may actually increase the Wasserstein reconstruction error with respect to the Wasserstein distance. To state this issue more clearly, we define the reconstruction error of log-PCA as

$$\tilde{r}(\tilde{\mathcal{U}}) = \frac{1}{n}\sum_{i=1}^{n} d_W^2\left(\nu_i, \exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(\tilde{\mathcal{U}})}\omega_i)\right). \tag{4.3.5}$$

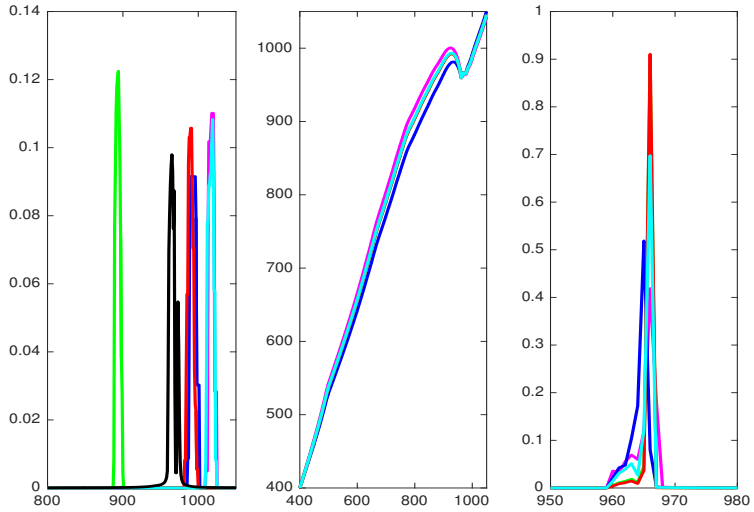

FIGURE 4.2: (Left) Distribution of the total precipitation (mm) collected in a year in $1 \leq i \leq 5$ stations among 60 in China - Source : Climate Data Bases of the People's Republic of China 1841-1988 downloaded from http://cdiac.ornl.gov/ndps/tr055.html. The black curve is the density of the Wasserstein barycenter of the 60 stations. (Middle) Mapping $T_i = \mathrm{id} + \Pi_{\mathrm{Sp}(\tilde{u}_2)}\omega_i$ obtained from the projections of these 5 distributions onto the second eigenvector $\tilde{u}_2$ given by log-PCA of the whole dataset. (Right) Pushforward $\exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(\tilde{u}_2)}\omega_i) = T_i\#\bar{\mu}$ of the Wasserstein barycenter $\bar{\mu}$ for each $1 \leq i \leq 5$. As the derivative T_i' take very small values, the densities of the pushforward barycenter $T_i\#\bar{\mu}$ for $1 \leq i \leq 5$ exhibit large peaks (between 0.4 and 0.9) whose amplitude is beyond the largest values in the original data set (between 0.08 and 0.12).

and the reconstruction error of GPCA as

$$r(\mathcal{U}^*) = \frac{1}{n}\sum_{i=1}^{n} d_W^2\big(\nu_i, \exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(\mathcal{U}^*)\cap V_{\bar{\mu}}(\Omega)}\omega_i)\big). \tag{4.3.6}$$

where $\mathcal{U}^*$ is a minimizer of (4.2.6). Note that in (4.3.5), the projected measures $\exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(\tilde{\mathcal{U}})}\omega_i)$ might have a support that lie outside Ω. Hence, the Wasserstein distance d_W in (4.3.5) has to be understood for measures supported on $\mathbb{R}$ (with the obvious extension to zero of ν_i outside Ω).

The Wasserstein reconstruction error $\tilde{r}(\tilde{\mathcal{U}})$ of log-PCA is the sum of the Wasserstein distances of each data point ν_i to a point on the surface $\exp_{\bar{\mu}}(\mathrm{Sp}(\tilde{\mathcal{U}}))$ which is given by the decomposition of ω_i on the orthonormal basis $\tilde{\mathcal{U}}$. However, by Proposition 4.2.1, the isometry property (P1) only holds between $W_2(\mathbb{R})$ and the convex subset $V_{\bar{\mu}} \subset L^2_{\bar{\mu}}(\mathbb{R})$. Therefore, we may not have $d_W^2\big(\nu_i, \exp_{\bar{\mu}}\big(\Pi_{\mathrm{Sp}(\tilde{\mathcal{U}})}\omega_i\big)\big) = \|\omega_i - \Pi_{\mathrm{Sp}(\tilde{\mathcal{U}})}\omega_i\|^2_{\bar{\mu}}$ as $\Pi_{\mathrm{Sp}(\tilde{\mathcal{U}})}\omega_i$ is a function belonging to $L^2_{\bar{\mu}}(\mathbb{R})$ which may not necessarily be in $V_{\bar{\mu}}$. In this case, the minimal Wasserstein distance between ν_i and the surface $\exp_{\bar{\mu}}(\mathrm{Sp}(\mathcal{U}^*))$ is not equal to $\|\omega_i - \Pi_{\mathrm{Sp}(\mathcal{U})}\omega_i\|_{\bar{\mu}}$, and this leads to situations where $\tilde{r}(\tilde{\mathcal{U}}) > r(\mathcal{U}^*)$ as illustrated in Figure 4.3.

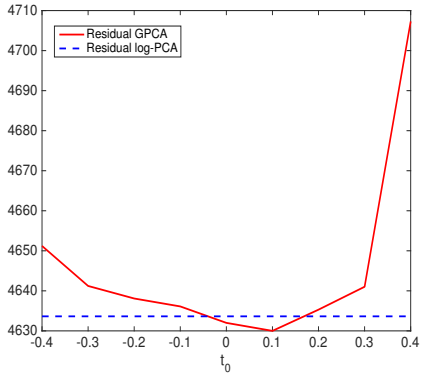

FIGURE 4.3: Comparison of the Wasserstein reconstruction error between GPCA and log-PCA on the synthetic dataset displayed in Figure 4.1 for the first component, with an illustration of the role of the parameter t_0 in (4.4.2).

4.4 Two algorithmic approaches for GPCA in $W_2(\Omega)$, for $\Omega \subset \mathbb{R}$

In this section, we introduce two algorithms which solve some of the issues of log-PCA that have been raised in Section 4.3. First, the output of the proposed algorithms guarantees that the computation of mappings to pushforward the Wasserstein barycenter to approximate elements in the data set are strictly increasing (that is they are optimal). As a consequence, the resulting pushforward density behaves numerically much better. Secondly, the geodesic curve or surface are constrained to lie in $W_2(\Omega)$, implying that the projections of the data are distributions whose supports do not lie outside Ω.

4.4.1 Iterative geodesic approach

In this section, we propose an algorithm to solve a variant of the convex-constrained GPCA problem (4.2.6). Rather than looking for a geodesic subset of a given dimension which fits well the data, we find iteratively orthogonal principal geodesics (i.e. geodesic set of dimension one). Assuming that that we already know a subset $\mathcal{U}^{k-1} \subset L^2_{\bar{\mu}}(\Omega)$ containing $k-1$ orthogonal principal directions $\{u_l\}_{l=1}^{k-1}$ (with $\mathcal{U}^0 = \emptyset$), our goal is to find a new direction $u_k \in L^2_{\bar{\mu}}(\Omega)$ of principal variation by solving the optimization problem:

$$u_k \in \underset{v \perp \mathcal{U}^{k-1}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \|\omega_i - \Pi_{\mathrm{Sp}(v) \cap V_{\bar{\mu}}(\Omega)} \omega_i\|^2_{\bar{\mu}}, \tag{4.4.1}$$

where the infimum above is taken over all $v \in L^2_{\bar{\mu}}(\Omega)$ belonging to the orthogonal of $\mathcal{U}^{k-1}$. This iterative process is not equivalent to the GPCA problem (4.2.6), with the exception of the first principal geodesic ($k = 1$). Nevertheless, it computes principal subsets $\mathcal{U}^k$ of dimension k such that the projections of the data onto every direction of principal variation lie in the convex set $V_{\bar{\mu}}$.

The following proposition is the key result to derive an algorithm to solve (4.4.1) on real data.

Proposition 4.4.1. *Introducing the characteristic function of the convex set $V_{\bar{\mu}}(\Omega)$ as:*

$$\chi_{V_{\bar{\mu}}(\Omega)}(v) = \begin{cases} 0 & if\ v \in V_{\bar{\mu}}(\Omega) \\ +\infty & otherwise \end{cases}$$

the optimization problem (4.4.1) is equivalent to

$$u_k = \underset{v \perp \mathcal{U}^{k-1}}{\operatorname{argmin}} \min_{t_0 \in [-1;1]} H(t_0, v), \tag{4.4.2}$$

where

$$H(t_0, v) = \frac{1}{n} \sum_{i=1}^{n} \min_{t_i \in [-1;1]} \|\omega_i - (t_0 + t_i)v\|^2_{\bar{\mu}} + \chi_{V_{\bar{\mu}}(\Omega)}((t_0 - 1)v) + \chi_{V_{\bar{\mu}}(\Omega)}((t_0 + 1)v). \tag{4.4.3}$$

Proof. We first observe that $\Pi_{\mathrm{Sp}(u) \cap V_{\bar{\mu}}(\Omega)} \omega_i = \beta_i u$, with $\beta_i \in \mathbb{R}$ and $\beta_i u \in V_{\bar{\mu}}(\Omega)$. Hence, for u_k solution of (4.4.1), we have:

$$\frac{1}{n} \sum_{i=1}^{n} \|\omega_i - \Pi_{\mathrm{Sp}(u_k) \cap V_{\bar{\mu}}(\Omega)} \omega_i\|^2_{\bar{\mu}} = \frac{1}{n} \sum_{i=1}^{n} \|\omega_i - \beta_i u_k\|^2_{\bar{\mu}}.$$

such that $\beta_i \in \mathbb{R}$ and $\beta_i u_k \in V_{\bar{\mu}}(\Omega)$ for all $i \in \{1, \ldots, n\}$. We take $M \in \operatorname{argmax}_{1 \leq i \leq n} \beta_i$ and $m \in \operatorname{argmin}_{1 \leq i \leq n} \beta_i$. Without loss of generality, we can assume that $\beta_M > 0$ and $\beta_m < 0$. We then define $v = (\beta_M - \beta_m)u_k/2$ and $t_0 = (\beta_M + \beta_m)/(\beta_M - \beta_m)$, that checks $|t_0| < 1$. Hence, for all $i = 1, \ldots, n$, there exists $t_i \in [-1; 1]$ such that: $\beta_i u_k = (t_0 + t_i)v \in V_{\bar{\mu}}$. In particular, one has $t_M = 1$ and $t_m = -1$, which means that $(t_0 \pm 1)v \in V_{\bar{\mu}}(\Omega)$. Reciprocally, $(t_0 \pm 1)v \in V_{\bar{\mu}}(\Omega)$ ensures us by convexity of $V_{\bar{\mu}}(\Omega)$ that for all $t_i \in [-1; 1]$, $(t_0 + t_i)v \in V_{\bar{\mu}}(\Omega)$. □

Proposition 4.4.1 may be interpreted as follows. For a given $t_0 \in [-1; 1]$, let $v \in \perp\ \mathcal{U}^{k-1}$ satisfying $(t_0 - 1)v \in V_{\bar{\mu}}$ and $(t_0 + 1)v \in V_{\bar{\mu}}$. Then, if one defines the curve

$$g_t(t_0, v) = (\mathrm{id} + (t_0 + t)v)\#\bar{\mu} \text{ for } t \in [-1; 1], \tag{4.4.4}$$

it follows, from Lemma 4.2.1, that $(g_t(t_0, v))_{t \in [-1;1]}$ is a geodesic since it can be written as $g_t(t_0, v) = \exp_{\bar{\mu}}((1 - u)w_0 + uw_1), u \in [0, 1]$ with $w_0 = (t_0 - 1)v$, $w_1 = (t_0 + 1)v$, $u = (t + 1)/2$, and with w_0 and w_1 belonging to $V_{\bar{\mu}}$ for $|t_0| < 1$. From the isometry property (P1) in Proposition 4.2.1, one has

$$\min_{t_i \in [-1;1]} \|\omega_i - (t_0 + t_i)v\|^2_{\bar{\mu}} = \min_{t_i \in [-1;1]} d^2_W(\nu_i, g_{t_i}(v)), \tag{4.4.5}$$

and thus the objective function $H(t_0, v)$ in (4.4.2) is equal to the sum of the squared Wasserstein distances between the data set and the geodesic curve $(g_t(t_0, v))_{t \in [-1;1]}$.

The choice of the parameter t_0 corresponds to the location of the mid-point of the geodesic $g_t(t_0, v)$, and it plays a crucial role. Indeed, the minimization of $H(t_0, v)$ over $t_0 \in [-1; 1]$ in (4.4.2) cannot be avoided to obtain an optimal Wasserstein reconstruction error. This is illustrated by the Figure 4.3, where the Wasserstein reconstruction error $\tilde{r}(\tilde{\mathcal{U}})$ of log-PCA (see equation (4.3.5)) is compared with the ones of GPCA, for different t_0, obtained for $k = 1$ as

$$t_0 \in [-1; 1] \mapsto H(t_0, u_1^{t_0})$$

with $u_1^{t_0} = \text{argmin}_v H(t_0, v)$. This shows that GPCA can lead to a better low dimensional data representation than log-PCA in term of Wasserstein residual errors.

4.4.2 Geodesic surface approach

Once a family of vectors $(v_1, \cdots, v_k)$ has been found through the minimization of problem (4.4.1), one can recover a geodesic subset of dimension k by considering all convex combinations of the vectors $((t_0^1+1)v_1, (t_0^1-1)v_1, \cdots, (t_0^k+1)v_k, (t_0^k-1)v_k)$. However, this subset may not be a solution of (4.2.6) since we have no guarantee that a data point ν_i is actually close to this geodesic subset. This discussion suggests that we may consider solving the GPCA problem (4.2.6) over geodesic set parameterized as in Proposition 4.4.1. In order to find principal geodesic subsets which are close to the data set, we consider a family $V^K = (v_1, \cdots, v_K)$ of linearly independant vectors and $\mathbf{t}_0^K = (t_0^1, \cdots, t_0^K) \in [-1,1]^K$ such that $(t_0^1-1)v_1, (t_0^1+1)v_1, \cdots, (t_0^K-1)v_K, (t_0^K+1)v_K$ are all in $V_{\bar{\mu}}$. Convex combinations of the latter family provide a parameterization of a geodesic set of dimension K by taking the exponential map $\exp_{\bar{\mu}}$ of

$$\hat{V}_{\bar{\mu}}(V^K, \mathbf{t}_0^K) = \{\sum_{k=1}^K (\alpha_k^+(t_0^k+1) + \alpha_k^-(t_0^k-1))v_k, \alpha^\pm \in A\} \tag{4.4.6}$$

where A is a simplex constraint: $\alpha^\pm \in A \Leftrightarrow \alpha_k^+, \alpha_k^- \geq 0$ and $\sum_{k=1}^K (\alpha_k^+ + \alpha_k^-) \leq 1$. We hence substitute the general sets $\text{Sp}(\mathcal{U}) \cap V_{\bar{\mu}}(\Omega)$ in the definition of the GPCA problem (4.2.6) to obtain,

$$\begin{aligned}
(u_1, \cdots, u_K) &= \underset{V^K, \mathbf{t}_0^K}{\text{argmin}} \quad \frac{1}{n}\sum_{i=1}^n \|\omega_i - \Pi_{\hat{V}_{\bar{\mu}}(V^K, t^K)}\omega_i\|_{\bar{\mu}}^2, \\
&= \underset{v_1, \cdots, v_K}{\text{argmin}} \quad \min_{\mathbf{t}_0^K \in [-1,1]^K} \frac{1}{n}\sum_{i=1}^n \min_{\alpha_i^\pm \in A} \|\omega_i - \sum_{k=1}^K (\alpha_{ik}^+(t_0^k+1) + \alpha_{ik}^-(t_0^k-1))v_k\|_{\bar{\mu}}^2 \\
&\qquad + \sum_{k=1}^K \left(\chi_{V_{\bar{\mu}}(\Omega)}((t_0^k+1)v_k) + \chi_{V_{\bar{\mu}}(\Omega)}((t_0^k-1)v_k)\right) + \sum_{i=1}^n \chi_A(\alpha_i^\pm).
\end{aligned} \tag{4.4.7}$$

4.4.3 Discretization and optimization

In this section we follow the framework of the iterative geodesic algorithm. We provide additional details when the optimization procedure of the geodesic surface approach differs from the iterative one.

