POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Clustering probability measures in the Wasserstein space

Tesi di Laurea Triennale in
Ingegneria Matematica

Authors:
Pietro Masini
Maria Chiara Menicucci

Student IDs: 984017 983132
Advisor: Prof. Mario Beraha

Academic Year: 2023-24

Abstract

In this thesis we tackle the problem of clustering probability measures in the Wasserstein space. After presenting the necessary background in optimal transport and defining the Wasserstein distance, we introduce the problem of clustering and review two major algorithms used in clustering Euclidean data: K-Means and Expectation-Maximization. Then we show how they can be modified for the purpose of clustering probability measures, focusing on proposing a novel algorithm which extends the EM algorithm to the case of measures. We implement such extensions in Python and compare their performances on simulated datasets, ascertaining that EM performs better than K-Means.

Keywords: clustering, Wasserstein space, K-Means algorithm, Expectation-Maximization algorithm

Abstract in lingua italiana

In questa tesi affrontiamo il problema del clustering di misure di probabilità nello spazio di Wasserstein. Dopo aver presentato il necessario background di trasporto ottimo e aver definito la distanza di Wasserstein, presentiamo il problema del clustering ed esaminiamo due importanti algoritmi usati per fare clustering di dati euclidei: l'algoritmo K-Means e l'algoritmo Expectation-Maximization. Quindi mostriamo come possono essere modificati allo scopo di fare clustering di misure di probabilità, concentrandoci sul proporre un nuovo algoritmo che estenda l'algoritmo EM al caso delle misure. Implementiamo tali estensioni in Python e confrontiamo le loro prestazioni su dati simulati, constatando che l'algoritmo EM ha risultati migliori di K-Means.

Parole chiave: clustering, spazio di Wasserstein, algoritmo K-Means, algoritmo Expectation-Maximization

Contents

Introduction

Clustering data is a fundamental problem in the field of unsupervised machine learning aimed at discovering inherent structures within a dataset: the purpose of clustering is grouping similar data points (which are unlabeled, unlike supervised learning) that are similar, according to an established definition of similarity.

This problem has been described with a large variety of models, which have led to many different algorithms. We quickly review the main clustering models. For a more complete survey of clustering see e. g. [22].

Centroid models represent each cluster with a mean vector, considered the "center of the cluster". The popular K-Means algorithm, proposed by [10] and largely used in machine learning, arises from centroid models: given the desired number K of clusters, it partitions the data in K clusters in order to minimize the variance within each cluster. Such algorithm uses the Euclidean distance, so K-Means turns out not to work well when the data are grouped in clusters whose shape does not resemble that of Euclidean neighbourhoods, such as elliptic data. A few drawbacks of K-Means are highlighted in [20].

Connectivity models lead to algorithms such as hierarchical clustering, which do not require to specify in advance the number of clusters, as they seek to build a hierarchy of clusters. The key idea is that a cluster is identified by the maximum distance needed to connect parts of the cluster: different clusters will arise at different distances. These algorithms indeed provide a hierarchy of clusters which merge with each other at certain distances. A possible reference for hierarchical clustering is [7].

Density models identify dense regions, seeking for points which have many nearby neighbors, and assign the data points within these regions to the same cluster. DBSCAN, proposed by [5], and OPTICS algorithm, developed by the same authors a few years later, are based on such models.
Distribution models, which will play a big role in this thesis, view the data as realizations of random variables with a certain distribution. For instance, Gaussian mixture models (GMM) assume that the data are realizations of a mixture of multivariate normal distributions with parameters (μ_i, Σ_i). The expectation-maximization algorithm exploits such models to cluster data (a possible reference is [3]). This algorithm is a powerful generalization of K-Means, as the latter is recovered when Σ_i is the identity matrix for all i. But the higher generality of Σ_i allows to account for much more general cluster forms (such as elliptic) than circular ones.

However, data's complexity is increasing, leading to the need to cluster less straightforward types of data as well, such as probability measures. In such a context, to group together similar data, a definition of distance in the space of probability measures is needed. This is provided by optimal transport, a field of mathematical analysis dealing with the efficient transportation of goods or resources from one location to another, taking into account both the supply and demand constraints as well as the associated transportation costs. It is a flourishing field, for which many different references are available, such as [18], [1] and [21].
Optimal transport allows to define a metric, called the Wasserstein distance, in the space $\mathbb{W}_p(\mathbb{R})$, a space of probability measures with certain features: informally, the distance between two probability measures is defined as the minimum cost required to transform one probability distribution into another by matching their mass according to certain transportation costs.
Computing such distance in $\mathbb{W}_p(\mathbb{R})$ is a pretty straightforward task when $p = 2$, thanks to the isometric isomorphism between $\mathbb{W}_2(\mathbb{R})$, endowed with

the Wasserstein distance, and the space of quantile functions, endowed with the usual L^2 norm.
Although this may look as a rather abstract framework, the Wasserstein setting is witnessing a significant popularity for its applications to statistics and data science ([15] has been our main reference for this matter). Indeed, given the Wasserstein distance, applying the K-Means algorithm to cluster probability measures in $\mathbb{W}_p(\mathbb{R})$ is immediate: this has been done in [23], and examples of its usage can be found, for instance, in [17] or [11].

Yet - just like in the Euclidean setting - this algorithm does not perform well on some kinds of data. As the Expectation-Maximization solves this issue in the Euclidean case, it is natural to try to extend it, as well, to the Wasserstein setting.
In Chapter 3 we develop and discuss a novel algorithm which extends Expectation-Maximization to the Wasserstein space. The extension is less straightforward than the K-Means algorithm and requires some careful examination of the statistical meaning of various terms. However, this algorithm yields indeed better results than K-Means, showing a correct clustering of data (i. e. consistent with eye-obvious clusters) and solving the problem of clustering probability measures at least for measures in $\mathbb{W}_2(\mathbb{R})$.

1 | Background on optimal transport

1.1. Background on optimal transport

We quickly review the basics of optimal transport theory, following [15].

Optimal transport theory arises from Monge's following question: given a pile of sand and a pit of equal volume, how to transport the sand into the pit (i.e. choose the destination of each unit of sand) so as to minimize the cost of the transportation? The problem has been formalized by Monge in [13] as follows.