Discrete optimization problem

Let $\Omega = [a; b]$ be a compact interval, and consider its discretization over N points $a = x_1 < x_2 < \cdots < x_N = b$, $\Delta_j = x_{j+1} - x_j$, $j = 1, \ldots, N-1$. We recall that the functions $\omega_i = \log_{\bar{\mu}}(\nu_i)$ for $1 \leq i \leq n$ are elements of $L^2_{\bar{\mu}}(\Omega)$ which correspond to the mapping of the data to the tangent space at the Wasserstein barycenter $\bar{\mu}$. In what follows, for each $1 \leq i \leq n$, the discretization of the function ω_i over the grid reads $\mathbf{w}_i = (w_i^j)_{j=1}^N \in \mathbb{R}^N$. We also recall that $\chi_A(u)$ is the characteristic function of a given set A, namely $\chi_A(u) = 0$ if $u \in A$ and $+\infty$ otherwise. Finally, the space $\mathbb{R}^N$ is understood to be endowed with the following inner product and norm $\langle \mathbf{u}, \mathbf{v} \rangle_{\bar{\mu}} = \sum_{j=1}^N \bar{f}(x_j) u_j v_j$ and $\|\mathbf{v}\|_{\bar{\mu}}^2 = \langle \mathbf{v}, \mathbf{v} \rangle_{\bar{\mu}}$ for $\mathbf{u}, \mathbf{v} \in \mathbb{R}^N$, where $\bar{f}$ denotes the density of the measure $\bar{\mu}$. Let us now suppose that we have already computed $k-1$ orthogonal (in

the sense $\langle \mathbf{u}, \mathbf{v} \rangle_{\bar{\mu}} = 0$) vectors $\mathbf{u}_1, \cdots \mathbf{u}_{k-1}$ in $\mathbb{R}^N$ which stand for the discretization of orthonormal functions $u_1, \ldots, u_{k-1}$ in $L^2_{\bar{\mu}}(\Omega)$ over the grid $(x_j)_{j=1}^N$.

Discretizing problem (4.4.2) for a fixed $t_0 \in]-1;1[$, our goal is to find a new direction $\mathbf{u}_k \in \mathbb{R}^N$ of principal variations by solving the following problem over all $\mathbf{v} = \{v_j\}_{j=1}^N \in \mathbb{R}^N$:

$$\mathbf{u}_k \in \operatorname*{argmin}_{\mathbf{v}\in\mathbb{R}^N} \frac{1}{n}\sum_{i=1}^{n}\left(\min_{t_i\in[-1;1]} \|\mathbf{w}_i - (t_0+t_i)\mathbf{v}\|^2_{\bar{\mu}}\right) + \chi_S(\mathbf{v}) + \chi_V((t_0-1)\mathbf{v}) + \chi_V((t_0+1)\mathbf{v}), \tag{4.4.8}$$

where $S = \{\mathbf{v} \in \mathbb{R}^N \text{ s.t. } \langle \mathbf{v}, \mathbf{u}_l \rangle_{\bar{\mu}} = 0,\ l = 1 \cdots k-1\}$ is a convex set that deals with the orthogonality constraint $\mathbf{v} \perp \mathcal{U}^{k-1}$ and V corresponds to the discretization of the constraints contained in $V_{\bar{\mu}}(\Omega)$. From Proposition 4.2.1 (P3), we have that $\forall v \in V_{\bar{\mu}}(\Omega)$, $T := \mathrm{id} + v$ is non decreasing and $T(x) \in \Omega$ for all $x \in \Omega$. Hence the discrete convex set V is defined as

$$V = \{\mathbf{v} \in \mathbb{R}^N \text{ s.t. } x_{j+1} + v_{j+1} \geq x_j + v_j,\ j = 1 \cdots N-1 \text{ and } x_j + v_j \in [a;b],\ j = 1 \cdots N\}$$

and can be rewritten as the intersection of two convex sets dealing with each constraint separately.

Proposition 4.4.2. *One has*

$$\chi_V((t_0-1)\mathbf{v}) + \chi_V((t_0+1)\mathbf{v}) = \chi_D(\mathbf{v}) + \chi_E(K\mathbf{v}),$$

where the convex sets D and E respectively deal with the domain constraints $x_j + (t_0+1)v_j \in [a;b]$ and $x_j + (t_0-1)v_j \in [a;b]$, i.e.:

$$D = \{\mathbf{v} \in \mathbb{R}^N, \text{ s.t. } m_j \leq v_j \leq M_j\}, \tag{4.4.9}$$

with $m_j = \max\left(\frac{a-x_j}{t_0+1}, \frac{b-x_j}{t_0-1}\right)$ and $M_j = \min\left(\frac{a-x_j}{t_0-1}, \frac{b-x_j}{t_0+1}\right)$, and the non decreasing constraint of $\mathrm{id} + (t_0 \pm 1)\mathbf{v}$:

$$E = \{\mathbf{z} \in \mathbb{R}^N \text{ s.t. } -1/(t_0+1) \leq z_j \leq 1/(1-t_0)\}. \tag{4.4.10}$$

with the differential operator $K : \mathbb{R}^N \to \mathbb{R}^N$ computing the discrete derivative of $\mathbf{v} \in \mathbb{R}^n$ as

$$(K\mathbf{v})_j = \begin{cases} (v_{j+1} - v_j)/(x_{j+1} - x_j) & \text{if } 1 \leq j < N \\ 0 & \text{if } j = N, \end{cases} \tag{4.4.11}$$

Having D and E both depending on t_0 is not an issue since problem (4.4.8) is solved for fixed t_0.

Introducing $\mathbf{t} = \{t_i\}_{i=1}^n \in \mathbb{R}^n$, problem (4.4.8) can be reformulated as:

$$\min_{\mathbf{v}\in\mathbb{R}^N} \min_{\mathbf{t}\in\mathbb{R}^n} J(\mathbf{v},\mathbf{t}) := \underbrace{\sum_{i=1}^{n} \|\mathbf{w}_i - (t_0+t_i)\mathbf{v}\|^2_{\bar{\mu}}}_{F(\mathbf{v},\mathbf{t})} + \underbrace{\chi_S(\mathbf{v}) + \chi_D(\mathbf{v}) + \chi_E(K\mathbf{v}) + \chi_{B_1^n}(\mathbf{t})}_{G(\mathbf{v},\mathbf{t})}. \tag{4.4.12}$$

where B_1^n is the L^∞ ball of $\mathbb{R}^n$ with radius 1 dealing with the constraint $t_i \in [-1;1]$. Notice that F is differentiable but non-convex in $(\mathbf{v},\mathbf{t})$ and G is non-smooth and convex.

Geodesic surface approach For fixed $(t_0^1, \dots, t_0^K) \in \mathbb{R}^K$ and $\mathbf{ff}^{\pm} = \{\alpha_k^+, \alpha_k^-\}_{k=1}^K$, the discretized version of (4.4.7) is then

$$\min_{\mathbf{v_1},\dots,\mathbf{v_K}\in\mathbb{R}^N} \min_{\mathbf{ff_1^{\pm}},\dots,\mathbf{ff_n^{\pm}}\in\mathbb{R}^{2K}} F'(\mathbf{v},\mathbf{t}) + G'(\mathbf{v},\mathbf{t}), \tag{4.4.13}$$

where $F'(\mathbf{v},\mathbf{t}) = \sum_{i=1}^n \|\mathbf{w}_i - \sum_{k=1}^K (\alpha_{ik}^+(t_0^k+1) + \alpha_{ik}^-(t_0^k-1))\mathbf{v_k}\|_{\tilde{\mu}}^2$ is still non-convex and differentiable, $G'(\mathbf{v},\mathbf{t}) = \sum_{k=1}^K \left(\chi_E(K\mathbf{v_k}) + \chi_{D_k}(\mathbf{v_k})\right) + \sum_{i=1}^n \chi_A(\mathbf{ff_i^{\pm}})^2$ is convex and non smooth, A is the simplex of $\mathbb{R}^{2K}$ and D_k is defined as in (4.4.9), depending on t_0^k. We recall that the orthogonality between vectors $\mathbf{v_k}$ is not taken into account in the geodesic surface approach.

Optimization through the forward-backward algorithm

Following [Attouch et al., 2013], in order to compute a critical point of problem (4.4.12), one can consider the forward-backward algorithm (see also [Ochs et al., 2014] for an acceleration using inertial terms). Denoting as $X = (\mathbf{v},\mathbf{t}) \in \mathbb{R}^{N+n}$, taking $\tau > 0$ and $X^{(0)} \in \mathbb{R}^{N+n}$, it reads:

$$X^{(\ell+1)} = \mathrm{Prox}_{\tau G}(X^{(\ell)} - \tau \nabla F(X^{(\ell)})), \tag{4.4.14}$$

where $\mathrm{Prox}_{\tau G}(\tilde{X}) = \mathrm{argmin}_X \frac{1}{2\tau}||X-\tilde{X}||^2 + G(X)$ with the Euclidean norm $||\cdot||$. In order to guarantee the convergence of this algorithm, the gradient of F has to be Lipschitz continuous with parameter $M > 0$ and the time step should be taken as $\tau < 1/M$. The details of computation of ∇F and $\mathrm{Prox}_{\tau G}$ for the two algorithms are given in the appendix of this chapter.

4.5 Comparison between log-PCA and GPCA on synthetic and real data

4.5.1 Synthetic example - Iterative versus geodesic surface approaches

First, for the synthetic example displayed in Figure 4.1, we compare the two algorithms (iterative and geodesic surface approaches) described in Section 4.4. The results are reported in Figure 4.4 by comparing the projection of the data onto the first and second geodesics computed with each approach. We also display the projection of the data onto the two-dimensional surface generated by each method. It should be recalled that the principal surface for the iterative geodesic algorithm is not necessarily a geodesic surface but each $g_t(t_0^k, u_k)_{t\in[-1;1]}$ defined by (4.4.4) for $k = 1, 2$ is a geodesic curve for $\mathcal{U} = \{u_1, u_2\}$. For data generated from a location-scale family of Gaussian distributions, it appears that each algorithm provides a satisfactory reconstruction of the data set. The main divergence concerns the first and second principal geodesic. Indeed enforcing the orthogonality between components in the iterative approach enables to clearly separate the modes of variation in location and scaling, whereas searching directly a geodesic surface in the second algorithm implies a mixing of these two types of variation.

Note that the barycenter of Gaussian distributions $\mathcal{N}(m_i, \sigma_i^2)$ can be shown to be Gaussian with mean $\sum m_i$ and variance $(\sum \sigma_i)^2$.

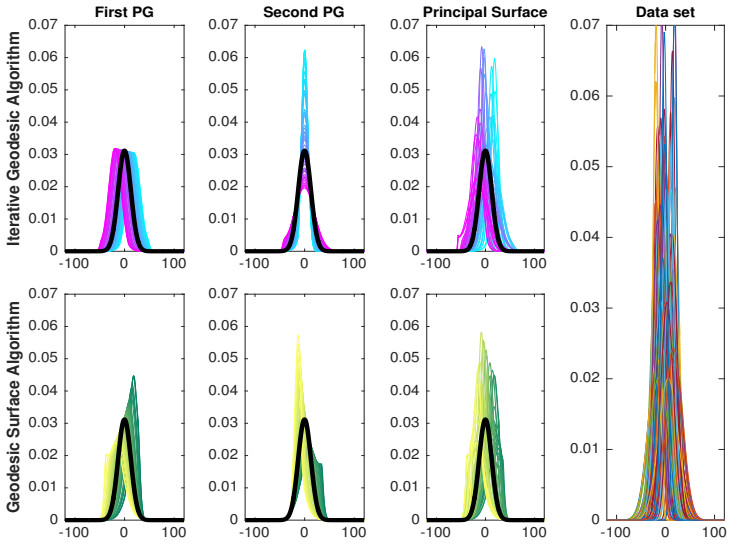

FIGURE 4.4: Synthetic example - Data sampled from a location-scale family of Gaussian distributions. The first row is the GPCA of the data set obtained with the iterative geodesic approach. The second row is the GPCA through the geodesic surface approach. The black curve is the density of the Wasserstein barycenter. Colors encode the progression of the pdf of principal geodesic components in $W_2(\Omega)$.

4.5.2 Population pyramids

As a first real example, we consider a real data set whose elements are histograms representing the population pyramids of $n = 217$ countries for the year 2000 (this data set is produced by the International Programs Center, US Census Bureau (IPC, 2000), available at https://www.census.gov/programs-surveys/international-programs.html). Each histogram in the database represents the relative frequency by age, of people living in a given country. Each bin in a histogram is an interval of one year, and the last interval corresponds to people older than 85 years. The histograms are normalized so that their area is equal to one, and thus they represent a set of pdf. In Figure 4.5, we display the population pyramids of 4 countries, and the whole data set. Along the interval $\Omega = [0, 84]$, the variability in this data set can be considered as being small.

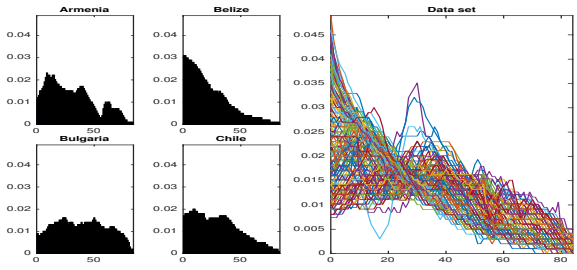

FIGURE 4.5: Population pyramids. A subset of population pyramids for 4 countries (left) for the year 2000, and the whole data set of $n = 217$ population pyramids (right) displayed as pdf over the interval $[0, 84]$.

For $K = 2$, log-PCA and the iterative GPCA algorithm lead to the same principal orthogonal directions in $L^2_{\tilde{\mu}}(\Omega)$, namely that $\tilde{u}_1 = u^*_1$ and $\tilde{u}_2 = u^*_2$ where $(\tilde{u}_1, \tilde{u}_2)$

minimizes (4.3.1) and (u_1^*, u_2^*) are minimizers of (4.4.2). In this case, all projections of data $\omega_i = \log_{\bar{\mu}}(\nu_i)$ for $i = 1, \dots, n$ onto $\mathrm{Sp}(\{\tilde{u}_1, \tilde{u}_2\})$ lie in $V_{\bar{\mu}}(\Omega)$, which means that log-PCA and the iterative geodesic algorithm lead exactly the same principal geodesics. Therefore, population pyramids is an example of data that are sufficiently concentrated around their Wasserstein barycenter so that log-PCA and GPCA are equivalent approaches (see Remark 3.5 in [Bigot et al., 2015] for further details). Hence, we only display in Figure 4.6 the results of the iterative and geodesic surface algorithms.

In the iterative case, the projection onto the first geodesic exhibits the difference between less developed countries (where the population is mostly young) and more developed countries (with an older population structure). The second geodesic captures more subtle divergences concentrated on the middle age population. It can be observed that the geodesic surface algorithm gives different results since the orthogonality constraint on the two principal geodesics is not required. In particular, the principal surface mainly exhibit differences between countries with a young population with countries having an older population structure, but the difference between its first and second principal geodesic is less contrasted.

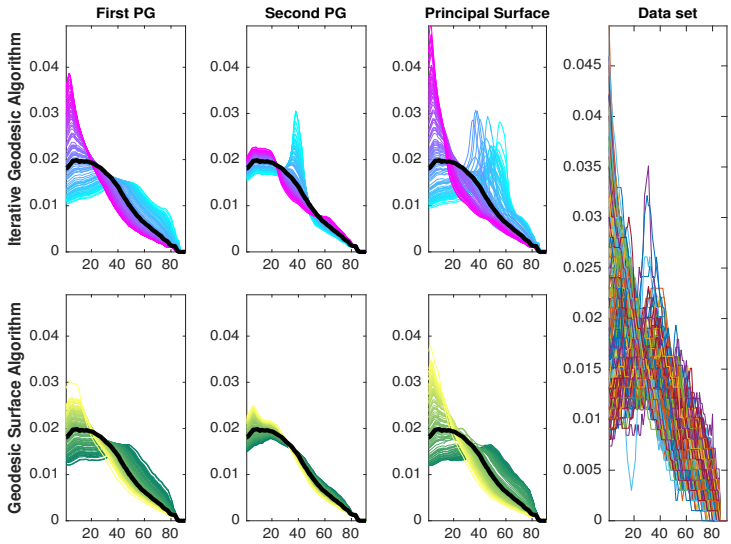

FIGURE 4.6: Population pyramids. The first row is the GPCA of the data set obtained with the iterative geodesic approach. The second row is the GPCA through the geodesic surface approach. The first (resp. second) column is the projection of the data into the first (resp. second) principal direction. The black curve is the density of the Wasserstein barycenter. Colors encode the progression of the pdf of principal geodesic components in $W_2(\Omega)$.

4.5.3 Children's first name at birth

In a second example, we consider a data set of histograms which represent, for a list of $n = 1060$ first names, the distribution of children born with that name per year in France between years 1900 and 2013. In Figure 4.7, we display the histograms of four different names, as well as the whole data set. Along the interval $\Omega = [1900, 2013]$, the variability in this data set is much larger than the one observed for population pyramids. This data set has been provided by the INSEE (French Institute of Statistics and Economic Studies).