Let $\mathbb{X}$ be the space of sand, $\mathbb{Y}$ the pit space; we shall assume they are both complete and separable metric spaces, endowed with the topology induced by their metric and the σ-algebras generated by their topologies (which we will denote by $\sigma_{\mathbb{X}}, \sigma_{\mathbb{Y}}$ respectively).
Let $c : \mathbb{X} \times \mathbb{Y} \to [0, +\infty)$ be a cost function, such that $c(x, y)$ represents the cost of transporting a unit of sand lying in $x \in \mathbb{X}$ to a location $y \in \mathbb{Y}$ in the pit; we assume that c is measurable with respect to the product σ-algebra $\sigma_{\mathbb{X}} \otimes \sigma_{\mathbb{Y}}$. Let the sand distribution in $\mathbb{X}$ be represented by a finite measure μ on $\mathbb{X}$, and the pit shape described by a finite measure ν on $\mathbb{Y}$; without loss of generality, we will henceforth assume that μ, ν are probability measures.

Monge problem Establishing how to transport the sand, i.e. choosing the destination y of every unit of sand x, boils down to identifying a measurable function $T : \mathbb{X} \to \mathbb{Y}$, hence said transport map. The transport cost associated

with T, denoted by $C(T)$, is

$$C(T) = \int_{\mathbb{X}} c(x, T(x))\, d\mu(x)$$

Such integral is well defined (it may also be $C(T) = +\infty$) thanks to the measurability assumption on c and T and the nonnegativity of c.

Monge seeks to solve this problem under the constraint that for any set $B \in \sigma_{\mathbb{Y}}$ the sand allocated in B, $T^{-1}(B)$, has a volume equal to the amount of space in B. That is to say, T must satisfy

$$\mu\left(T^{-1}(B)\right) = \nu(B) \;\; \forall\; B \in \sigma_{\mathbb{Y}} \tag{1.1}$$

Definition Let $\mathbb{X}$, $\mathbb{Y}$ be complete and separable metric spaces[1], $T : \mathbb{X} \to \mathbb{Y}$ a measurable function, μ a probability measure on $\mathbb{X}$, ν a probability measure on $\mathbb{Y}$. We say that T pushes μ forward to ν, and we write $T\#\mu = \nu$, if for each $B \in \sigma_{\mathbb{Y}}$ $\mu\left(T^{-1}(B)\right) = \nu(B)$.

As a consequence, if $T : \mathbb{X} \to \mathbb{Y}$, $U : \mathbb{Z} \to \mathbb{X}$ are measurable functions and μ is a probability measure on $\mathbb{Z}$, $(T(U))\#\mu = T\#(U\#\mu)$. We will need this later.

Monge problem is finding the optimal transport map satisfying (1.1), i. e. solving

$$\inf_{T:T\#\mu=\nu} C(T)$$

It natural to assume that the cost function is continuous and nonnegative and, when $\mathbb{X} = \mathbb{Y}$ is endowed with the metric d, that c is increasing with the distance $d(x, y)$, e. g. $c(x, y) = (d(x, y))^p$, $p \geq 0$.

Monge problem, as formulated above, may be ill-posed. Indeed, for some

[1]The reader might ask what is the purpose of such hypothesis. They are a standard assumption since they are needed to prove the existence of a solution to the Kantorovich problem (which will be presented later); such result will not be used in this work, since we are interested to the much less general case of $\mathbb{X} = \mathbb{Y} = \mathbb{R}$.

choices of μ and ν, the set of T satisfying (1.1) may be empty. This happens when e. g. μ is a Dirac measure at $x_0 \in \mathbb{X}$ and ν is not a Dirac measure: the Monge formulation does not allow to "split the sand". In such a case, given $\mu\left(T^{-1}(B)\right) = \nu(B)$, $\mu\left(T^{-1}(B)\right)$ may as well be written as the indicator function $I_{T^{-1}(B)}(x_0) = I_B(T(x_0))$ as well, and we cannot have $I_B(T(x_0)) = \nu(B)$ for any B if ν is not Dirac at $T(x_0)$.
Vice versa, if ν is Dirac at some $y_0 \in \mathbb{Y}$, $\mu\left(T^{-1}(B)\right) = I_B(y_0)$ always yields the solution $T(x) = y_0$ (which means all sand is transported at y_0): μ and ν have an asymmetric role in Monge's formulation.
The essential reason for these issues is the fact that the sand transportation is governed by a function $T : \mathbb{X} \to \mathbb{Y}$, which necessarily attaches a unique destination y to each x: so, if we have all mass at x_0, we cannot but move all mass to an only point $T(x_0)$.

Kantorovich problem Kantorovich, in [9], solves such issues relaxing Monge's formulation: this ends up in not having a deterministic function $x \mapsto T(x)$, but a probability measure γ_x which expresses how the mass in x is split among different destinations. If γ_x is Dirac at some y, all mass in x is sent to a unique destination, and we get back to the previous setting (in which to each x we associate a unique $y = T(x)$).

This is formalized using a bivariate probability measure π on the space[2] $\mathbb{X}\times\mathbb{Y}$ where $\pi(A \times B)$ represents the amount of sand transported from $A \in \sigma_{\mathbb{X}}$ to $B \in \sigma_{\mathbb{Y}}$. The constraints that naturally follow from mass conservation are that the total mass sent from A $\pi(A \times \mathbb{Y})$ is equal to the mass present in A $\mu(A)$, and the total mass sent in B $\pi(\mathbb{X} \times B)$ is equal to the space in B $\nu(B)$, that is to say

$$\begin{aligned} \pi(A \times \mathbb{Y}) &= \mu(A) \ \forall\ A \in \sigma_{\mathbb{X}} \\ \pi(\mathbb{X} \times B) &= \nu(B) \ \forall\ B \in \sigma_{\mathbb{Y}} \end{aligned}$$

[2]The product space $\mathbb{X} \times \mathbb{Y}$ is, too, a complete and separable (with respect to the product topology) metric space, endowed with the σ-algebra generated by its topology.