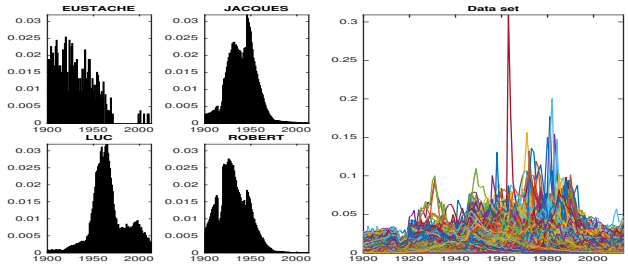

FIGURE 4.7: Children's first name at birth. An subet of 4 histograms representing the distribution of children born with that name per year in France, and the whole data set of $n = 1060$ histograms (right), displayed as pdf over the interval $[1900, 2013]$

This is an example of real data where log-PCA and GPCA are not equivalent procedures for $K = 2$ principal components. We recall that log-PCA leads to the computation of principal orthogonal directions $\tilde{u}_1, \tilde{u}_2$ in $L^2_{\bar{\mu}}(\Omega)$ minimizing (4.3.1). First observe that in the left column of Figure 4.8, for some data $\omega_i = \log_{\bar{\mu}}(\nu_i)$, the mappings $\tilde{T}_i = \mathrm{id} + \Pi_{\mathrm{Sp}(\{\tilde{u}_1\})}\omega_i$ are decreasing, and their range is larger than the interval Ω (that is, for some $x \in \Omega$, one has that $\tilde{T}_i(x) \notin \Omega$). Hence, such $\tilde{T}_i$ are not optimal mappings. Therefore, the condition $\Pi_{\mathrm{Sp}(\tilde{U})}\omega_i \in V_{\bar{\mu}}(\Omega)$ for all $1 \leq i \leq n$ (with $\tilde{U} = \{\tilde{u}_1, \tilde{u}_2\}$) is not satisfied, implying that log-PCA does not lead to a solution of GPCA thanks to Proposition 3.5 in [Bigot et al., 2015].

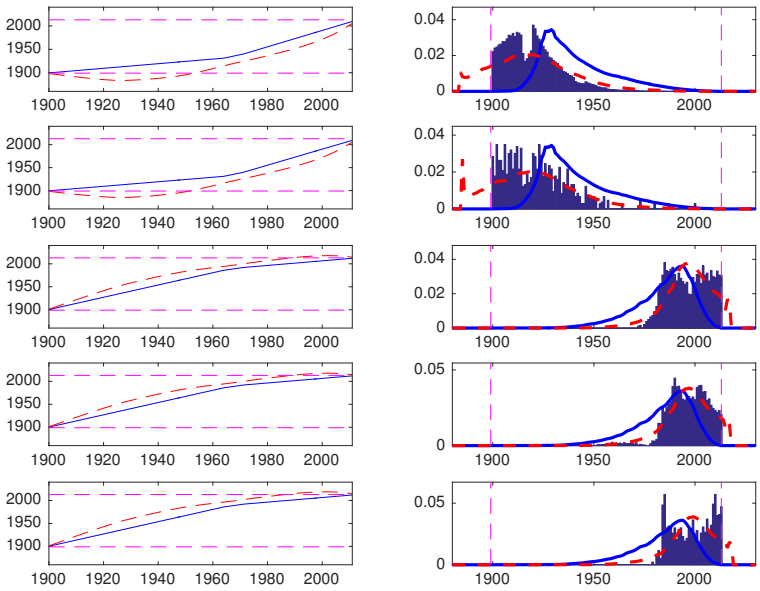

FIGURE 4.8: Children's first name at birth with support $\Omega = [1900, 2013]$. (Left) The dashed red curves represent the mapping $\tilde{T}_i = \mathrm{id} + \Pi_{\mathrm{Sp}(\{\tilde{u}_1\})}\omega_i$ where $\omega_i = \log_{\bar{\mu}}(\nu_i)$, and $\tilde{u}_1$ is the first principal direction in $L^2_{\bar{\mu}}(\Omega)$ obtained via log-PCA. The blue curves are the mapping $T_i = \mathrm{id} + \Pi_{\mathrm{Sp}(\{u_1^*\})}\omega_i$, where u_1^* is the first principal direction in $L^2_{\bar{\mu}}(\Omega)$ obtained via the iterative algorithm. (Right) The histogram stands for the pdf of measures ν_i that have a large Wasserstein distance with respect to the barycenter $\bar{\mu}$. The red curves are the pdf of the projection $\exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(\tilde{u}_1)}\omega_i)$ with log-PCA, while the blue curves are the pdf of the projection $\exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(u_1^*)}\omega_i)$ with GPCA.

Hence, for log-PCA, the corresponding histograms displayed in the right column of Figure 4.8 are such that $\Pi_{\mathrm{Sp}(\{\tilde{u}_1\})}\omega_i \notin V_{\bar{\mu}}(\Omega)$. This implies that the densities of the projected measures $\exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(\tilde{u}_1)}\omega_i)$ have a support outside $\Omega = [1900, 2013]$. Hence, the estimation of the measure $\nu_i = \exp_{\bar{\mu}}(\omega_i)$ by its projection onto the first mode of variation obtained with log-PCA is not satisfactory.

In Figure 4.8, we also display the results given by the iterative geodesic algorithm, leading to orthogonal directions u_1^*, u_2^* in $L^2_{\bar{\mu}}(\Omega)$ that are minimizers of (4.4.2). Contrary to the results obtained with log-PCA, one observes in Figure 4.8 that all the mapping $T_i = \mathrm{id} + \Pi_{\mathrm{Sp}(\{u_1^*\})}\omega_i$ are non-decreasing, and such that $T_i(x) \in \Omega$ for all $x \in \Omega$. Nevertheless, by enforcing these two conditions, one has that a good estimation of the measure $\nu_i = \exp_{\bar{\mu}}(\omega_i)$ by its projection $\exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(u_1^*)}\omega_i)$ is made difficult as most of the mass of ν_i is located at either the right or left side of the interval Ω which is not the case for its projection. The histograms displayed in the right column of Figure 4.8 correspond to the elements in the data set that have a large Wasserstein distance with respect to the barycenter $\bar{\mu}$. This explains why it is difficult to have good projected measures with GPCA. For elements in the data set that are closest to $\bar{\mu}$, the projected measures $\exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(\tilde{u}_1)}\omega_i)$ and $\exp_{\bar{\mu}}(\Pi_{\mathrm{Sp}(u_1^*)}\omega_i)$ are much closer to ν_i and for such elements, log-PCA and the iterative geodesic algorithm lead to similar results in terms of data projection.

To better estimate the extremal data in Figure 4.8, a solution is to increase the support of the data to the interval $\Omega_0 = [1850, 2050]$, and to perform log-PCA and GPCA in the Wasserstein space $W_2(\Omega_0)$. The results are reported in Figure 4.9. In that case, it can be observed that both algorithms lead to similar results, and that a better projection is obtained for the extremal data. Notice that with this extended support, all the mappings $\tilde{T}_i = \mathrm{id} + \Pi_{\mathrm{Sp}(\{\tilde{u}_1\})}\omega_i$ obtained with log-PCA are optimal in the sense that they are non-decreasing with a range inside Ω_0.

Finally, we display in Figure 4.10 and Figure 4.11 the results of the iterative and geodesic surface algorithms with either $\Omega = [1900, 2013]$ or with data supported on the extended support $\Omega_0 = [1850, 2050]$. The projection of the data onto the first principal geodesic suggests that the distribution of a name is deeply dependent on the part of the century. The second geodesic expresses a popular trend through a spike effect. In Figure 4.10, the artefacts in the principal surface that are obtained with the iterative algorithm at the end of the century, correspond to the fact that the projection of the data ω_i onto the surface spanned by the first two components is not ensured to belong to the set $V_{\bar{\mu}}(\Omega)$.

4.6 Conclusion

In this chapter we have shown how to leverage the Riemannian structure of the Wasserstein space of probability measures supported on the real line to perform geodesic PCA in that space. Our proposed algorithms can be useful for both visualization, by plotting modes of variations of a data set of histograms or probability densities, and also dimensionality reduction. The applicability is however limited since probability measures must be supported on the real line. In the next chapter, we investigate how to perform Wasserstein geodesic PCA in the more general setting of probability measures supported on a Hilbert space.

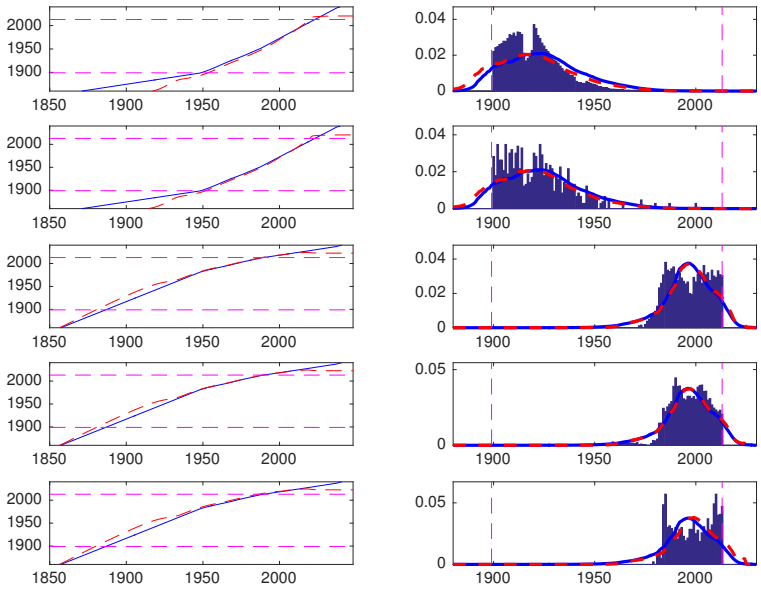

FIGURE 4.9: Children's first name at birth with extended support $\Omega_0 = [1850, 2050]$. (Left) The dashed red curves represent the mapping $\tilde{T}_i = \mathrm{id} + \Pi_{\mathrm{Sp}(\{\tilde{u}_1\})}\omega_i$ where $\omega_i = \log_{\tilde{\mu}}(\nu_i)$, and $\tilde{u}_1$ is the first principal direction in $L^2_{\tilde{\mu}}(\Omega)$ obtained via log-PCA. The blue curves are the mapping $T_i = \mathrm{id} + \Pi_{\mathrm{Sp}(\{u_1^*\})}\omega_i$, where u_1^* is the first principal direction in $L^2_{\tilde{\mu}}(\Omega)$ obtained via the iterative algorithm. (Right) The histogram stands for the pdf of measures ν_i that have a large Wasserstein distance with respect to the barycenter $\tilde{\mu}$. The red curves are the pdf of the projection $\exp_{\tilde{\mu}}(\Pi_{\mathrm{Sp}(\tilde{u}_1)}\omega_i)$ with log-PCA, while the blue curves are the pdf of the projection $\exp_{\tilde{\mu}}(\Pi_{\mathrm{Sp}(u_1^*)}\omega_i)$ with GPCA.

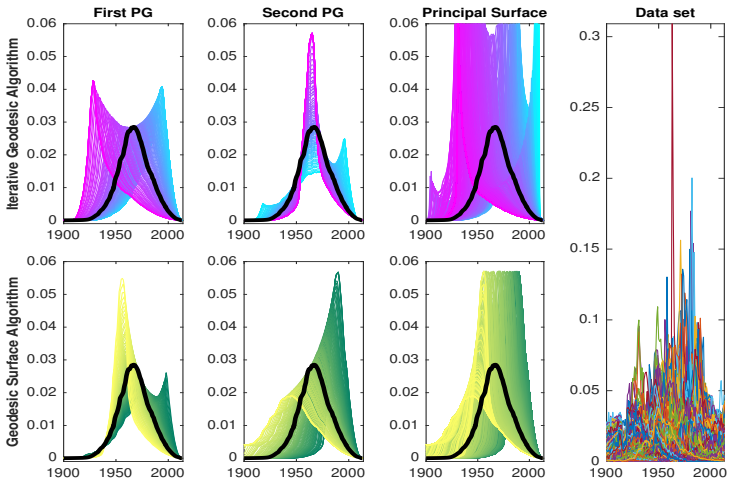

FIGURE 4.10: Children's first name at birth with support $\Omega = [1900, 2013]$. The first row is the GPCA of the data set obtained with the iterative geodesic approach. The second row is the GPCA through the geodesic surface approach. The first (resp. second) column is the projection of the data into the first (resp. second) principal direction. The black curve is the density of the Wasserstein barycenter. Colors encode the progression of the pdf of principal geodesic components in $W_2(\Omega)$.

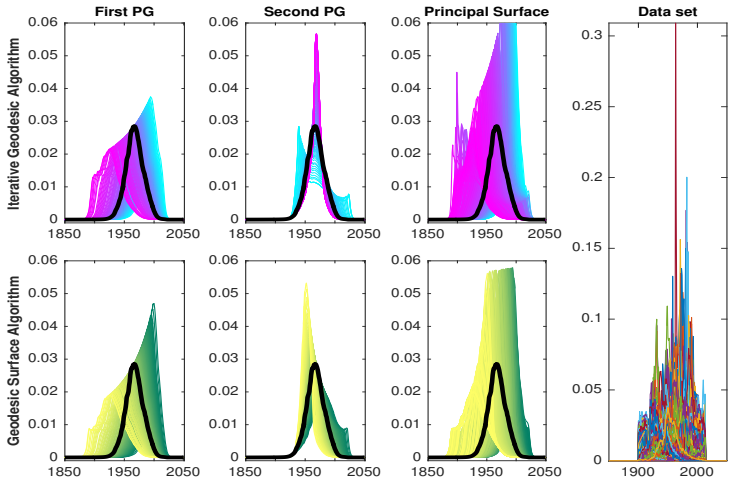

FIGURE 4.11: Children's first name at birth with extended support $\Omega_0 = [1850, 2050]$. The first row is the GPCA of the data set obtained with the iterative geodesic approach. The second row is the GPCA through the geodesic surface approach. The first (resp. second) column is the projection of the data into the first (resp. second) principal direction. The black curve is the density of the Wasserstein barycenter. Colors encode the progression of the pdf of principal geodesic components in $W_2(\Omega)$.

4.7 Appendix

We here detail the application of Algorithm (4.4.14) to the iterative GPCA procedure that consists in solving the problem (4.4.12):

$$\min_{\mathbf{v}\in\mathbb{R}^N}\min_{\mathbf{t}\in\mathbb{R}^n} J(\mathbf{v},\mathbf{t}) := \underbrace{\sum_{i=1}^{n}\|\mathbf{w}_i-(t_0+t_i)\mathbf{v}\|_{\bar{\mu}}^2}_{F(\mathbf{v},\mathbf{t})} + \underbrace{\chi_S(\mathbf{v})+\chi_D(\mathbf{v})+\chi_E(K\mathbf{v})+\chi_{B_1^n}(\mathbf{t})}_{G(\mathbf{v},\mathbf{t})}.$$

4.7.1 Lispschitz constant of ∇F

Let us now look at the Lipschitz constant of $\nabla F(\mathbf{v},\mathbf{t})$ on the restricted acceptable set $D\times B_1^n$. We first denote as $\mathcal{H}$ the Hessian matrix (of size $(N+n)\times(N+n)$) of the $\mathcal{C}^2$ function $F(X)$. We know that if the spectral radius of $\mathcal{H}$ is bounded by a scalar value M, i.e. $\rho(\mathcal{H})\leq M$, then ∇F is a Lipschitz continuous function with constant M. Hence, we look at the eigenvalues of the Hessian matrix of $F=\sum_{i=1}^n\sum_{j=1}^N \bar{f}_n(x_j)(w_i^j-(t_0+t_i)v_j)^2$ that is

$$\frac{\partial^2 F}{\partial t_i^2}=\sum_{j=1}^{N}2v_j^2\bar{f}_n(x_j),\quad \frac{\partial^2 F}{\partial v_j^2}=\sum_{i=1}^{n}2(t_0+t_i)^2\bar{f}_n(x_j),\quad \frac{\partial^2 F}{\partial t_i\partial v_j}=2\bar{f}_n(x_j)(2(t_0+t_i)v_j-w_i^j)$$

and $\frac{\partial^2 F}{\partial t_i\partial t_{i'}}=\frac{\partial^2 F}{\partial v_j\partial v_{j'}}=0$, for all $i\neq i'$ or $j\neq j'$. Being $\{\mu_k\}_{k=1}^{n+N}$ the eigenvalues of $\mathcal{H}$, we have $\rho(\mathcal{H})=\max_k|\mu_k|\leq\max_k\sum_l|\mathcal{H}_{kl}|$. We denote as $f_\infty=\max_j|\bar{f}_n(x_j)|$ and likewise $w_\infty=\max_{i,j}|w_i^j|$. Since $|t_0|<1$, $t_i^2\leq 1$, $\forall\mathbf{t}\in B_1^n$ and $v_j^2\leq\alpha^2=(b-a)^2$, $\forall\mathbf{v}\in D$, by defining $\gamma=2(1+|t_0|)\alpha+w_\infty$, we thus have

$$\rho(\mathcal{H})\leq 2f_\infty\max\left\{n\alpha^2+N\gamma, n\gamma+N(1+|t_0|)^2\right\}:=M. \tag{4.7.1}$$

4.7.2 Computing $\text{Prox}_{\tau G}$

In order to implement the algorithm (4.4.14), we finally need to compute the proximity operator of G defined as:

$$(\mathbf{v}^*, \mathbf{t}^*) = \text{Prox}_{\tau G}(\tilde{\mathbf{v}}, \tilde{\mathbf{t}}) = \underset{\mathbf{v},\mathbf{t}}{\text{argmin}} \frac{1}{2\tau}(||\mathbf{v}-\tilde{\mathbf{v}}||^2 + ||\mathbf{t}-\tilde{\mathbf{t}}||^2) + \chi_S(\mathbf{v}) + \chi_D(\mathbf{v}) + \chi_E(K\mathbf{v}) + \chi_{B_1^n}(\mathbf{t}).$$

This problem can be solved independently on $\mathbf{v}$ and $\mathbf{t}$. For $\mathbf{t}$, it can be done pointwise as $t_i^* = \text{argmin}_{t_i} \frac{1}{2\tau}||t_i - \tilde{t}_i||^2 + \chi_{B_1^1}(t_i) = \text{Proj}_{[-1;1]}(\tilde{t}_i)$. Unfortunately, there is no closed form expression of the proximity operator for the component $\mathbf{v}$. It requires to solve the following intern optimization problem at each extern iteration (ℓ) of the algorithm (4.4.14):

$$\mathbf{v}^* = \underset{\mathbf{v}}{\text{argmin}} \frac{1}{2\tau}||\mathbf{v}-\tilde{\mathbf{v}}||^2 + \chi_S(\mathbf{v}) + \chi_D(\mathbf{v}) + \chi_E(K\mathbf{v}), \tag{4.7.2}$$

where, to avoid confusions, we denote by $\mathbf{v}$ the variable that is optimized within the intern optimization problem (4.7.2).

Remark 4.7.1. The Lipschitz constant of $\nabla F(\mathbf{v}, \mathbf{t})$ in (4.7.1) relies independantly on $\mathbf{v}$ and $|t_0|$, thus we can choose the optimal gradient descent step τ for $\mathbf{v}^*$ and $\mathbf{t}^*$.

Primal-dual reformulation Using duality (through Fenchel transform), one has:

$$\begin{aligned} &\min_{\mathbf{v}\in\mathbb{R}^N} \frac{1}{2\tau}||\mathbf{v}-\tilde{\mathbf{v}}||^2 + \chi_S(\mathbf{v}) + \chi_D(\mathbf{v}) + \chi_E(K\mathbf{v}) \\ &= \min_{\mathbf{v}\in\mathbb{R}^N} \max_{\mathbf{z}\in\mathbb{R}^N} \frac{1}{2\tau}||\mathbf{v}-\tilde{\mathbf{v}}||^2 + \chi_S(\mathbf{v}) + \chi_D(\mathbf{v}) + \langle K\mathbf{v}, \mathbf{z}\rangle - \chi_E^*(\mathbf{z}), \end{aligned} \tag{4.7.3}$$

where $\mathbf{z} = \{z_j\}_{j=1}^N \in \mathbb{R}^N$ is a dual variable and $\chi_E^* = \sup_{\mathbf{v}} \langle \mathbf{v}, \mathbf{z}\rangle - \chi_E(\mathbf{v})$ is the convex conjugate of χ_E that reads:

$$(\chi_E^*(\mathbf{z}))_j = \begin{cases} -z_j/(1+t_0) & \text{if } z_j \leq 0, \\ z_j/(1-t_0) & \text{if } z_j > 0. \end{cases}$$

Hence, one can use the primal-dual algorithm proposed in [Chambolle and Pock, 2016] to solve the problem (4.7.3). For two parameters $\sigma, \theta > 0$ such that $||K||^2 \leq \frac{1}{\sigma}(\frac{1}{\theta} - \frac{1}{\tau})$ and given $\mathbf{v}^0, \bar{\mathbf{v}}^0, \mathbf{z}^0 \in \mathbb{R}^N$, the algorithm is:

$$\begin{cases} \mathbf{z}^{(m+1)} &= \text{Prox}_{\sigma\chi_E^*}(\mathbf{z}^{(m)} + \sigma K\bar{\mathbf{v}}^{(m)}) \\ \mathbf{v}^{(m+1)} &= \text{Prox}_{\theta(\chi_D+\chi_S)}(\mathbf{v}^{(m)} - \theta(K^*\mathbf{z}^{(m+1)} + \frac{1}{\tau}(\mathbf{v}^{(m)} - \tilde{\mathbf{v}})) \\ \bar{\mathbf{v}}^{(m+1)} &= 2\mathbf{v}^{(m+1)} - \mathbf{v}^{(m)} \end{cases} \tag{4.7.4}$$

where K^* is defined as $\langle K\mathbf{v}, \mathbf{z}\rangle = \langle \mathbf{v}, K^*\mathbf{z}\rangle$. Using the operator K defined in (4.4.11), we thus have:

$$(K^*\mathbf{z})_j = \begin{cases} -z_1/\Delta_1 & \text{if } j = 1 \\ z_{j-1}/\Delta_{j-1} - z_j/\Delta_j & \text{if } 1 < j < N. \\ z_{N-1}/\Delta_{N-1} & \text{if } j = N, \end{cases} \tag{4.7.5}$$

where $\Delta_j = x_{j+1} - x_j$. We have that $||K||^2 = \rho(K^*K)$, the largest eigenvalue of K^*K. With the discrete operators (4.4.11) and (4.7.5), $\rho(K^*K)$ can be bounded by

$$\delta^2 = 2 \max_j (1/\Delta_j^2 + 1/\Delta_{j+1}^2). \tag{4.7.6}$$

One can therefore for instance take $\sigma = \frac{1}{\delta}$ and $\theta = \tau/(1+\delta\tau)$.

Proximity operators in (4.7.4) The proximity operator of $\chi_D + \chi_S$ is the projection on $D \cap S$,

$$\text{Prox}_{\theta(\chi_D+\chi_S)}(\mathbf{v}) = \text{Proj}_{D\cap S}(\mathbf{v}) \tag{4.7.7}$$

which can be obtained by using for instance the Dykstra's projection algorithm [Boyle and Dykstra, 1986]. One can finally show that the proximity operator of χ_E^* can be computed pointwise as:

$$(\text{Prox}_{\sigma\chi_E^*}(\mathbf{z}))_j = \begin{cases} z_j - \sigma/(1-t_0) & \text{if } z_j > \sigma/(1-t_0) \\ z_j + \sigma/(1+t_0) & \text{if } z_j < -\sigma/(1+t_0) \\ 0 & \text{otherwise.} \end{cases} \tag{4.7.8}$$

4.7.3 Algorithms for GPCA

Gathering all the previous elements, we can finally find a critical point of the non-convex problem (4.4.12) using the forward-backward (FB) framework (4.4.14), as detailed in Algorithm 3.

Algorithm 3 Resolution with FB of problem (4.4.12): $\min_{\mathbf{v},\mathbf{t}} F(\mathbf{v},\mathbf{t}) + G(\mathbf{v},\mathbf{t})$

Require: $w_i \in \mathbb{R}^N$ for $i = 1 \cdots n$, $\mathbf{u}_1, \cdots \mathbf{u}_{k-1}$, $t_0 \in]-1;1[$, $\alpha = (b-a) > 0$, $\eta > 0$, $\delta > 0$ (defined in (4.7.6)) and $M > 0$ (defined in (4.7.1)).
Set $(\mathbf{v}^{(0)}, \mathbf{t}^{(0)}) \in D \times B_1^n$
Set $\tau < 1/M$, $\sigma = 1/\delta$ and $\theta = \tau/(1+\delta\tau)$.
%Extern loop:
while $||\mathbf{v}^{(\ell)} - \mathbf{v}^{(\ell-1)}||/||\mathbf{v}^{(\ell-1)}|| > \eta$ **do**
  % FB on $\mathbf{t}$ with $\mathbf{t}^{(\ell+1)} = Prox_{\tau G}(\mathbf{t}^{(\ell)} - \tau\nabla F(\mathbf{v}^{(\ell)}, \mathbf{t}^{(\ell)}))$:
  $t_i{}^{(\ell+1)} = \text{Proj}_{[-1;1]}\left(t_i{}^{(\ell)} - \tau\sum_{j=1}^N v_j{}^{(\ell)} \bar{f}_n(x_j)\left((t_0 + t_i{}^{(\ell)})v_j{}^{(\ell)} - w_i^j\right)\right)$
  % Gradient descent on $\mathbf{v}$ with $\tilde{\mathbf{v}} = \mathbf{v}^{(\ell)} - \tau\nabla F(\mathbf{v}^{(\ell)}, \mathbf{t}^{(\ell)})$:
  $\tilde{v}_j = v_j{}^{(\ell)} - \tau\bar{f}_n(x_j)\sum_{i=1}^n (t_0 + t_i{}^{(\ell)})\left((t_0 + t_i{}^{(\ell)})\, v_j{}^{(\ell)} - w_i^j\right)$
  %Intern loop for $\mathbf{v}^{(\ell+1)} = Prox_{\tau G}(\tilde{\mathbf{v}})$:
  Set $\mathbf{z}^{(0)} \in E$, $\mathbf{v}^{(0)} = \tilde{\mathbf{v}}$, $\bar{\mathbf{v}}^{(0)} = \tilde{\mathbf{v}}$
  while $||\mathbf{v}^{(m)} - \mathbf{v}^{(m-1)}||/||\mathbf{v}^{(m-1)}|| > \eta$ **do**
    $\mathbf{z}^{(m+1)} = \text{Prox}_{\sigma\chi_E^*}\left(\mathbf{z}^{(m)} + \sigma K\bar{\mathbf{v}}^{(m)}\right)$ (using (4.7.8))
    $\mathbf{v}^{(m+1)} = \text{Prox}_{\theta(\chi_D+\chi_S)}\left(\mathbf{v}^{(m)} - \theta(K^*\mathbf{z}^{(m+1)} + \frac{1}{\tau}(\mathbf{v}^{(m)} - \tilde{\mathbf{v}})\right)$ (using (4.7.7))
    $\bar{\mathbf{v}}^{(m+1)} = 2\mathbf{v}^{(m+1)} - \mathbf{v}^{(m)}$
    $m := m + 1$
  end while
  $\mathbf{v}^{(\ell+1)} = \mathbf{v}^{(m)}$
  $\ell := \ell + 1$
end while
return $\mathbf{u}_k = \mathbf{v}^{(\ell)}$

Geodesic surface approach In order to solve the problem (4.4.13), we follow the same steps as in the section 4.7.1-4.7.2. First we obtain the Lipchitz constant of the function $\tilde{F}$ by the same computations performed for the iterative algorithm. Then, since the constraints' problem in G' are separable, we can compute each component $\mathbf{v_k}$ and each $\mathbf{ff_i^{\pm}}$ independantly. The only difference with the iterative algorithm concerns the proximal operator of the function χ_A, which is the projection into the simplex of $\mathbb{R}^{2K}$.

Chapter 5

Principal Geodesic Analysis in the Wasserstein Space: The General Case

5.1 Introduction

In the previous chapter, we have seen how to exploit the structure of the Wasserstein space $W_2(\mathbb{R})$ of probability measures supported on the real line to compute principal geodesics. In this chapter, we are interested in generalizing this approach to the more general case of the Wasserstein space $W_2(\mathcal{X})$ of probability measures supported on a Hilbert space $\mathcal{X}$, typically $\mathbb{R}^d$ with d arbitrary. When $\mathcal{X}$ is not just metric but a Hilbert space, $W_2(\mathcal{X})$ is an infinite-dimensional Riemannian manifold (Ambrosio et al. 2006, Chap. 8; Villani 2008, Part II). Along with the previous chapter, three recent contributions by Boissard et al. [2015, §5.2], Bigot et al. [2015] and Wang et al. [2013] exploit directly or indirectly this structure to extend Principal Component Analysis (PCA) to $W_2(\mathcal{X})$. These important seminal papers are, however, limited in their applicability and/or the type of curves they output. Our goal in this chapter is to propose more general and scalable algorithms to carry out Wasserstein principal geodesic analysis on probability measures, and not simply dimensionality reduction as explained below.

Principal geodesics in $W_2(\mathcal{X})$ *vs.* dimensionality reduction on $P(\mathcal{X})$ We provide in Fig. 5.1 a simple example that illustrates the motivation of this study, and which also shows how our approach differentiates itself from existing dimensionality reduction algorithms (linear and non-linear) that draw inspiration from PCA. As shown in Fig. 5.1, linear PCA cannot produce components that remain in $W_2(\mathcal{X})$. Even more advanced tools, such as those proposed by Hastie and Stuetzle [1989], fall slightly short of that goal. On the other hand, Wasserstein geodesic analysis yields geodesic components in $W_2(\mathcal{X})$ that are easy to interpret and which can also be used to reduce dimensionality.

Foundations of PCA and Riemannian extensions Carrying out PCA on a family $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ of points taken in a space X can be described in abstract terms as: *(i)* define a mean element $\bar{\mathbf{x}}$ for that data set; *(ii)* define a family of *components* in X, typically geodesic curves, that contain $\bar{\mathbf{x}}$; *(iii)* fit a component by making it follow the x_i's as closely as possible, in the sense that the sum of the distances of each point $\mathbf{x}_i$ to that component is minimized; *(iv)* fit additional components by iterating step *(iii)* several times, with the added constraint that each new component is different (orthogonal) enough to the previous components. When X is Euclidean and the $\mathbf{x}_i$'s

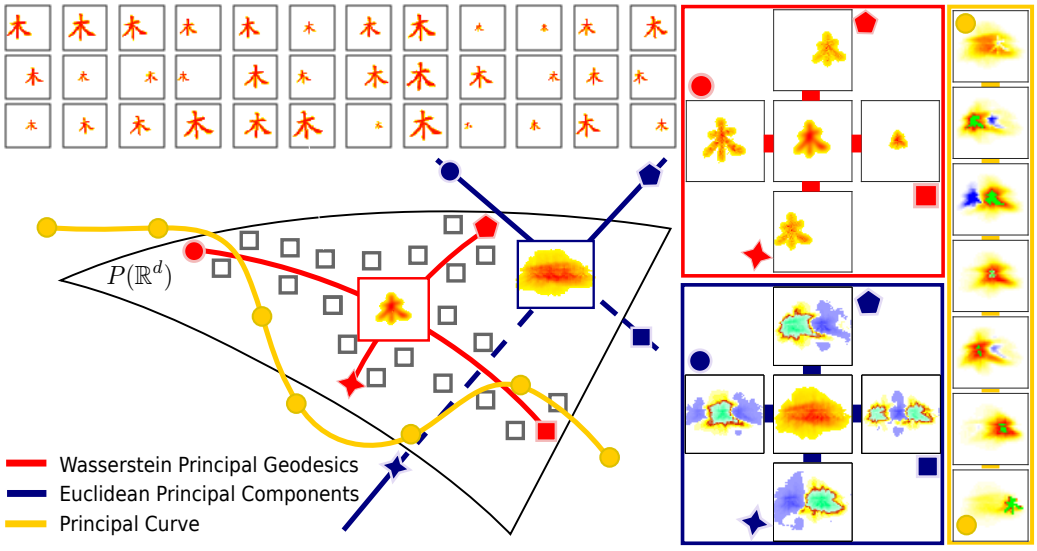

FIGURE 5.1: (Top-left) Data set: 60×60 images of a single Chinese character randomly translated, scaled and slightly rotated (36 images displayed out of 300 used). Each image is handled as a normalized histogram of $3,600$ non-negative intensities. (Middle-left) Data set schematically drawn on $W_2(\mathcal{X})$. The Wasserstein principal geodesics of this data set are depicted in red, its Euclidean components in blue, and its principal curve (Verbeek et al., 2002) in yellow. (Right) Actual curves (blue colors depict negative intensities, green intensities ≥ 1). Neither the Euclidean components nor the principal curve belong to $W_2(\mathcal{X})$, nor can they be interpreted as meaningful axis of variation.

are vectors in $\mathbb{R}^d$, the $(n+1)$-th component v_{n+1} can be computed iteratively by solving:

$$v_{n+1} \in \operatorname*{argmin}_{v \in V_n^\perp, \|v\|_2=1} \sum_{i=1}^{N} \min_{t \in \mathbb{R}} \|\mathbf{x}_i - (\bar{\mathbf{x}} + tv)\|_2^2, \text{ where } V_0 \stackrel{\text{def.}}{=} \emptyset, \text{ and } V_n \stackrel{\text{def.}}{=} \operatorname{span}\{v_1, \cdots, v_n\}. \tag{5.1.1}$$

Since PCA is known to boil down to a simple eigen-decomposition when X is Euclidean or Hilbertian [Schölkopf et al., 1997], Eq. (5.1.1) looks artificially complicated. This formulation is, however, extremely useful to generalize PCA to Riemannian manifolds [Fletcher et al., 2004]. This generalization proceeds first by replacing vector means, lines and orthogonality conditions using respectively Fréchet means [1948], geodesics, and orthogonality in tangent spaces. Riemannian PCA builds then upon the knowledge of the *exponential map* at each point $\mathbf{x}$ of the manifold X. Each exponential map $\exp_{\mathbf{x}}$ is locally bijective between the tangent space $T_{\mathbf{x}}$ of $\mathbf{x}$ and X. After computing the Fréchet mean $\bar{\mathbf{x}}$ of the data set, the logarithmic map $\log_{\bar{\mathbf{x}}}$ at $\bar{\mathbf{x}}$ (the inverse of $\exp_{\bar{\mathbf{x}}}$) is used to map all data points $\mathbf{x}_i$ onto $T_{\bar{\mathbf{x}}}$. Because $T_{\bar{\mathbf{x}}}$ is a Euclidean space by definition of Riemannian manifolds, the data set $(\log_{\bar{\mathbf{x}}} \mathbf{x}_i)_i$ can be studied using Euclidean PCA. Principal geodesics in X can then be recovered by applying the exponential map to a principal component v^*, $\{\exp_{\bar{\mathbf{x}}}(tv^*), |t| < \varepsilon\}$.