This couple of equations describes Kantorovich constraints.

π satisfying such constraints is called transference plan; the set of such measures will be denoted with $\Pi(\mu, \nu)$. Each $\pi \in \Pi(\mu, \nu)$ is a coupling of μ and ν, which are said to be marginal distributions of π. The cost associated with the transference plan π is

$$C(\pi) = \int_{\mathbb{X}\times\mathbb{Y}} c(x, y)\, d\pi(x, y)$$

Kantorovich problem is finding the optimal transport map, i. e. solving

$$\inf_{\pi\in\Pi(\mu,\nu)} C(\pi)$$

This formulation brings some advantages, e. g. $\Pi(\mu, \nu)$ is never empty: the product measure $\mu \otimes \nu$ always satisfies Kantorovich constraints.

Both formulations have a natural probabilistic interpretation that, though not directly useful to our needs, we believe to be enlightening. Let $X : (\Omega, \mathcal{A}, \mathbf{P}) \to \mathbb{X}, Y : (\Omega, \mathcal{A}, \mathbf{P}) \to \mathbb{Y}$ be random variables with probability distributions μ, ν respectively. The Monge problem is to find T such that the law of the random variable $T(X)$ is ν and such that $C(T) = \int_{\mathbb{X}} c(x, T(x))\, d\mu(x)$, i.e. the expectation

$$C(T) = \int_{\Omega} c(X(\omega), T(X(\omega)))\, d\mathbf{P}(\omega) = \mathbf{E}(c(X, T(X)))$$

is minimised.

The Kantorovich problem, in the same setting, attempts to find a joint distribution for (X, Y), denoted with π, having marginals μ and ν; γ_x is the conditional distribution of Y given that $X = x$. We can see $C(\pi)$ as $\mathbf{E}(c(X, Y))$: then we wish to find π such that $C(\pi) = \int_{\mathbb{X}\times\mathbb{Y}} c(x, y)\, d\pi(x, y)$, i. e. the

expectation

$$C(\pi) = \int_{\Omega} c(X(\omega), Y(\omega)) \, d\mathbf{P}(\omega) = \mathbf{E}(c(X, Y))$$

is minimised.

On the other hand, according to the disintegration theorem proved in probability textbooks such as [8], since $\mathbb{X}$ is a complete and separable metric space, π, given μ, can be represented by the set of measures $\{\gamma_x\}_{x \in \mathbb{X}}$ on $\mathbb{Y}$, and the equation $C(\pi) = \mathbf{E}(c(X, Y))$ yields

$$C(\pi) = \int_{\mathbb{X}} \left(\int_{\mathbb{Y}} c(x, y) \, d\gamma_x(y) \right) d\mu(x)$$

This gives a confirmation of what we said in the beginning about γ_x being Dirac at some y: if γ_x is Dirac at $T(x)$ (with T such that $T\#\mu = \nu$), we get $C(\pi) = \int_{\mathbb{X}} \left(\int_{\mathbb{Y}} c(x, y) \, d\gamma_x(y) \right) d\mu(x) = \int_{\mathbb{X}} c(x, T(x)) \, d\mu(x) = C(T)$.[3]

1.2. The Wasserstein space

We denote by $P(\mathbb{X})$ the set of probability measures on $\mathbb{X}$.

Definition Let $(\mathbb{X}, ||\cdot||)$ be a separable Banach space. Given $p \geq 1$, we call p-Wasserstein space on $\mathbb{X}$ the set $\mathbb{W}_p(\mathbb{X}) = \left\{ \mu \in P(\mathbb{X}) : \int_{\mathbb{X}} ||x||^p \, d\mu(x) \in \mathbb{R} \right\}$.

$\mathbb{W}_p(\mathbb{X})$ is a subspace of $P(\mathbb{X})$ formed by probability measures with finite p-th momentum. If we think of μ as the law of a random variable X taking values in $\mathbb{X}$, $\mathbb{W}_p(\mathbb{X})$ is the subspace of $P(\mathbb{X})$ formed by laws that make X a random variable in L^p.

The optimal transportation concepts developed above provide a natural way of defining a distance between two probability measures in $\mathbb{W}_p(\mathbb{X})$.

[3] Note that choosing μ_x Dirac at $T(x)$ is legit because it identifies a transference plan $\pi \in \Pi(\mu, \nu)$ satisfying Kantorovich constraints: indeed, $\pi(A \times B) = \mu\left(A \cap T^{-1}(B)\right)$, thus $\pi(\mathbb{X} \times B) = \mu\left(T^{-1}(B)\right) = \nu(B)$ and $\pi(A \times \mathbb{Y}) = \mu\left(A \cap T^{-1}(\mathbb{Y})\right) = \mu(A)$.

Definition Given $\mu, \nu \in \mathbb{W}_p(\mathbb{X})$, we call Wasserstein distance between μ and ν $W_p(\mu,\nu) = \left(\inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathbb{X}^2} ||x-y||^p \, d\pi(x,y)\right)^{\frac{1}{p}}$.

$W_p(\mu,\nu)$ is the minimal transportation cost between μ and ν according to Kantorovich formulation of optimal transport problem, where the chosen cost function is $c(x,y) = ||x-y||^p$. The integral is well defined when $\mu, \nu \in \mathbb{W}_p(\mathbb{X})$ thanks to the inequality $||x-y||^p \leq 2^p ||x||^p + 2^p ||y||^p$.

It can be proved that $\mathbb{W}_p(\mathbb{X})$, endowed with such distance, is a metric space (see e. g. [18]); to do so, we need a lemma. We denote by μ_X the law of a random variable taking values in $\mathbb{X}$ and by $\mu_{(X_1,X_2)}$ the law of a random variable taking values in $\mathbb{X}^2$.

Proposition 1.1. ***Gluing lemma***
Let $\mathbb{X}$ be a polish space and $(Y_1,Y_2), (Z_1,Z_2)$ be random vectors taking values in $\mathbb{X}^2$ such that $\mu_{Y_2} = \mu_{Z_1}$. Then there exists a random vector (X_1,X_2,X_3) such that $\mu_{(X_1,X_2)} = \mu_{(Y_1,Y_2)}$ and $\mu_{(X_2,X_3)} = \mu_{(Z_1,Z_2)}$.