From Riemannian PCA to Wasserstein PCA: Related Work In the previous chapter and in [Boissard et al., 2015] and [Bigot et al., 2015], the geodesic PCA problem is studied in restricted scenarios: Bigot et al. [2015] and thus focused on measures supported on $\mathcal{X} = \mathbb{R}$, which simplifies the analysis since it is known in that case that the

Wasserstein space on the real line $W_2(\mathbb{R})$ can be embedded isometrically in $L^2(\mathbb{R})$ (see Chapter 4 Prop. 4.2.1). Wang et al. [2013] proposed a more general approach: given a family of input empirical measures $(\mu_1, \dots, \mu_N)$, they propose to compute first a "template measure" $\tilde{\mu}$ using k-means clustering on $\sum_i \mu_i$. They consider next all optimal transport plans π_i between that template $\tilde{\mu}$ and each of the measures μ_i, and propose to compute the barycentric projection (see Eq. 3.4.1) of each optimal transport plan π_i to recover Monge maps T_i, on which standard PCA can be used. This approach is computationally attractive since it requires the computation of only one optimal transport per input measure. Its weakness lies, however, in the fact that the curves in $W_2(\mathcal{X})$ obtained by displacing $\tilde{\mu}$ along each of these PCA directions are not geodesics in general.

Contributions and outline We propose a new algorithm to compute geodesic PCA in $W_2(\mathcal{X})$ for arbitrary Hilbert spaces $\mathcal{X}$. We use several approximations—both of the optimal transport metric and of its geodesics—to obtain tractable algorithms that can scale to thousands of measures. We provide first in §5.2 a review of the key concepts used in this work, namely Wasserstein distances and means, geodesics and tangent spaces in the Wasserstein space. We propose in §5.3 to parameterize a Wasserstein principal component (PC) using two velocity fields defined on the support of the Wasserstein mean of all measures, and formulate the geodesic PCA problem as that of optimizing these velocity fields so that the average distance of all measures to that PC is minimal. This problem is non-convex and non-smooth. We propose to optimize smooth upper-bounds of that objective using entropy regularized optimal transport in §5.4. The practical interest of our approach is demonstrated in §5.5 on toy samples, data sets of shapes and histograms of colors.

Notations We write $\langle A, B \rangle$ for the Frobenius dot-product of matrices A and B. $\mathbf{D}(u)$ is the diagonal matrix of vector u. We write p_1 and p_2 for the canonical projection operators $\mathcal{X}^2 \to \mathcal{X}$, defined as $p_1(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1$ and $p_2(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_2$.

5.2 The Riemannian Structure of $W_2(\mathcal{X})$

Wasserstein Geodesics Given two measures ν and η, let $\Pi^*(\nu, \eta)$ be the set of optimal couplings for between ν and η. Informally speaking, it is well known that if either ν or η are absolutely continuous measures, then any optimal coupling $\pi^* \in \Pi^*(\nu, \eta)$ is degenerated in the sense that, assuming for instance that ν is absolutely continuous, for all x in the support of ν only one point $y \in \mathcal{X}$ is such that $d\pi^*(\mathbf{x}, \mathbf{y}) > 0$. In that case, the optimal transport is said to have *no* mass splitting, and there exists an optimal mapping $T : \mathcal{X} \to \mathcal{X}$ such that π^* can be written, using a pushforward, as $\pi^* = (\mathrm{id} \times T)\#\nu$. When there is no mass splitting to transport ν to η, McCann's interpolant [1997]:

$$g_t = ((1-t)\mathrm{id} + tT)\#\nu, \quad t \in [0, 1], \tag{5.2.1}$$

defines a geodesic curve in the Wasserstein space, *i.e.* $(g_t)_t$ is locally the shortest path between any two measures located on the geodesic, with respect to W_2. In the more general case, where no optimal map T exists and mass splitting occurs (for some locations x one may have $d\pi^*(\mathbf{x}, \mathbf{y}) > 0$ for several y), then a geodesic can still be defined, but it relies on the optimal plan π^* instead: $g_t = ((1-t)p_1 + tp_2)\#\pi^*, t \in [0, 1]$, [Ambrosio et al., 2006, §7.2]. Both cases are shown in Fig. 5.2.

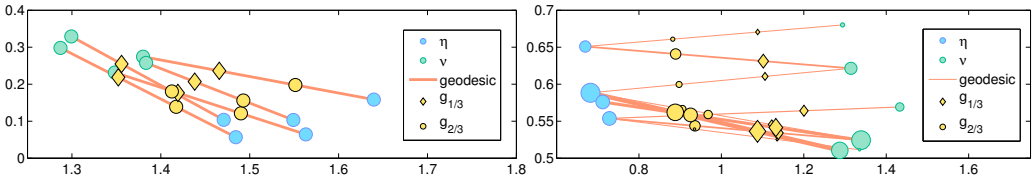

FIGURE 5.2: Both plots display geodesic curves between two empirical measures ν and η on $\mathbb{R}^2$. An optimal map exists in the left plot (no mass splitting occurs), whereas some of the mass of ν needs to be split to be transported onto η on the right plot.

Tangent Space and Tangent Vectors We briefly describe in this section the tangent spaces of $W_2(\mathcal{X})$, and refer to [Ambrosio et al., 2006, Chap. 8] for more details. Let $\mu : I \subset \mathbb{R} \to W_2(\mathcal{X})$ be a curve in $W_2(\mathcal{X})$. For a given time t, the tangent space of $W_2(\mathcal{X})$ at μ_t is a subset of $L^2(\mu_t, \mathcal{X})$, the space of square-integrable velocity fields supported on $\text{Supp}(\mu_t)$. At any t, there exists tangent vectors v_t in $L^2(\mu_t, \mathcal{X})$ such that $\lim_{h\to 0} W_2(\mu_{t+h}, (\text{id} + hv_t)\#\mu_t)/|h| = 0$. Given a geodesic curve in $W_2(\mathcal{X})$ parameterized as Eq. (5.2.1), its corresponding tangent vector at time zero is $v = T - \text{id}$.

5.3 Wasserstein Principal Geodesics

Geodesic parameterization The goal of principal geodesic analysis is to define geodesic curves in $W_2(\mathcal{X})$ that go through the mean $\bar{\mu}$ and which pass close enough to all target measures μ_i. To that end, geodesic curves can be parameterized with two end points ν and η. However, to avoid dealing with the constraint that a principal geodesic needs to go through $\bar{\mu}$, one can start instead from $\bar{\mu}$, and consider a velocity field $v \in L^2(\bar{\mu}, \mathcal{X})$ which displaces all of the mass of $\bar{\mu}$ in both directions:

$$g_t(v) \stackrel{\text{def.}}{=} (\text{id} + tv)\,\#\bar{\mu}, \quad t \in [-1, 1]. \tag{5.3.1}$$

Lemma 7.2.1 of Ambrosio et al. [2006] implies that any geodesic going through $\bar{\mu}$ can be written as Eq. (5.3.1) (see also Lemma 4.2.1 of the previous chapter). Hence, we do not lose any generality using this parameterization. However, given an arbitrary vector field v, the curve $(g_t(v))_t$ is not necessarily a geodesic. Indeed, the maps $\text{id} \pm v$ are not necessarily in the set $\mathcal{C}_{\bar{\mu}} \stackrel{\text{def.}}{=} \{r \in L^2(\bar{\mu}, \mathcal{X}) | (\text{id} \times r)\#\bar{\mu} \in \Pi^*(\bar{\mu}, r\#\bar{\mu})\}$ of optimal maps. Ensuring thus, at each step of our algorithm, that v is still such that $(g_t(v))_t$ is a geodesic curve is particularly challenging. To relax this strong assumption, we propose to use a generalized formulation of geodesics, which builds upon not one but *two* velocity fields, as introduced by Ambrosio et al. [2006, §9.2]:

Definition 4. *(adapted from [Ambrosio et al., 2006, §9.2]) Let $\sigma, \nu, \eta \in W_2(\mathcal{X})$, and assume there is an optimal mapping $T^{(\sigma,\nu)}$ from σ to ν and an optimal mapping $T^{(\sigma,\eta)}$ from σ to η. A* generalized *geodesic, illustrated in Fig. 5.3 between ν and η with base σ is defined by,*

$$g_t = \left((1-t)T^{(\sigma,\nu)} + tT^{(\sigma,\eta)}\right)\#\sigma, \quad t \in [0, 1].$$

Choosing $\bar{\mu}$ as the base measure in Definition 4, and two fields v_1, v_2 such that $\text{id} - v_1, \text{id} + v_2$ are optimal mappings (in $\mathcal{C}_{\bar{\mu}}$), we can define the following generalized geodesic $g_t(v_1, v_2)$:

$$g_t(v_1, v_2) \stackrel{\text{def.}}{=} (\text{id} - v_1 + t(v_1 + v_2))\,\#\bar{\mu}, \text{ for } t \in [0, 1]. \tag{5.3.2}$$

Generalized geodesics become true geodesics when v_1 and v_2 are positively proportional. We can thus consider a regularizer that controls the deviation from that property by defining $\Omega(v_1, v_2) = (\langle v_1, v_2 \rangle_{L^2(\bar{\mu},\mathcal{X})} - \|v_1\|_{L^2(\bar{\mu},\mathcal{X})}\|v_2\|_{L^2(\bar{\mu},\mathcal{X})})^2$, which is minimal when v_1 and v_2 are indeed positively proportional. We can now formulate the geodesic PCA problem as computing, for $n \geq 0$, the $(n+1)^{\text{th}}$ principal (generalized) geodesic component of a family of measures $(\mu_i)_i$ by solving, with $\lambda > 0$:

$$\min_{v_1,v_2 \in L^2(\bar{\mu},\mathcal{X})} \lambda\Omega(v_1, v_2) + \sum_{i=1}^{N} \min_{t\in[0,1]} W_2^2\left(g_t(v_1, v_2), \mu_i\right), \text{s.t.} \begin{cases} \text{id} - v_1, \text{id} + v_2 \in \mathcal{C}_{\bar{\mu}}, \\ v_1 + v_2 \in \text{span}(\{v_1^{(i)} + v_2^{(i)}\}_{i\leq n})^{\perp}. \end{cases} \tag{5.3.3}$$

This problem is not convex in v_1, v_2. We propose to find an approximation of that minimum by a projected gradient descent, with a projection that is to be understood in terms of an alternative metric on the space of vector fields $L^2(\bar{\mu}, \mathcal{X})$. To preserve the optimality of the mappings $\text{id} - v_1$ and $\text{id} + v_2$ between iterations, we introduce in the next paragraph a suitable projection operator on $L^2(\bar{\mu}, \mathcal{X})$.

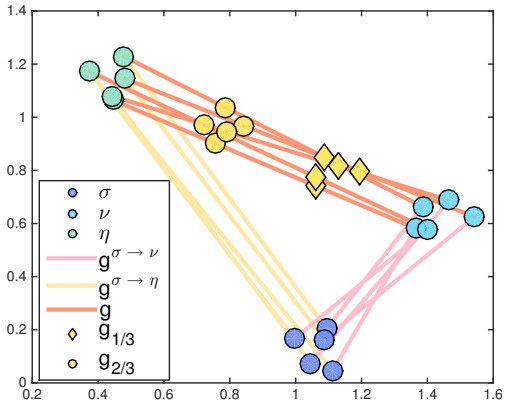

FIGURE 5.3: Generalized geodesic interpolation between two empirical measures ν and η using the base measure σ, all defined on $\mathcal{X} = \mathbb{R}^2$.

Remark 1. *A trivial way to ensure that $(g_t(v))_t$ is geodesic is to impose that the vector field v is a translation, namely that v is uniformly equal to a vector τ on all of Supp$(\bar{\mu})$. One can show in that case that the geodesic PCA problem described in Eq. (5.3.3) outputs an optimal vector τ which is the Euclidean principal component of the family formed by the means of each measure μ_i.*

Projection on the set of optimal maps We use a projected gradient descent method to solve Eq. (5.3.3) approximately. We will compute the gradient of a local upper-bound of the objective of Eq. (5.3.3) and update v_1 and v_2 accordingly. We then need to ensure that v_1 and v_2 are such that $\text{id} - v_1$ and $\text{id} + v_2$ belong to the set of optimal mappings $\mathcal{C}_{\bar{\mu}}$. To do so, we would ideally want to compute the projection r_2 of $\text{id} + v_2$ in $\mathcal{C}_{\bar{\mu}}$

$$r_2 = \operatorname*{argmin}_{r\in\mathcal{C}_{\bar{\mu}}} \|(\text{id} + v_2) - r\|^2_{L^2(\bar{\mu},\mathcal{X})}, \tag{5.3.4}$$

to update $v_2 \leftarrow r_2 - \text{id}$. Westdickenberg [2010] has shown that the set of optimal mappings $\mathcal{C}_{\bar{\mu}}$ is a convex closed cone in $L^2(\bar{\mu}, \mathcal{X})$, leading to the existence and the unicity of the solution of Eq. (5.3.4). However, there is to our knowledge no known method to compute the projection r_2 of $\text{id} + v_2$. There is nevertheless a well known and efficient approach to find a mapping r_2 in $\mathcal{C}_{\bar{\mu}}$ which is close to $\text{id} + v_2$. That approach, known as the the barycentric projection, requires to compute first an optimal coupling π^* between $\bar{\mu}$ and $(\text{id} + v_2)\#\bar{\mu}$, to define then a (conditional expectation) map

$$T_{\pi^*}(\mathbf{x}) \overset{\text{def.}}{=} \int_{\mathcal{X}} y d\pi^*(\mathbf{y}|\mathbf{x}). \tag{5.3.5}$$

Ambrosio et al. [2006, Theorem 12.4.4] or Reich [2013, Lemma 3.1] have shown that T_{π^*} is indeed an optimal mapping between $\bar{\mu}$ and $T_{\pi^*}\#\bar{\mu}$. We can thus set the velocity field as $v_2 \leftarrow T_{\pi^*} - \mathrm{id}$ to carry out an approximate projection. We show in the following paragraph that this operator can be in fact interpreted as a projection under a *pseudo*-metric $GW_{\bar{\mu}}$ on $L^2(\bar{\mu}, \mathcal{X})$.