Proposition 1.2. *W_p is a distance on $\mathbb{W}_p(\mathbb{X})$.*

Proof W_p is well defined and obviously nonnegative. Symmetry holds because $W_p(\mu,\nu) = \left(\inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathbb{X}^2} ||x-y||^p \, d\pi(x,y)\right)^{\frac{1}{p}}$ equals $W_p(\nu,\mu) = \left(\inf_{\tilde{\pi}\in\Pi(\nu,\mu)} \int_{\mathbb{X}^2} ||x-y||^p \, d\tilde{\pi}(y,x)\right)^{\frac{1}{p}}$ as we remarked above: if $\pi \in \Pi(\mu,\nu)$, $\tilde{\pi}(B\times A) := \pi(A\times B)$ is such that $\tilde{\pi} \in \Pi(\nu,\mu)$; since $\tilde{c}(y,x) := c(x,y) = ||x-y||^p$, then $C(\pi) = \int_{\mathbb{X}^2} c(x,y)\, d\pi(x,y) = \int_{\mathbb{X}^2} \tilde{c}(y,x)\, d\tilde{\pi}(y,x) = \tilde{C}(\tilde{\pi})$ for each $\pi \in \Pi(\mu,\nu)$, $\tilde{\pi} \in \Pi(\nu,\mu)$.

The triangle inequality is the non-trivial part. We want to show that $W_p(\mu,\nu) \leq W_p(\mu,\gamma) + W_p(\gamma,\nu)$ for each μ,ν,γ. Suppose $(Y_1,Y_2), (Z_1,Z_2)$ are random vectors such that their laws (π^*, π^{**} respectively) are optimal couplings of μ,γ and γ,ν respectively, so that $\mu_{Y_2} = \mu_{Z_1} = \gamma$, $\mu_{Y_1} = \mu$, $\mu_{Z_2} = \nu$. We denote with $\tilde{\pi}$ probability measures on $\mathbb{X}^2$ which couple μ,γ and $\bar{\pi}$ probability measures on $\mathbb{X}^2$ which couple γ,ν.

The gluing lemma states that there exists a random vector (X_1, X_2, X_3) such that $\mu_{(X_1,X_2)} = \pi^*$ and $\mu_{(X_2,X_3)} = \pi^{**}$. (X_1, X_3) couples μ, ν: its law is $\pi \in \Pi(\mu, \nu)$ (we do not know whether it is optimal). Hence $W_p(\mu, \nu) \leq \left(\int_{\mathbb{X}^2} ||x - z||^p \, d\pi(x, z)\right)^{\frac{1}{p}} = \left(\int_{\mathbb{X}^2} ||x - z||^p \, d\mu_{\mathbf{X}}(x, z)\right)^{\frac{1}{p}} = ||x - z||_{L^p(\mathbb{X}, \sigma_{\mathbb{X}}, \mu_{\mathbf{X}})} \leq ||x - y||_{L^p(\mu_{\mathbf{X}})} + ||y - z||_{L^p(\mu_{\mathbf{X}})} = \left(\int_{\mathbb{X}^2} ||x - y||^p \, d\pi^*(x, y)\right)^{\frac{1}{p}} + \left(\int_{\mathbb{X}^2} ||y - z||^p \, d\pi^{**}(y, z)\right)^{\frac{1}{p}} = W_p(\mu, \gamma) + W_p(\gamma, \nu)$. ■

Note that, if $\nu = \delta_0$, ν is also the conditional distribution of Y given that $X = x$, thus $\int_{\mathbb{X}^2} ||x - y||^p \, d\pi(x, y) = \int_{\mathbb{X}} \left(\int_{\mathbb{X}} ||x - y||^p \, d\nu(y)\right) d\mu(x) = \int_{\mathbb{X}} ||x||^p \, d\mu(x)$: then $W_p(\mu, \delta_0) = \int_{\mathbb{X}} ||x||^p \, d\mu(x)$ and $\mathbb{W}_p(\mathbb{X})$ may also be defined as the set of measures μ such that $W_p(\mu, \delta_0) < +\infty$.

1.3. The one-dimensional case

The case $\mathbb{X} = \mathbb{Y} = \mathbb{R}$ is all we will need in our work.
We will denote by $P(\mathbb{R})$ the set of probability measures on $\mathbb{R}$. Given a probability measure $\mu \in P(\mathbb{R})$, F will denote its cumulative distribution function, defined as $F : \mathbb{R} \to [0, 1], F(x) := \mu((-\infty, x])$. It is well known that F is non-decreasing and right-continuous. Let us give a definition which generalizes the notion of inverse; we follow [4].

Definition Given a cumulative distribution function $F : \mathbb{R} \to [0, 1]$, we call pseudo-inverse of F, and we denote it by F^{-1}, the function $F^{-1} : (0, 1) \to \mathbb{R}, F^{-1}(u) = \inf\{x \in \mathbb{R} : F(x) \geq u\}$.[4]

This is a generalization of the usual notion of inverse, which the above definition reduces to when F is bijective. F is right-continuous, hence F^{-1} is also monotonically increasing left-continuous (therefore sometimes said left-continuous inverse).

F^{-1} is also said quantile function of the random variable having cumulative

[4] F^{-1} is well defined if $u \in (0, 1)$, because F is a cumulative distribution function, thus for any $u \in (0, 1)$ $\{x \in \mathbb{R} : F(x) \geq u\} = [a, +\infty))$; if $u = 0$, $F^{-1}(u) = -\infty$; if $u = 1$, $F^{-1}(u)$ may be empty.

distribution function F. For each $u \in (0,1)$ $F\left(F^{-1}(u)\right) \geq u$, and the equality holds if and only if $u \in Im(F)$; for each $x \in \mathbb{R}$ $F^{-1}(F(x)) \leq x$, and the equality holds if and only if $x \in Im\left(F^{-1}\right)$.

The set of left-continuous, non-decreasing functions defined on $(0,1)$ and taking values in $\mathbb{R}$ will be denoted with $L_2^{\uparrow}([0,1])$.

In order to face one of the few cases in which we have a unique and explicit solution to Monge problem, we need a preliminary result whose proof is taken by [18], lemma 2.4.

Proposition 1.3. *Let F be the cumulative distribution function of $\mu \in P(\mathbb{R})$. If F is absolutely continuous, then $F\#\mu = Leb_{(0,1)}$.*

Proof Thanks to the uniqueness of extension of real measures defined on a π-system generating $\mathcal{B}(\mathbb{R})$, it is enough to test the equality of the measures $\mu\left(F^{-1}(B)\right)$ and $Leb_{(0,1)}(B)$ on intervals $B = (-\infty, u]$, with $u \in (0,1)$.

F is absolutely continuous, then the set $F^{-1}((-\infty, u])$ has the form $(-\infty, a]$, where a is such that $F(a) = u$: as a consequence, $\mu\left(F^{-1}(B)\right) = \mu((-\infty, a])$ $= F(a) = u$; on the other hand, $Leb_{(0,1)}((-\infty, u]) = u$, so we have the thesis.

If $u < 0$ or $u > 1$, both measures are respectively zero and one. If $u = 0$, we have $F^{-1}(B) = F^{-1}(\{0\})$: such set is empty (then both measures are zero) or it has the form $(-\infty, a]$ with $F(a) = 0$, hence the same arguments above hold. If $u = 1$, $F^{-1}(B) = F^{-1}((-\infty, 1])$: such set is $\mathbb{R}$, then both measures take the value 1. ■

Let us suppose $c(x,y) = (x-y)^2$ (which is indeed of the previewed form proportional to a distance): in such a case it is reasonable to assume T monotonically non-decreasing.

Theorem 1.1. ***Solution of Monge problem in one-dimensional case*** *Let F, G be the cumulative distribution functions of $\mu, \nu \in P(\mathbb{R})$ respectively. If F is absolutely continuous, then $T = G^{-1} \circ F$ is the unique, up to a μ-null set, non-decreasing map $T : \mathbb{R} \to \mathbb{R}$ such that $T\#\mu = \nu$.*

Then the set of T satisfying $T\#\mu = \nu$ contains at most one non-decreasing map. We follow the proof of [18], Theorem 2.5.

Proof We show that $T(x) = G^{-1}(F(x))$ is a non-decreasing map satisfying $T\#\mu = \nu$.

T is well defined if $F(x) \in (0,1)$, i. e. μ-almost everywhere[5]. T is non-decreasing as it is the composition of two non-decreasing functions.

Moreover, $T\#\mu = \nu$ because $\left(G^{-1}(F)\right)\#\mu = G^{-1}\#(F\#\mu) = G^{-1}\#Leb_{(0,1)}$. We test the equality of the measures $G^{-1}\#Leb_{(0,1)}$ and ν on intervals $B = (-\infty, x]$ with $x \in (0,1)$.
$\left(G^{-1}\#Leb_{(0,1)}\right)(B) = Leb_{(0,1)}\left(\left\{u : G^{-1}(u) \leq x\right\}\right)$. $\left\{u : G^{-1}(u) \leq x\right\} = \{u : G(x) \geq u\}$ because $G\left(G^{-1}(u)\right) \geq u$ e $G^{-1}(G(x)) \leq x$; but $\{u : G(x) \geq u\}$ has the form $[0, a]$ with a such that $G(x) = a$. Hence $\left(G^{-1}\#Leb_{(0,1)}\right)(B) = G(x) = \nu((-\infty, u]) = \nu(B)$.

We prove the uniqueness. Let T be a non-decreasing map such that $T\#\mu = \nu$; let $x \in \mathbb{R}$ be fixed. Since T is non-decreasing, $T^{-1}((-\infty, T(x)]) \supseteq (-\infty, x]$: then $\mu\left(T^{-1}((-\infty, T(x)])\right) = \nu((-\infty, T(x)]) \geq \mu((-\infty, x]) = F(x)$, that is $F(x) \leq G(T(x))$. Taking G^{-1} at both sides we get $G^{-1}(F(x)) \leq T(x)$. Suppose the inequality is strict: then there exists $\varepsilon_0 > 0 : F(x) \leq G(T(x) - \varepsilon)\ \forall\ \varepsilon \in (0, \varepsilon_0)$.
On the other hand, $T^{-1}((-\infty, T(x) - \varepsilon]) \subseteq (-\infty, x)$, then (taking the measure of both sides) $G(T(x) - \varepsilon) \leq F(x)$ and the equality $F(x) = G(T(x) - \varepsilon)$ holds $\forall\ \varepsilon \in (0, \varepsilon_0)$. This implies that G takes the costant

[5] $\{x : F(x) = 1\}$ either is empty (and its measure is zero) or it has the form $[a, +\infty)$ with $F(a) = 1$, hence $\mu(\{x : F(x) = 1\}) = \mu([a, +\infty)) = 1 - F(a) = 0$, since F is absolutely continuous.
An analogous argument holds for $\{x : F(x) = 0\}$.

value $F(x)$ on an interval (as $T(x)-\varepsilon$ varies in $(T(x)-\varepsilon_0, T(x))$): such intervals are a countable quantity, hence the set of values of F on such intervals is countable: we call such values l_i. Thus the set of $x : G^{-1}(F(x)) < T(x)$ is contained in the set of $\bigcup_i \{x : F(x) = l_i\}$, which has measure 0 under μ. Then $G^{-1}(F(x)) = T(x)$ up to a μ-null set. ■

Suppose the hypotheses of Theorem 1.1 are satisfied. Given $T(x) = G^{-1}(F(x))$ and $c(x,y) = (x-y)^2$, we find

$$C(T) = \int_{\mathbb{R}} \left(x - G^{-1}(F(x))\right)^2 d\mu(x) = \int_0^1 \left(F^{-1}(u) - G^{-1}(u)\right)^2 du$$

using the substitution $F(x) = u$. The next theorem (see theorem 1.5.1 in [15]) states - for a more general form of $c(x,y)$, and even when F is not absolutely continuous - that $\int_0^1 \left(F^{-1}(u) - G^{-1}(u)\right)^2 du$ is the optimal transport cost.