Pseudo geodesic metric Let us first recall from [Ambrosio et al., 2006, Section 12.4] that the set of geodesics in the Wasserstein space can be identified to some plans having first marginal equal to $\bar{\mu}$,

$$\mathbf{G}(\bar{\mu}) = \left\{\pi \in P_2(\mathcal{X}^2),\ p_1\#\pi = \bar{\mu},\ (p_1, p_1 + \varepsilon p_2)\#\pi \text{ optimal for some } \varepsilon > 0\right\}. \tag{5.3.6}$$

Ambrosio et al. [2006, Definition 12.4.1] defined a metric on $\mathbf{G}(\bar{\mu})$,

$$W_{\bar{\mu}}(\pi_1, \pi_2)^2 = \min\left\{\int_{\mathcal{X}^3} |\mathbf{x}_3 - \mathbf{x}_2|^2 d\gamma,\ \gamma \in \Gamma(\pi_1, \pi_2)\right\}, \tag{5.3.7}$$

where $\Gamma(\pi_1, \pi_2) \subset P(X^3)$ is the set of a plans verifying $p_{12}\#\gamma = \pi_1$ and $p_{23}\#\gamma = \pi_2$, with $p_{12}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = (\mathbf{x}_1, \mathbf{x}_2)$ and $p_{13}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = (\mathbf{x}_1, \mathbf{x}_3)$. If for example π_2 is induced by a mapping T, namely $\pi_2 = (\mathrm{id} \times T)\#\bar{\mu}$, then this metric has the more simple expression [Ambrosio et al., 2006, page 316],

$$W_{\bar{\mu}}(\pi_1, \pi_2) = \left(\int_{\mathcal{X}^2} \|\mathbf{x}_2 - T(\mathbf{x}_1)\|_{\mathcal{X}}^2 d\pi_1(\mathbf{x}_1, \mathbf{x}_2)\right)^{1/2}. \tag{5.3.8}$$

Interestingly, if we look for the T which minimizes $W_{\bar{\mu}}(\pi_1, \pi_2)$ in Equation (5.3.9), we get that the solution is unique $\bar{\mu}$-almost surely and is equal to the barycentric projection of π_1. This can be seen by disintegrating π_1,

$$W_{\bar{\mu}}(\pi_1, \pi_2)^2 = \int_{\mathcal{X}} \left(\int_{\mathcal{X}} \|\mathbf{x}_2 - T(\mathbf{x}_1)\|_{\mathcal{X}}^2 d\pi_{1,\mathbf{x}_1}(\mathbf{x}_2)\right) d\bar{\mu}(\mathbf{x}_1). \tag{5.3.9}$$

For each $\mathbf{x}_1$ the minimum in the inner integral is achieved indeed for $T(\mathbf{x}_1) = \int_{\mathcal{X}} \mathbf{x}_2 d\pi_{1,\mathbf{x}_1}$, which is the barycentric projection of π_1. As seen above, if moreover π_1 is an optimal transport plan, then its barycentric projection is an optimal mapping. This observation motivates definition 5, which introduces a quantification of the difference between two vector fields which can be minimized with the barycentric projection.

Definition 5 (Geodesic pseudo metric on $L^2(\bar{\mu}, \mathcal{X})$)**.** *Let u and v in $L^2(\bar{\mu}, \mathcal{X})$. Let Π_o^u be the set of optimal transport plans between $\bar{\mu}$ and $(id + u)\#\bar{\mu}$, and Π_o^v be the set of optimal transport plans between $\bar{\mu}$ and $(id + v)\#\bar{\mu}$. We define,*

$$GW_{\bar{\mu}}(u, v) = \inf_{\pi_1 \in \Pi_o^u,\ \pi_2 \in \Pi_o^v} W_{\bar{\mu}}\left((p_1, p_2 - p_1)\#\pi_1, (p_1, p_2 - p_1)\#\pi_2\right)$$

which is the minimal distance between all geodesics starting from $\bar{\mu}$ and going through $(id + u)\#\bar{\mu}$ at time $t = 1$, and all geodesics starting from $\bar{\mu}$ and going through $(id + v)\#\bar{\mu}$ at time $t = 1$.

$GW_{\bar{\mu}}$ does not always satisfy the triangular inequality and is thus not a metric on $L^2(\bar{\mu}, \mathcal{X})$. $GW_{\bar{\mu}}$ becomes a metric when Π_o^w contains a unique element for any $w \in L^2(\bar{\mu}, \mathcal{X})$, which is the case for example if $\bar{\mu}$ admits a density. If moreover $\mathrm{id} + u$ and $\mathrm{id} + v$ are optimal mappings, then $\pi_1 = (\mathrm{id} \times (\mathrm{id} + u))\#\bar{\mu}$ and $\pi_2 = (\mathrm{id} \times (\mathrm{id} + v))\#\bar{\mu}$ are the unique optimal plans, and then $GW_{\bar{\mu}}(u, v) = \|v - u\|_{L^2(\bar{\mu}, \mathcal{X})}$.

To summarize, these results yield the following Proposition, which motivates the use of the barycentric projection.

Proposition 1. *Let v in $L^2(\bar{\mu}, \mathcal{X})$ and π_o^v an optimal transport plan between $\bar{\mu}$ and $(id + v)\#\bar{\mu}$. Assume π_o^v is unique and that there exists a solution w to,*

$$w \in \min_{id+u \in \mathcal{C}_{\bar{\mu}}(\mathcal{X})} GW_{\bar{\mu}}^2(u, v),$$

such that the optimal transport plan π_o^w between $\bar{\mu}$ and $(id + w)\#\bar{\mu}$ is unique. Then,

$$w = B((p_1, p_2 - p_1)\#\pi_o^v). \tag{5.3.10}$$

Proof. Since we assume that the solution w to the minimization problem has the property that there is a unique optimal transport plan between π_o^w between $\bar{\mu}$ and $(\mathrm{id} + w)\#\bar{\mu}$, it is equivalent to restrict to the u which also verify this property. The constraint $u \in \mathcal{C}_{\bar{\mu}}(\mathcal{X}) - \mathrm{id}$ means that $(\mathrm{id} \times (\mathrm{id} + u))\#\bar{\mu}$ is an optimal transport plan between $\bar{\mu}$ and $(\mathrm{id} + u)\#\bar{\mu}$, and then $(p_1, p_2 - p_1)\#\pi_o^u = (\mathrm{id} \times u)\#\bar{\mu}$. This leads to,

$$GW_{\bar{\mu}}^2(u, v) = \min_u \int_{\mathcal{X}^2} \|\mathbf{x}_2 - u(\mathbf{x}_1)\|_{\mathcal{X}}^2 d(p_1, p_2 - p_1)\#\pi_o^v(\mathbf{x}_1, \mathbf{x}_2),$$

which is minimum if and only if w is the barycentric projection of $(p_1, p_2 - p_1)\#\pi_u^v$ as discussed earlier. □

Although we are not able to compute a solution of Equation (5.3.4), the last proposition shows that substituting the L^2 norm in Eq. (5.3.4) by the pseudo metric defined in definition 5, we have an analytic solution which is simple to obtain through the computation of an optimal transport plan and a barycentric projection. As stated above, this pseudo metric and the $L^2_{\bar{\mu}}$ norm are equal on the subset $\mathcal{C}_{\bar{\mu}}(\mathcal{X}) - \mathrm{id}$ of $L^2(\bar{\mu}, \mathcal{X})$ when $\bar{\mu}$ admits a density.

5.4 Computing Principal Generalized Geodesics in Practice

We show in this section that when $\mathcal{X} = \mathbb{R}^d$, the steps outlined above can be implemented efficiently.

Input measures and their barycenter Each input measure in the family $(\mu_1, \cdots, \mu_N)$ is a finite weighted sum of Diracs, described by n_i points contained in a matrix X_i of size $d \times n_i$, and a (non-negative) weight vector a_i of dimension n_i summing to 1. The Wasserstein mean of these measures is given and equal to $\bar{\mu} = \sum_{k=1}^p b_k \delta_{\mathbf{y}_k}$, where the nonnegative vector $b = (b_1, \cdots, b_p)$ sums to one, and $Y = [\mathbf{y}_1, \cdots, \mathbf{y}_p] \in \mathbb{R}^{d \times p}$ is the matrix containing locations of $\bar{\mu}$.

Generalized geodesic Two velocity vectors for each of the p points in $\bar{\mu}$ are needed to parameterize a generalized geodesic. These velocity fields will be represented by two matrices $V_1 = [v_1^1, \cdots, v_p^1]$ and $V_2 = [v_1^2, \cdots, v_p^2]$ in $\mathbb{R}^{d \times p}$. Assuming that these velocity fields yield optimal mappings, the points at time t of that generalized geodesic are the measures parameterized by t,

$$g_t(V_1, V_2) = \sum_{k=1}^p b_k \delta_{\mathbf{z}_k^t}, \text{ with locations } Z_t = [\mathbf{z}_1^t, \ldots, \mathbf{z}_p^t] \stackrel{\text{def.}}{=} Y - V_1 + t(V_1 + V_2).$$

The squared 2-Wasserstein distance between datum μ_i and a point $g_t(V_1, V_2)$ on the geodesic is:

$$W_2^2(g_t(V_1, V_2), \mu_i) = \min_{P \in U(b, a_i)} \langle P, M_{Z_t X_i} \rangle, \tag{5.4.1}$$

where $U(b, a_i)$ is the transportation polytope $\{P \in \mathbb{R}_+^{p \times n_i}, P\mathbf{1}_{n_i} = b, P^T\mathbf{1}_p = a_i\}$, and $M_{Z_t X_i}$ stands for the $p \times n_i$ matrix of squared-Euclidean distances between the p and n_i column vectors of Z_t and X_i respectively. Writing $\tilde{Z}_t = \mathbf{D}(Z_t^T Z_t)$ and $\tilde{X}_i = \mathbf{D}(X_i^T X_i)$, we have that

$$M_{Z_t X_i} = \tilde{Z}_t \mathbf{1}_{n_i}^T + \mathbf{1}_p \tilde{X}_i^T - 2 Z_t^T X_i \in \mathbb{R}^{p \times n_i},$$

which, by taking into account the marginal conditions on $P \in U(b, a_i)$, leads to,

$$\langle P, M_{Z_t X_i} \rangle = b^T \tilde{Z}_t + a_i^T \tilde{X}_i - 2\langle P, Z_t^T X_i \rangle. \tag{5.4.2}$$

1. Majorization of the distance of each μ_i to the principal geodesic Using Eq. (5.4.2), the distance between each μ_i and the PC $(g_t(V_1, V_2))_t$ can be cast as a function f_i of (V_1, V_2):

$$f_i(V_1, V_2) \overset{\text{def.}}{=} \min_{t \in [0,1]} \left(b^T \tilde{Z}_t + a_i^T \tilde{X}_i + \min_{P \in U(b, a_i)} -2\langle P, (Y - V_1 + t(V_1 + V_2))^T X_i \rangle \right). \tag{5.4.3}$$

where we have replaced Z_t above by its explicit form in t to highlight that the objective above is quadratic convex plus piecewise linear concave as a function of t, and thus neither convex nor concave. Assume that we are given $P^\sharp$ and $t^\sharp$ that are approximate arg-minima for $f_i(V_1, V_2)$. For any A, B in $\mathbb{R}^{d \times p}$, we thus have that each distance $f_i(V_1, V_2)$ appearing in Eq. (5.3.3), is such that

$$f_i(A, B) \leqslant m_i^{V_1 V_2}(A, B) \overset{\text{def.}}{=} \langle P^\sharp, M_{Z_{t^\sharp} X_i} \rangle. \tag{5.4.4}$$

We can thus use a *majorization-minimization* procedure [Hunter and Lange, 2000] to minimize the sum of terms f_i by iteratively creating majorization functions $m_i^{V_1 V_2}$ at each iterate (V_1, V_2). All functions $m_i^{V_1 V_2}$ are quadratic convex. Given that we need to ensure that these velocity fields yield optimal mappings, and that they may also need to satisfy orthogonality constraints with respect to lower-order principal components, we use gradient steps to update V_1, V_2, which can be recovered using [Cuturi and Doucet, 2014, §4.3] and the chain rule as:

$$\nabla_1 m_i^{V_1 V_2} = 2(t^\sharp - 1)(Z_{t^\sharp} - X_i P^{\sharp T} \mathbf{D}(b^{-1})), \quad \nabla_2 m_i^{V_1 V_2} = 2t^\sharp (Z_{t^\sharp} - X_i P^{\sharp T} \mathbf{D}(b^{-1})). \tag{5.4.5}$$

2. Efficient approximation of $P^\sharp$ and $t^\sharp$ As discussed above, gradients for majorization functions $m_i^{V_1 V_2}$ can be obtained using approximate minima $P^\sharp$ and $t^\sharp$ for each function f_i. Because the objective of Eq. (5.4.3) is not convex w.r.t. t, we propose to do an exhaustive 1-d grid search with K values in $[0, 1]$. This approach would still require, in theory, to solve K optimal transport problems to solve Eq. (5.4.3) for each of the N input measures. To carry out this step efficiently, we propose to use entropy regularized transport [Cuturi, 2013], which allows for much faster computations and efficient parallelizations to recover approximately optimal transports $P^\sharp$.

3. Projected gradient update Velocity fields are updated with a gradient stepsize $\beta > 0$,

$$V_1 \leftarrow V_1 - \beta \left(\sum_{i=1}^{N} \nabla_1 m_i^{V_1 V_2} + \lambda \nabla_1 \Omega \right), \; V_2 \leftarrow V_2 - \beta \left(\sum_{i=1}^{N} \nabla_2 m_i^{V_1 V_2} + \lambda \nabla_2 \Omega \right),$$

followed by a projection step to enforce that V_1 and V_2 lie in $\text{span}(V_1^{(1)} + V_2^{(1)}, \cdots, V_1^{(n)} + V_2^{(n)})^{\perp}$ in the $L^2(\bar{\mu}, \mathcal{X})$ sense when computing the $(n+1)^{\text{th}}$ PC. We finally apply the barycentric projection operator defined in the end of §5.3. We first need to compute two optimal transport plans,

$$P_1^* \in \underset{P \in U(b,b)}{\text{argmin}} \langle P, M_{Y(Y-V_1)} \rangle, \quad P_2^* \in \underset{P \in U(b,b)}{\text{argmin}} \langle P, M_{Y(Y+V_2)} \rangle, \tag{5.4.6}$$

to form the barycentric projections, which then yield updated velocity vectors:

$$V_1 \leftarrow -\Big((Y - V_1) P_1^{*T} \mathbf{D}(b^{-1}) - Y \Big), \quad V_2 \leftarrow (Y + V_2) P_2^{*T} \mathbf{D}(b^{-1}) - Y. \tag{5.4.7}$$

We repeat steps 1,2,3 until convergence. Pseudo-code is provided in Alg. 4.

Algorithm 4 Compute the $(n+1)^{\text{th}}$ generalized geodesic principal component

1: **Input:** For $i \leqslant N$: $X_i \in \mathbb{R}^{d \times n_i}$, $a_i \in \mathbb{R}_+^{n_i}$ in the simplex. $Y \in \mathbb{R}^{d \times p}$, $b \in \mathbb{R}_+^p$ in the simplex. $K \in \mathbb{N}$, gradient step size $\beta > 0$, parameter $\lambda > 0$. V_1 and V_2 initial random matrices in $\mathbb{R}^{d \times p}$ with small norms.
2: **while** not converged **do**
3: For all i and $t_k = k/K$, form $M_{Z_{t_k} X_i}$ and solve Eq. (9).
4: For all i, compute the optimal projection time $t_i^\sharp$ and the corresponding optimal plan $P_i^\sharp$.
5: For all i, compute the gradients of m_i as in Eq. (13).
6: Update

$$V_1 \leftarrow V_1 - \beta \left(\sum_{i=1}^{N} \nabla_1 m_i^{V_1 V_2} + \lambda \nabla_1 \Omega \right), \; V_2 \leftarrow V_2 - \beta \left(\sum_{i=1}^{N} \nabla_2 m_i^{V_1 V_2} + \lambda \nabla_2 \Omega \right)$$

.
7: Project V_1 and V_2 on $\text{span}(V_1^{(1)} + V_2^{(1)}, \cdots, V_1^{(n)} + V_2^{(n)})^{\perp}$ in the $L^2_{\bar{\mu}}$ sense.
8: Compute the optimal plans P_1^* and P_2^* as in Eq. (14).
9: Update V_1 and V_2 through Eq. (15):

$$V_1 \leftarrow -\Big((Y - V_1) P_1^{*T} \mathbf{D}(b^{-1}) - Y \Big), \quad V_2 \leftarrow (Y + V_2) P_2^{*T} \mathbf{D}(b^{-1}) - Y.$$

10: **end while**

5.5 Experiments

Toy samples: We first run our algorithm on two simple synthetic examples. We consider respectively 4 and 3 empirical measures supported on a small number of locations in $\mathcal{X} = \mathbb{R}^2$, so that we can compute their exact Wasserstein means, using the multi-marginal linear programming formulation given in [Agueh and Carlier,

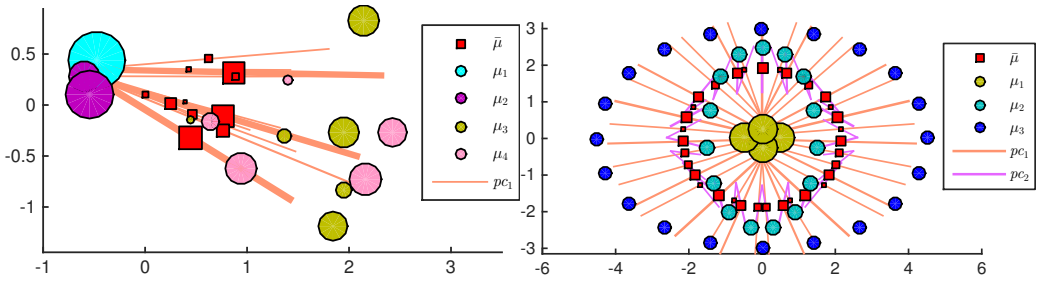

FIGURE 5.4: Wasserstein mean $\bar{\mu}$ and first PC computed on a data set of four (left) and three (right) empirical measures. The second PC is also displayed in the right figure.