Theorem 1.2. ***Solution of Kantorovich problem in one-dimensional case*** *Let F, G be the cumulative distribution functions of $\mu, \nu \in P(\mathbb{R})$ respectively. If $h : \mathbb{R} \to [0, +\infty)$ is a convex function and $c(x,y) = h(|x-y|)$, then*

$$\inf_{\pi \in \Pi(\mu,\nu)} C(\pi) = \int_0^1 h\left(G^{-1}(u) - F^{-1}(u)\right) du$$

Furthermore, if F is absolutely continuous, such infimum is attained by the transport map $T = G^{-1} \circ F$.

Note that, if we do not ask F to be absolutely continuous, the set of maps satisfying Monge constraints may be empty.

A common choice of h is $h(z) = |z|^p$.

In the special case $\mathbb{X} = \mathbb{R}$ with the absolute value norm and $p = 2$ - which will be our main focus -, given $\mu, \nu \in P(\mathbb{R})$ with cumulative distribution functions F, G respectively, the definition of the Wasserstein distance reads $W_2(\mu,\nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^2} (x-y)^2 d\pi(x,y)\right)^{\frac{1}{2}} = \int_0^1 \left(F^{-1}(u) - G^{-1}(u)\right)^2 du$,

thanks to the theorem stated above. Thus the following result holds.

Theorem 1.3. *The map attaching to each $\mu \in W_2(\mathbb{R})$ its quantile function F^{-1} is an isometric isomorphism.*

Proof With change of variables, it is easy to show that $\mu \in \mathbb{W}_2(\mathbb{R})$ if and only if its pseudo-inverse belongs to $L^2(0,1)$. The map which assigns to each probability measure in $\mathbb{W}_2(\mathbb{R})$ its pseudo-inverse is bijective. We showed in the proof of 1.1 that, given $F^{-1} \in L^2_{\uparrow}([0,1])$ (i. e. a L^2 non-decreasing, left-continuous function defined on $(0,1)$), $F^{-1}\#Leb_{(0,1)}$ is a probability measure whose cumulative distribution function is F, therefore the above map is surjective. On the other hand, two different probability measures have two different cumulative distribution functions and so two different inverses: hence the above map is injective. The equality $W_2(\mu,\nu) = \left(\inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathbb{R}^2} (x-y)^2 \, d\pi(x,y)\right)^{\frac{1}{2}} = \int_0^1 \left(F^{-1}(u) - G^{-1}(u)\right)^2 du$, stated in Theorem 1.2 above, yields the isometry. ■

This is the main result from optimal transport theory we will need:

$$W_2(\mu,\nu) = \int_0^1 \left(F^{-1}(u) - G^{-1}(u)\right)^2 du = \left|\left|F^{-1} - G^{-1}\right|\right|_{L^2(0,1)} \tag{1.2}$$

1.4. Statistics in the Wasserstein space

We have to do a last step in the Wasserstein world before we can apply this knowledge to clustering: since we will need to see probability measures as our data, we have to define for them some statistical quantities and introduce special random variables called random measures (we follow both [15] and [16]).

We give up on higher generality and focus on $\mathbb{W}_2(\mathbb{R})$.

Definition The Fréchet functional associated with measures $\mu_1, ..., \mu_n \in \mathbb{W}_2(\mathbb{R})$ is $F : \mathbb{W}_2(\mathbb{R}) \to \mathbb{R}$, $F(\gamma) = \frac{1}{2n}\sum_{i=1}^n W_2^2(\gamma, \mu_i)$. A minimiser of F in

$W_2(\mathbb{R})$ is said to be a Fréchet mean of $(\mu_1, ..., \mu_n)$.

Actually, thanks to Theorem 1.3 it is possibile to prove that there exists a unique Fréchet mean of $\mu_1, ..., \mu_n \in \mathbb{W}_2(\mathbb{R})$: it is the measure μ having quantile function $F_\mu^{-1} = \frac{1}{n}\sum_{i=1}^n F_{\mu_i}^{-1}$.

Let us now give a definition of random measures and extend to random measures notions well known for real random variables, such as cumulative distribution functions, expected value and variance.

Definition Given a probability space $(\Omega, \mathcal{A}, \mathbf{P})$, we call random measure on $\mathbb{W}_2(\mathbb{R})$ any measurable mapping $\mathfrak{F} : \Omega \to \mathbb{W}_2(\mathbb{R})$, where $\mathbb{W}_2(\mathbb{R})$ is endowed with its Borel σ-algebra[6].

From an intuitive point of view, $\mathfrak{F}$ is a random variable taking as values probability measures in $\mathbb{W}_2(\mathbb{R})$.

If $\mathfrak{F} : \Omega \to \mathbb{W}_2(\mathbb{R})$ is a random measure, its cumulative distribution function $\mathbb{F}$ can be defined in the usual way, but it is now a random variable taking values on the set of real cumulative distribution functions: $\mathbb{F}[\omega](t) := \mathfrak{F}[\omega]((-\infty, t])$.

We will work mainly with the cumulative distribution and quantile function of $\gamma_m(\mathfrak{F})$, which will be denoted respectively by Γ_m and Γ_m^{-1}.

Definition Given a random measure $\mathfrak{F} : \Omega \to \mathbb{W}_2(\mathbb{R})$, we call Wasserstein-Fréchet mean of $\mathfrak{F}$ the probability measure $\gamma_m(\mathfrak{F}) = \arg\min_{\gamma \in \mathbb{W}_2(\mathbb{R})} \frac{1}{2}\mathbf{E}\left(W_2^2(\mathfrak{F}, \gamma)\right)$; in such a case, we call Wasserstein-Fréchet variance of μ the real number $var(\mathfrak{F}) = \mathbf{E}\left(W_2^2(\mathfrak{F}, \gamma_m)\right)$.

Note that $W_2^2(\mathfrak{F}, \gamma)$, once γ is fixed, is a real random variable, and $\mathbf{E}\left(W_2^2(\mathfrak{F}, \gamma)\right)$ can be seen as a functional $T(\gamma)$ on $\mathbb{W}_2(\mathbb{R})$.