2011, §4]. These measures and their mean (red squares) are shown in Fig. 5.4. The first principal component on the left example is able to capture both the variability of average measure locations, from left to right, and also the variability in the spread of the measure locations. On the right example, the first principal component captures the overall elliptic shape of the supports of all considered measures. The second principal component reflects the variability in the parameters of each ellipse on which measures are located. The variability in the weights of each location is also captured through the Wasserstein mean, since each single line of a generalized geodesic has a corresponding location and weight in the Wasserstein mean.

MNIST: For each of the digits ranging from 0 to 9, we sample 1,000 images in the MNIST data set representing that digit. Each image, originally a 28x28 grayscale image, is converted into a probability distribution on that grid by normalizing each intensity by the total intensity in the image. We compute the Wasserstein mean for each digit using the approach of Benamou et al. [2015]. We then follow our approach to compute the first three principal geodesics for each digit. Geodesics for four of these digits are displayed in Fig. 5.5 by showing intermediary (rasterized) measures on the curves. While some deformations in these curves can be attributed to relatively simple rotations around the digit center, more interesting deformations appear in some of the curves, such as the the loop on the bottom left of digit 2. Fig. 5.6 displays the first PC obtained on a subset of MNIST composed of 2,000 images of 2 and 4 in equal proportions.

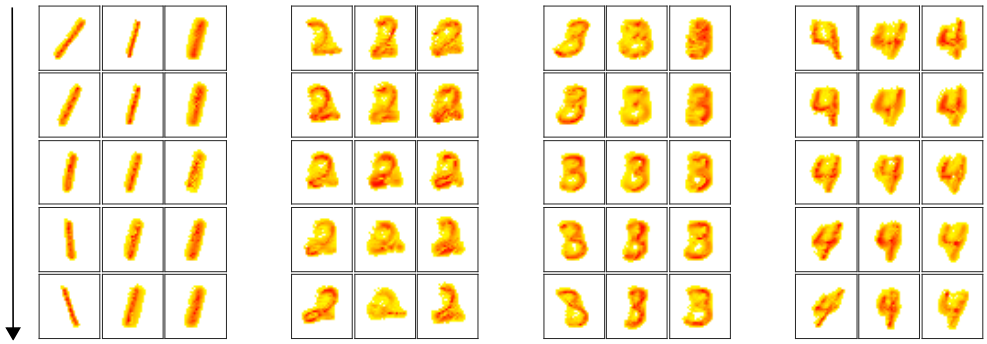

FIGURE 5.5: 1000 images for each of the digits 1,2,3,4 were sampled from the MNIST data set. We display above the first three PCs sampled at times $t_k = k/4$, $k = 0, \ldots, 4$ for each of these digits.

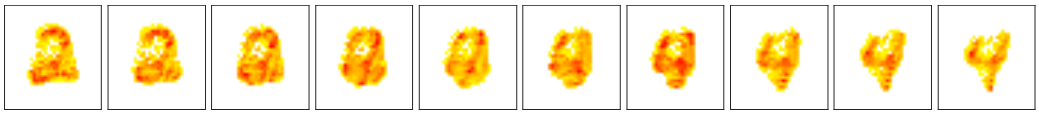

FIGURE 5.6: Samples from the first PC on a subset of the MNIST data set composed of one thousand 2s and one thousand 4s.
]

Color histograms: We consider a subset of the Caltech-256 Data set composed of three image categories: waterfalls, tomatoes and tennis balls, resulting in a set of 295 color images. The pixels contained in each image can be seen as a point-cloud in the RGB color space $[0,1]^3$. We use k-means quantization to reduce the size of these uniform point-clouds into a set of $k = 128$ weighted points, using cluster assignments to define the weights of each of the k cluster centroids. Each image can be thus regarded as a discrete probability measure of 128 atoms in the tridimensional RGB space. We then compute the Wasserstein barycenter of these measures supported on $p = 256$ locations using [Cuturi and Doucet, 2014, Alg.2]. Principal components are then computed as described in §5.4. The computation for a single PC is performed within 15 minutes on an iMac (3.4GHz Intel Core i7). Fig. 5.7 displays color palettes sampled along each of the first three PCs. The first PC suggests that the main source of color variability in the data set is the illumination, each pixel going from dark to light. Second and third PCs display the variation of colors induced by the typical images' dominant colors (blue, red, yellow). Fig. 5.8 displays the second PC, along with three images projected on that curve. The projection of a given image on a PC is obtained by finding first the optimal time t^* such that the distance of that image to the PC at t^* is minimum, and then by computing an optimal color transfer [Pitié et al., 2007] between the original image and the histogram at time t^*.

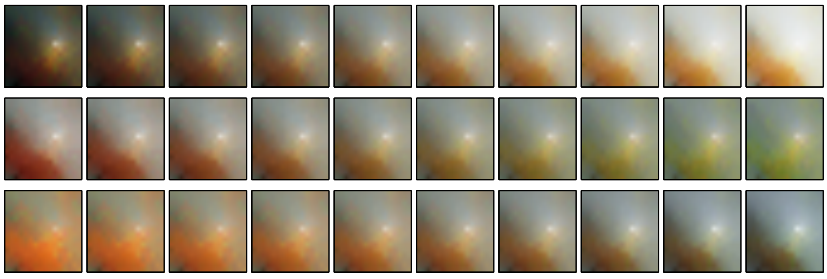

FIGURE 5.7: Each row represents a PC displayed at regular time intervals from $t = 0$ (left) to $t = 1$ (right), from the first PC (top) to the third PC (bottom).

Conclusion We have proposed an approximate projected gradient descent method to compute generalized geodesic principal components for probability measures. Our experiments suggest that these principal geodesics may be useful to analyze shapes and distributions, and that they do not require any parameterization of shapes or deformations to be used in practice.

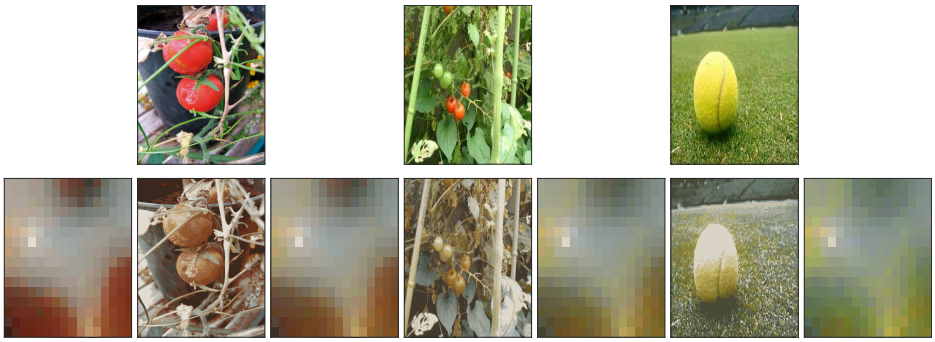

FIGURE 5.8: Color palettes from the second PC ($t = 0$ on the left, $t = 1$ on the right) displayed at times $t = 0, \frac{1}{3}, \frac{2}{3}, 1$. Images displayed in the top row are original; their projection on the PC is displayed below, using a color transfer with the palette in the PC to which they are the closest.

Chapter 6

Conclusion

6.1 Achieved work

In the present thesis, I have investigated how the concept of measure transport can be used for several purposes such as providing appealing visualizations of statistical data or in machine learning frameworks, starting in Chapter 2 with an application: the efficient computation of *cartograms*. Cartograms provide insightful density-equalizing geographic maps obtained by distorting a two-dimensional rectangular domain so that the surface area of each predefined region becomes proportional to some statistical data. The proposed flow-based algorithm was derived by observing that the computation of a transport map between an initial probability measure and its average density measure provides such a density-equalizing map. Measure transport can become more powerful when the transport map is optimal with respect to some underlying ground cost. Hence the concept of *optimal transport* has been the focus of the following chapters. As recent machine learning applications require algorithms which scale to larger and larger data sets, Chapter 3 has investigated the stochastic computation of regularized optimal transport, as well as how to learn approximately optimal maps by parameterizing them as a deep neural networks and minimizing the barycentric projection loss with respect to the parameters of these deep neural networks.

Such maps were shown to be useful in domain adaptation, and were even able to generate some realistic grayscale images of digits by applying the proposed algorithms between a 784-dimensional Gaussian measure and the MNIST data set of (28×28)-dimensional grayscale digit images. Going beyond computational aspects of optimal transport, Chapter 4 and Chapter 5 have focused on how to leverage the Riemannian structure of the space of probability measures equipped with the 2-Wasserstein distance (i.e. the optimal transport metric with respect to the squared Euclidean distance as the ground metric), in order to derive algorithms for finding principal geodesics of a data set of histograms or discrete probability measures. Both Chapter 4 and Chapter 5 have relied on the parameterization of curves in the space of probability measures and the explicit minimization of the 2-Wasserstein distances between the data set and a given parameterized curve through proximal-based or gradient-based optimization methods. Chapter 4 focused on probability measures supported on the real line, which is simpler than the general case thanks to some isometry property of $W_2(\mathbb{R})$ and the fact that optimal maps are gradient of nondecreasing functions. Hence it was possible to minimize the resulting objective function over the true set of geodesic curves, with some guarantees on the convergence of the proposed forward-backward algorithm. For the more general case of probability measures supported on a Hilbert space, which was the focus of Chapter 5, optimization was carried out over the set of generalized geodesics, a relaxed version of geodesics for which a projection step was proposed in order to remain in the

set of generalized geodesics along the gradient descent iterations of the proposed majorization-minimization algorithm. It was shown in numerical experiments that these principal geodesics can be useful for discovering and visualizing modes of variations of a data set of probability measures.

6.2 Future work

Overall, this thesis has presented new algorithms, applications and approaches regarding the use of measure transport for data visualization and learning. Each proposed approach has however some inherent limitations in their achieved goal or in their applicability, hence motivating further research. The density-equalizing maps computed in Chapter 2 are obtained through a transport map whose computation is simple but does not aim at having desirable properties such as approximate conformality, i.e. minimizing the distortion of angles, in order to obtain cartograms in which the shape of each region is preserved as much as possible. In Chapter 3, consistency theorems regarding the convergence of regularized optimal plans and their barycentric projection were proved in the case of the entropic regularization. First, it would be interesting to generalize such results to some other regularizers such as the squared L^2 norm which was also used in the experiments of this chapter. In addition, obtaining probabilistic bounds and rates of convergence of regularized discrete optimal plans and their barycentric projection with respect to the size of the empirical measures as well as the regularization amplitude would be of great interest in practice. Another aspect which has not been tackled is the choice of the ground cost. For both the domain adaptation task and the proposed generative optimal transport (GOT) model, the ground cost can have an important impact on the results. Hence, learning this ground cost through some learnable embeddings of the data may be a relevant approach, paving the way to challenging applications such as image-to-image translation. Finally, the approaches proposed in Chapter 4 and Chapter 5 for performing geodesic PCA in the Wasserstein space suffer from their computational complexity, preventing their applicability to large-scale data sets. Developing stochastic algorithms for minimizing the geodesic PCA objective rather than resorting to full-batch proximal or gradient-based optimization methods could alleviate this problem and hence unlock some powerful applications such as dimensionality reduction (with respect to the 2-Wasserstein distance) or filtering.

Bibliography

Martial Agueh and Guillaume Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

M. J. Alam, S. G. Kobourov, and S. Veeramoni. Quantitative measures for cartogram generation techniques. *Computer Graphics Forum*, 34(3):351–360, 2015. ISSN 1467-8659.

Jason Altschuler, Jonathan Weed, and Philippe Rigollet. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems*, pages 1961–1971, 2017.

Luigi Ambrosio and Nicola Gigli. A user's guide to optimal transport. *Editors*, page 1, 2013.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer, 2006.

J. D. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill, New York, 2nd edition, 1991.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

H. Attouch, J. Bolte, and B. Svaiter. Convergence of descent methods for semi-algebraic and tame problems. *Mathematical Programming*, 137(1-2):91–129, 2013.

A. Avinyó, J. Solà-Morales, and M. València. On maps with given Jacobians involving the heat equation. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 54 (6):919–936, 2003.

Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The Cramer distance as a solution to biased Wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3): 375–393, 2000.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

Patrick Bernard and Boris Buffoni. Optimal mass transportation and Mather theory. *arXiv preprint math/0412299*, 2004.

Dimitri P Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21(1):152–171, 1981.

Jérémie Bigot, Raúl Gouet, Thierry Klein, and Alfredo López. Geodesic PCA in the Wasserstein space by convex PCA. *Annales de l'Institut Henri Poincaré B: Probability and Statistics*, 2015.

Garrett Birkhoff. Three observations on linear algebra. *Univ. Nac. Tucumán. Revista A*, 5:147–151, 1946.

Mathieu Blondel, Vivien Seguy, and Antoine Rolet. Smooth and sparse optimal transport. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 880–889, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR. URL http://proceedings.mlr.press/v84/blondel18a.html.

Emmanuel Boissard, Thibaut Le Gouic, et al. On the mean speed of convergence of empirical and occupation measures in Wasserstein distance. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 50, pages 539–563. Institut Henri Poincaré, 2014.

Emmanuel Boissard, Thibaut Le Gouic, Jean-Michel Loubes, et al. Distribution's template estimate with Wasserstein metrics. *Bernoulli*, 21(2):740–759, 2015.

Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using Lagrangian mass transport. In *ACM Transactions on Graphics (TOG)*, volume 30, page 158. ACM, 2011.

Nicolas Bonneel, Gabriel Peyré, and Marco Cuturi. Wasserstein barycentric coordinates: Histogram regression using optimal transport. *ACM Transactions on Graphics*, 35(4), 2016.

James P Boyle and Richard L Dykstra. A method for finding projections onto the intersection of convex sets in Hilbert spaces. In *Advances in order restricted statistical inference*, pages 28–47. Springer, 1986.

Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.

R. P. Brent. Multiple-precision zero-finding methods and the complexity of elementary function evaluation. In J. F. Traub, editor, *Analytic Computational Complexity*, pages 151–176. Academic Press, New York, 1975.

R. G. Cano, K. Buchin, T. Castermans, A. Pieterse, W. Sonke, and B. Speckmann. Mosaic drawings and cartograms. *Computer Graphics Forum*, 34(3):361–370, 2015.

Guillaume Carlier. Optimal transportation and economic applications. *Lecture Notes.(Cited on page 2.)*, 2012.

Guillaume Carlier, Alfred Galichon, and Filippo Santambrogio. From Knothe's transport to Brenier's map and a continuation method for optimal transport. *SIAM Journal on Mathematical Analysis*, 41(6):2554–2576, 2010.

C. Cauvin and C. Schneider. Cartographic transformations and the piezopleth maps method. *The Cartographic Journal*, 26(2):96–104, 1989.

Elsa Cazelles, Vivien Seguy, Jérémie Bigot, Marco Cuturi, and Nicolas Papadakis. Geodesic PCA versus log-PCA of histograms in the Wasserstein space. *SIAM Journal on Scientific Computing*, 40(2):B429–B456, 2018. doi: 10.1137/17M1143459. URL https://doi.org/10.1137/17M1143459.

Antonin Chambolle and Thomas Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 159(1-2):253–287, 2016.

Thierry Champion, Luigi De Pascale, et al. The Monge problem in Rd. *Duke mathematical journal*, 157(3):551–572, 2011.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: geometry and Kantorovich formulation. *arXiv preprint arXiv:1508.05216*, 2015.

Lenaic Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. An interpolating distance between optimal transport and Fisher–Rao metrics. *Foundations of Computational Mathematics*, pages 1–44, 2016a.

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

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Unbalanced optimal transport: Dynamic and Kantorovich formulations. *Journal of Functional Analysis*, 2018.

Roberto Cominetti and Jaime San Martín. Asymptotic analysis of the exponential penalty trajectory in linear programming. *Mathematical Programming*, 67(1-3):169–187, 1994.

Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 274–289. Springer, 2014.

Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3730–3739. Curran Associates, Inc., 2017a. URL http://papers.nips.cc/paper/6963-joint-distribution-optimal-transportation-for-domain-adaptation.pdf.