It can be proved that, for each random measure $\mathfrak{F} : \Omega \to \mathbb{W}_2(\mathbb{R})$ with finite Fréchet functional, the Fréchet mean exists and it is unique: its quantile

[6]We are considering the topology inducted on $\mathbb{W}_2(\mathbb{R})$ by the Wasserstein distance.

function is $\Gamma_m^{-1}(t) = \mathbf{E}\left(\mathbb{F}^{-1}[\omega](t)\right)$.[7]

Let us remember that, thanks to the isometric isomorphism,

$$\gamma_m(\mathfrak{F}) = \arg\min_{\gamma\in\mathbb{W}_2(\mathbb{R})} \frac{1}{2}\mathbf{E}\left(W_2^2(\mathfrak{F},\gamma)\right) = \arg\min_{\gamma\in\mathbb{W}_2(\mathbb{R})} \frac{1}{2}\mathbf{E}\left(\int_0^1 \left(\mathbb{F}^{-1}[\omega](u) - \Gamma^{-1}(u)\right)^2 du\right)$$

while

$$\mathfrak{S}^2 = \mathbf{E}\left(\int_0^1 \left(\mathbb{F}^{-1}[\omega](t) - \Gamma_m^{-1}(t)\right)^2 dt\right)$$

[7] See [15] for a proof.

The issue of existence and uniqueness would be much trickier if we considered random measures on $\mathbb{W}_2(\mathbb{X})$. For uniqueness, for instance, absolute continuity of the random measure would be required. See Appendix A for a definition of absolutely continuous measures.

2 | Clustering Euclidean data

The term *clustering* refers to the following task: given a set of data[1], we want to partition them in homogeneous subgroups, called clusters, so that similar data are grouped together. Of course, the meaning of *similar* should be better precised, which we will do later on.

A motivation for clustering is obvious: on one side it reduces dataset complexity while preserving essential features, on the other side it highlights similarities in data. For instance, companies having a large dataset of customer information may want to use such data to understand their customers better, in order to tailor marketing strategies. With thousands of customers, it is difficult to analyze individual behaviors and patterns: however, using the available data it is possible to cluster customers. Instead of analyzing 1000 individual customers, the company can analyze, for example, 4 customer clusters, without losing significant information. This significantly reduces the complexity of data.

We will first tackle the Euclidean case, where data are vectors in $\mathbb{R}^n$: this case is a good start to review the basic algorithms. Then we will move to the case of probability measures, which is the subject of our thesis and builds on the Euclidean case.

[1] commonly, vectors in $\mathbb{R}^n$; our thesis focuses on the case of data being probability measures

2.1. K-Means clustering

A first model for clustering may be the following.
Let the set of given observations be $X = \{x_1, ..., x_n\}$, where x_i is an element of a metric space (A, d) for $i = 1, ..., n$. We want to partition X, i. e. identify a set of clusters $S = \{S_1, ..., S_k\}$, $k \leq n$, such that $\bigcup_{i=1}^{k} S_i = X$, $S_i \cap S_j = \varnothing$, $i, j = 1, ..., k, i \neq j$, and $S_i \neq \varnothing$, $i = 1, ..., k$.
We represent the cluster assignment with a matrix $C \in M_{\mathbb{R}}(n, k)$ whose element c_{ij} is 1 if the observation i belongs to cluster j, zero otherwise. Of course there is a one-to-one mapping between assignment matrices and partitions.

We denote with μ or M an ordered set of $\mathbb{R}^d$ vectors $(\mu_1, ..., \mu_k)$.
We say that a partition S and a set $(\mu_1, ..., \mu_k)$ solve the clustering problem presented above if they minimize the loss function

$$L(\mu, C) = \sum_{j=1}^{k} \sum_{\mathbf{x} \in S_j} d^2(\mathbf{x}, \mu_j) = \sum_{j=1}^{k} \sum_{i=1}^{n} c_{ij} d^2(\mathbf{x}_i, \mu_j)$$

μ_j is called centroid: it has the meaning of mean of observations in the cluster S_j. Note that a set of means identifies a unique partition if we decide to assign each datum to the nearest cluster, and vice versa a partition identifies a unique set of means if we compute each μ_j according to the formula $\mu_j = \frac{1}{|S_j|} \sum_{\mathbf{x} \in S_j} \mathbf{x}$.

L may be also rewritten as $L(\mu, C) = \sum_{j=1}^{k} \sum_{\mathbf{x} \in S_j} d^2(\mathbf{x}, \mu_j) = \sum_{j=1}^{k} |S_j| \, var(S_j)$, where $var(S_j)$ denotes the sample variance[2] within the cluster S_j. Of course in the Euclidean setting d is the Euclidean distance.

A straightforward algorithm to find such minimum, as explained in [12] or [7], is an iterative procedure alternating the partition and centroid estimation steps. Specifically, given a centroid estimate μ, each point $\mathbf{x}$ is assigned to

[2] considering the cardinality of S_j as denominator for the sample variance

the nearest cluster; given a partition, the centroid of each cluster is computed according to the established formula. So the algorithm proceeds as follows:

1. Given an initial guess of centroids M_0, observations grouped together in the cluster S_j are those to which μ_j is the nearest centroid: so the initial assignment matrix C_0 is such that $c_{ij} = 1$ if $d(x_i, \mu_j) \leq d(x_i, \mu_l)$ for $l = 1, ..., k$; $c_{il} = 0$ if $l \neq j$.

2. Given an assignment matrix C_t, centroids of given clusters are computed according to the formula $\mu_j = \frac{1}{|S_j|} \sum_{\mathbf{x} \in S_j} \mathbf{x}$, so we get M_{t+1}. If no better minimum is attained with M_{t+1} the algorithm stops, otherwise goes back to 3.

3. Given a set M_{t+1}, a new partition matrix C_{t+1} is determined as it is done in 1. If no better minimum is attained with M_{t+1} the algorithm stops, otherwise goes back to 2.

Assuming the usual notion of convergence of an algorithm, we present a theoretical result about the convergence of K-Means algorithm, as it is presented in [10].