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

Nicolas Courty, Remi Flamary, and Melanie Ducoffe. Learning Wasserstein embeddings. In *International Conference on Learning Representations (ICLR)*, 2018.

Imre Csiszár, Paul C Shields, et al. Information theory and statistics: A tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4):417–528, 2004.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013.

Marco Cuturi and Arnaud Doucet. Fast computation of Wasserstein barycenters. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 685–693, 2014.

B. Dacorogna and J. Moser. On a partial differential equation involving the Jacobian determinant. *Annales de l'IHP Analyse non linéaire*, 7(1):1–26, 1990.

George B Dantzig. Application of simplex method to a transportation problem. *Activity analysis of production and allocation*, 1951.

B. S. Daya Sagar. Cartograms via mathematical morphology. *Information Visualization*, 13(1):42–58, 2014.

R. D. de Veaux, P. F. Velleman, and D. E. Bock. *Stats: Data and Models*. Pearson, Harlow, 4th edition, 2016.

Arnaud Dessein, Nicolas Papadakis, and Jean-Luc Rouas. Regularized optimal transport and the rot mover's distance. *arXiv preprint arXiv:1610.06447*, 2016.

D. Dorling. *Area cartograms: their use and creation*. Concepts and Techniques in Modern Geography. Environmental Publications, Norwich, 1996.

D. Dorling. Mapping disease patterns. In *Encyclopedia of Biostatistics*. John Wiley & Sons, Hoboken, 2005.

D. Dorling. *The 32 Stops: The Central Line*. Penguin Books, London, 2013.

J. A. Dougenik, N. R. Chrisman, and D. R. Niemeyer. An algorithm to construct continuous area cartograms. *Professional Geographer*, 37(1):75–81, 1985.

H. Edelsbrunner and R. Waupotitsch. A combinatorial approach to cartograms. *Computational Geometry*, 7(5):343–360, 1997.

Lawrence C Evans and Wilfrid Gangbo. *Differential equations methods for the Monge-Kantorovich mass transfer problem*, volume 653. American Mathematical Soc., 1999.

Lawrence Craig Evans and Ronald F Gariepy. *Measure theory and fine properties of functions*. CRC press, 2015.

Sira Ferradans, Nicolas Papadakis, Gabriel Peyré, and Jean-François Aujol. Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, 7(3):1853–1882, 2014.

Denis Feyel and Ali S Üstünel. Monge-Kantorovitch measure transportation and Monge-Ampère equation on Wiener space. *Probability theory and related fields*, 128 (3):347–385, 2004.

P. Thomas Fletcher. Geodesic regression on Riemannian manifolds. In *Proceedings of the Third International Workshop on Mathematical Foundations of Computational Anatomy-Geometrical and Statistical Methods for Modelling Biological Shape Variability*, pages 75–86, 2011.

P. Thomas Fletcher. Geodesic regression and the theory of least squares on Riemannian manifolds. *International journal of computer vision*, 105(2):171–185, 2013.

P. Thomas Fletcher, Conglin Lu, Stephen M. Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *Medical Imaging, IEEE Transactions on*, 23(8):995–1005, 2004.

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3-4): 707–738, 2015.

Maurice Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. In *Annales de l'institut Henri Poincaré*, volume 10, pages 215–310. Presses universitaires de France, 1948.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

M. Frigo and S. G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a Wasserstein loss. In *Advances in Neural Information Processing Systems*, pages 2053–2061, 2015.

L. Gamio. Election maps are telling you big lies about small things. The Washington Post, 1 Nov 2016, https://www.washingtonpost.com/graphics/politics/2016-election/how-election-maps-lie/, 2016. Accessed 25 Apr 2017.

M. T. Gastner, C. R. Shalizi, and M. E. J. Newman. Maps and cartograms of the 2004 US presidential election results. *Advances in Complex Systems*, 8(1):117–123, 2005.

Michael T Gastner and Mark EJ Newman. Diffusion-based method for producing density-equalizing maps. *Proceedings of the National Academy of Sciences of the United States of America*, 101(20):7499–7504, 2004.

Michael T Gastner, Vivien Seguy, and Pratyush More. Fast flow-based algorithm for creating density-equalizing map projections. *Proceedings of the National Academy of Sciences*, 115(10):E2156–E2164, 2018.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. In *Advances in Neural Information Processing Systems*, pages 3432–3440, 2016.

Aude Genevay, Gabriel Peyre, and Marco Cuturi. Learning generative models with sinkhorn divergences. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1608–1617, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR. URL http://proceedings.mlr.press/v84/genevay18a.html.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

S. M. Gusein-Zade and V. S. Tikunov. A new technique for constructing continuous cartograms. *Cartography and Geographic Information Systems*, 20(3):167–173, 1993.

Steven Haker, Lei Zhu, Allen Tannenbaum, and Sigurd Angenent. Optimal mass transport for registration and warping. *International Journal of computer vision*, 60 (3):225–240, 2004.

Trevor Hastie and Werner Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

R. Heilmann, D. A. Keim, C. Panse, and M. Sips. Recmap: Rectangular map approximations. In *IEEE Symposium on Information Visualization*, pages 33–40, 2004. doi: 10.1109/INFVIS.2004.57.

B. Hennig. *Rediscovering the World*. Springer, Berlin, 2013.

R. Henriques, F. Bação, and V. Lobo. Carto-SOM: Cartogram creation using self-organizing maps. *International Journal of Geographical Information Science*, 23(4): 483–511, 2009.

B. Hopkins and J. G. Skellam. A new method for determining the type of distribution of plant individuals. *Annals of Botany*, 18(2):213–227, 1954.

D. H. House and C. J. Kocmoud. Continuous cartogram construction. In *Proceedings of the IEEE Conference on Visualization*, pages 197–204, Oct 1998.

David R Hunter and Kenneth Lange. Quantile regression via an mm algorithm. *Journal of Computational and Graphical Statistics*, 9(1):60–77, 2000.

R. Inoue and E. Shimizu. A new algorithm for continuous area cartogram construction with triangulation of regions and restriction on bearing changes of edges. *Cartography and Geographic Information Science*, 33(2):115–125, 2006.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image–to–image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017.

Xianhua Jiang, Zhi-Quan Luo, and Tryphon T Georgiou. Geometric methods for spectral analysis. *IEEE Transactions on Signal Processing*, 60(3):1064–1074, 2012.

J. H. Kämper, S. G. Kobourov, and M. Nöllenburg. Circular-arc cartograms. In *2013 IEEE Pacific Visualization Symposium (PacificVis)*, pages 1–8, Feb 2013.

Leonid Vitalievich Kantorovich. On the translocation of masses. In *Dokl. Akad. Nauk SSSR*, volume 37, pages 199–201, 1942.

D. A. Keim, S. C. North, C. Panse, and J. Schneidewind. Visualizing geographic information: VisualPoints vs CartoDraw. *Information Visualization*, 2(1):58–67, 2003.

D. A. Keim, S. C. North, and C. Panse. Cartodraw: a fast algorithm for generating contiguous cartograms. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):95–110, 2004.

D. A. Keim, C. Panse, and S. C. North. Medial-axis-based cartograms. *IEEE Computer Graphics and Applications*, 25(3):60–68, 2005.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

Herbert Knothe et al. Contributions to the theory of convex bodies. *The Michigan Mathematical Journal*, 4(1):39–52, 1957.

Soheil Kolouri, Se Rim Park, Matthew Thorpe, Dejan Slepcev, and Gustavo K Rohde. Optimal mass transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine*, 34(4):43–59, 2017.

Steven G Krantz and Harold R Parks. *Geometric integration theory*. Springer Science & Business Media, 2008.

Harold W Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97, 1955.

Solomon Kullback. *Information theory and statistics*. Courier Corporation, 1997.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. ISSN 00189219. doi: 10.1109/5.726791.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pages 2200–2210, 2017.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1718–1727, 2015.

Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new Hellinger-Kantorovich distance between positive measures. *Inventiones mathematicae*, 211(3):969–1117, 2018.

D. E. Lilienfeld and P. D. Stolley. *Foundations of Epidemiology*. Oxford University Press, New York, 3rd edition, 1994.

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pages 2200–2207, 2013.

D. A. Lovett, A. J. Poots, J. T. C. Clements, S. A. Green, E. Samarasundera, and D. Bell. Using geographical information systems and cartograms as a health service quality improvement tool. *Spatial and spatio-temporal epidemiology*, 10:67–74, 2014.

K. Ma. 2012 Electoral Vote. https://commons.wikimedia.org/wiki/File:Cartogram%E2%80%942012_Electoral_Vote.svg, 2012. Accessed 09 Feb 2018.

Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. An introduction to sampling via measure transport. *arXiv preprint arXiv:1602.05023*, 2016.

Robert J McCann. A convexity principle for interacting gases. *Advances in mathematics*, 128(1):153–179, 1997.

Quentin Mérigot. A multiscale approach to optimal transport. In *Computer Graphics Forum*, volume 30, pages 1583–1592, 2011.

D. W. Merrill, S. Selvin, and M. S. Mohr. Density equalizing map projections: A new algorithm. Technical Report LBL-31984, Lawrence Berkeley Laboratory, Feb 1992.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

G. Monge. Mémoire sur la theorie des deblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

Grégoire Montavon, Klaus-Robert Müller, and Marco Cuturi. Wasserstein training of restricted Boltzmann machines. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3718–3726. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6248-wasserstein-training-of-restricted-boltzmann-machines.pdf.

Tarek A. El Moselhy and Youssef M. Marzouk. Bayesian inference with optimal maps. *Journal of Computational Physics*, 231(23):7815 – 7850, 2012. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2012.07.022. URL http://www.sciencedirect.com/science/article/pii/S0021999112003956.

J. Moser. On the volume elements on a manifold. *Transactions of the American Mathematical Society*, 120(2):286–294, 1965.

Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

Boris Muzellec, Richard Nock, Giorgio Patrini, and Frank Nielsen. Tsallis regularized optimal transport and ecological inference. In *AAAI*, pages 2387–2393, 2017.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

Andrew Y Ng and Michael I Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*, pages 841–848, 2002.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pages 271–279, 2016.

S. Nusrat and S. Kobourov. The state of the art in cartograms. *Computer Graphics Forum*, 35(3):619–642, 2016.

P. Ochs, Y. Chen, T. Brox, and T. Pock. ipiano: Inertial proximal algorithm for non-convex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

Office for National Statistics. Land area and population density for MSOA and LSOA, 2015. Accessed 25 Apr 2017.

Office for National Statistics. Number of deaths from all causes, by sex, age and LSOA 2001 and 2011, England and Wales, deaths registered 2001 to 2014. https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/causesofdeath/, 2016. Accessed 25 Apr 2017.

J. M. Olson. Noncontiguous area cartograms. *The Professional Geographer*, 28(4):371–380, 1976.

Nicolas Papadakis, Gabriel Peyré, and Edouard Oudet. Optimal transport with proximal splitting. *SIAM Journal on Imaging Sciences*, 7(1):212–238, 2014.

A. Papadopoulos. Quasiconformal mappings, from Ptolemy's geography to the work of Teichmüller. arXiv:1702.03756, 2017.

A. Parlapiano. There are many ways to map election results. We've tried most of them. The New York Times, 1 Nov 2016, https://www.nytimes.com/interactive/2016/11/01/upshot/many-ways-to-map-election-results.html, 2016. Accessed 25 Apr 2017.

Michaël Perrot, Nicolas Courty, Rémi Flamary, and Amaury Habrard. Mapping estimation for discrete optimal transport. In *Advances in Neural Information Processing Systems*, pages 4197–4205, 2016.

A. Petersen and H.-G. Müller. Functional data analysis for density functions by transformation to a Hilbert space. *The Annals of Statistics*, 44(1):183–218, 2016. doi: 10.1214/15-AOS1363.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. Technical report, 2017.

François Pitié, Anil C Kokaram, and Rozenn Dahyot. Automated colour grading using colour distribution transfer. *Computer Vision and Image Understanding*, 107 (1):123–137, 2007.

Sebastian Reich. A dynamical systems framework for intermittent data assimilation. *BIT Numerical Mathematics*, 51(1):235–249, 2011.

Sebastian Reich. A nonparametric ensemble transform method for Bayesian inference. *SIAM Journal on Scientific Computing*, 35(4):A2013–A2024, 2013.

Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

Antoine Rolet, Marco Cuturi, and Gabriel Peyré. Fast dictionary learning with a smoothed Wasserstein loss. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 630–638, 2016.

Antoine Rolet, Vivien Seguy, Mathieu Blondel, and Hiroshi Sawada. Blind source separation with optimal transport non-negative matrix factorization. *arXiv preprint arXiv:1802.05429*, 2018.

Murray Rosenblatt. Remarks on a multivariate transformation. *The annals of mathematical statistics*, 23(3):470–472, 1952.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.

Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, pages 99–102, 2015.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Artificial Neural Networks ICANN'97*, pages 583–588. Springer, 1997.

Erwin Schrödinger. *Über die umkehrung der naturgesetze*. Verlag Akademie der wissenschaften in kommission bei Walter de Gruyter u. Company, 1931.

Vivien Seguy and Marco Cuturi. Principal geodesic analysis for probability measures under the optimal transport metric. In *Advances in Neural Information Processing Systems*, pages 3312–3320, 2015.

Vivien Seguy, Bharath Bhushan Damodaran, Rémi Flamary, Nicolas Courty, Antoine Rolet, and Mathieu Blondel. Large-scale optimal transport and mapping estimation. In *Proceedings of the International Conference in Learning Representations*, 2018.

S. Selvin, D. Merrill, S. Sacks, L. Wong, L. Bedell, and J. Schulman. Transformations of maps to investigate clusters of disease. Technical Report LBL-18550, Lawrence Berkeley Laboratory, Oct 1984.

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

S. S. Skiena. *The Algorithm Design Manual*. Springer, London, 2008.

J. P. Snyder. Map projections – a working manual. Technical Report 1395, U.S. Government Printing Office, Washington, 1987.

Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. Earth mover's distances on discrete surfaces. *ACM Transactions on Graphics (TOG)*, 33 (4):67, 2014.

S. Sommer, F. Lauze, S. Hauberg, and M. Nielsen. Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision ECCV 2010*, volume 6316 of *Lecture Notes in Computer Science*, pages 43–56. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15566-6. doi: 10.1007/978-3-642-15567-3_4. URL http://dx.doi.org/10.1007/978-3-642-15567-3_4.

Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, Gert RG Lanckriet, et al. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

Statistics Times. Indian states by GDP. http://statisticstimes.com/economy/gdp-of-indian-states.php, 2017. Accessed 3 May 2017.

Zhengyu Su, Yalin Wang, Rui Shi, Wei Zeng, Jian Sun, Feng Luo, and Xianfeng Gu. Optimal mass transport for shape matching and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(37):2246–2259, 2015.

S. Sun. A fast, free-form rubber-sheet algorithm for contiguous area cartograms. *International Journal of Geographical Information Science*, 27(3):567–593, 2013.

The Economist. Kensington and Chelsea: a wealthy but deeply divided borough. 24 June 2017, https://www.economist.com/news/britain/21723839-grenfell-tower-fire-has-become-stark-reminder-glaring-gap-between-rich-and-poor-even, 2017. Accessed 23 Dec 2017.

W. Tobler. Thirty five years of computer cartograms. *Annals of the Association of American Geographers*, 94(1):58–73, 2004.

W. R. Tobler. A continuous transformation useful for districting. *Annals of the New York Academy of Sciences*, 219(1):215–220, 1973.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

Vladimir Naumovich Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, 2002.

Jakob J Verbeek, Nikos Vlassis, and B Kröse. A k-segments algorithm for finding principal curves. *Pattern Recognition Letters*, 23(8):1009–1017, 2002.

R. Verde, A. Irpino, and A. Balzanella. Dimension reduction techniques for distributional symbolic data. *IEEE Transactions on Cybernetics*, 2(46):344–355, 2015.

Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2008.

Wei Wang, Dejan Slepčev, Saurav Basu, John A Ozolek, and Gustavo K Rohde. A linear optimal transportation framework for quantifying and visualizing variations in sets of images. *International journal of computer vision*, 101(2):254–269, 2013.

Michael Westdickenberg. Projections onto the cone of optimal transport maps and compressible fluid flows. *Journal of Hyperbolic Differential Equations*, 7(04):605–649, 2010.

S. C. Wieland, J. S. Brownstein, B. Berger, and K. D. Mandl. Density-equalizing Euclidean minimum spanning trees for the detection of all disease cluster shapes. *Proceedings of the National Academy of Sciences of the United States of America*, 104(22): 9404–9409, 2007.

Wikipedia. List of Chinese administrative divisions by GDP. https://en.wikipedia.org/wiki/List_of_Chinese_administrative_divisions_by_GDP, 2017. Accessed 3 May 2017.

Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K Nayar. Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *IEEE Transactions on Image Processing*, 19(9):2241–2253, 2010.