Proposition 2.1. *The proposed K-Means algorithm converges.*

Proof To prove convergence we show that the loss function is guaranteed to decrease monotonically in each iteration until convergence for both steps.

Assignment step Let us introduce the following notation: given an assignment matrix C, c_i, $i = 1, ..., n$, denotes the only element equal to one in the set $\{c_{i1}, ..., c_{ik}\}$: i. e. the index of the cluster observation i has been assigned to.
Thus the loss function may be rewritten as $L(\mu, C) = \sum_{i=1}^{n} ||\mathbf{x}_i - \mu_{c_{i\cdot}}||^2$.
Let us consider a data point $\mathbf{x}_i$, let $c_{i\cdot}$ be the assignment from the previous iteration and $c^*_{i\cdot}$ be the new assignment obtained as $c^*_{i\cdot} = \arg\min_{j=1,...,k} ||\mathbf{x}_i - \mu_j||^2$.
Let C^* denote the new assignment matrix for all the n points. The change

in the loss function after this assignment step is then given by

$$L\left(\mu, C^*\right) - L\left(\mu, C\right) = \sum_{i=1}^{n} \left(\left|\left| \mathbf{x}_i - \mu_{c_{i\cdot}^*} \right|\right|^2 - \left|\left| \mathbf{x}_i - \mu_{c_{i\cdot}} \right|\right|^2 \right) \leq 0$$

The inequality holds thanks to the rule by which $c_{i\cdot}^*$ is determined, i.e. to assign $\mathbf{x}_i$ to the nearest cluster.

Centroids estimate step We can write down the original loss function $L(\mu, C)$ as follows:

$$L\left(\mu, C\right) = \sum_{j=1}^{k} \sum_{\mathbf{x} \in S_j} \left|\left| \mathbf{x} - \mu_j \right|\right|^2$$

Let us consider the j-th cluster, and let μ_j be a cluster center from the previous iteration and μ_j^* be a new cluster center obtained as $\mu_j^* = \frac{1}{|S_j|} \sum_{\mathbf{x} \in S_j} \mathbf{x}$. Let μ^* denote the new cluster centers for all the k clusters. The change in loss function after this refitting step is then given by:

$$L\left(\mu^*, C\right) - L\left(\mu, C\right) = \sum_{j=1}^{k} \sum_{\mathbf{x} \in S_j} \left(\left|\left| \mathbf{x}_i - \mu_j^* \right|\right|^2 - \left|\left| \mathbf{x}_i - \mu_j \right|\right|^2 \right) \leq 0$$

The inequality holds because the update rule of μ_j^* essentially minimizes this quantity. ■

2.2. Gaussian mixture models

K-Means algorithm for clustering, simple it may be, has considerable weaknesses (highlighted in [20]), which we briefly recall.

A first reason is that it is intrinsically non-probabilistic: we are provided with no information about the degree of confidence in the cluster assignment. A way to address this problem is to compare the distance of each point to all cluster centers: points which are positioned near the boundary of a circle,

and whose cluster assignment in the K-Means algorithm is then especially fragile, will exhibit similar distances to at least two cluster centers.

In the second place, K-Means clustering tends to break data into clusters whose shape is approximately circular, because of the shape of neighbourhoods in $\mathbb{R}^d$ with the Euclidean distance: this happens even when observations are grouped in clusters whose eye-obvious shape is non-circular. In such a case, K-Means algorithm may not be the best fit. It will then be reasonable to allow for the cluster boundaries to be non-circular.

The two alterations described are the main ingredients for a finer cluster model called gaussian mixture model (which we will next refer to as GMM). GMM, given a dataset, leads to algorithms seeking to estimate the parameters of a mixture of multi-dimensional Gaussian measures that best fit the data. GMM will also provide a probabilistic cluster assignment, encoded in a $n \times k$ matrix whose element in (i, j) represents the probability that the observations i belongs to the cluster k.

Various techniques have been developed to choose wisely the number of clusters and recognize data paths which are eye-obvious. We will not discuss this difficulty in this thesis, though; [6] or [19] are possible references. We will always assume the desired number of clusters is given as an input.

2.2.1. Expectation-Maximization algorithm for Euclidean data

In this section we explain in detail what a Gaussian mixture is and present the EM algorithm, which is discussed e. g. in [2].
Suppose we have a sample $\mathbf{x}_1, ..., \mathbf{x}_n$ from the random vectors $\mathbf{X}_1, ..., \mathbf{X}_n : \Omega \to \mathbb{R}^d$. We assume that such vectors are independent and identically distributed as a mixture of k Gaussian distributions with expected value μ_j

and covariance matrix Σ_j, $j = 1, ..., k$, i. e.

$$\mathbf{X}_1, ..., \mathbf{X}_n \overset{\text{iid}}{\sim} \sum_{j=1}^{k} \tau_j \mathcal{N}(\mu_j, \Sigma_j)$$

where $\tau_j \geq 0, j = 0, ..., k$ and $\sum_{j=1}^{k} \tau_j = 1$.
If $\mathbf{Z} : \Omega \to \mathbb{R}^n$ is a discrete random vector, with support on $\{1, ..., k\}^n$, such that $\mathbf{P}(Z_i = j) = \tau_j$ for $i = 1, ..., n$,$j = 1, ..., k$, then the distribution of the $\mathbf{X}_i$ can be disintegrated as

$$\mathbf{X}_i | Z_i = j \sim \mathcal{N}(\mu_j, \Sigma_j).$$

Thus the value of Z_i determines the component from which the i-th observation originates. The components of $\mathbf{Z}$ are called latent variables.

Given the data, we want to the recover the parameters of the k Gaussian measures and the discrete law of $\mathbf{Z}$, which represents the mixing values between the Gaussians. Therefore we aim at estimating the set of parameters

$$\theta = (\tau, \mathbf{M}, \mathbf{\Sigma})$$

where $\tau = (\tau_1, ..., \tau_k) \in \mathbb{R}^k$, $\mathbf{M} \in M_{\mathbb{R}}(d, k)$ collects the mean column vectors of each Gaussian and $\mathbf{\Sigma} \in M_{\mathbb{R}}(k, d, d)$ is a tensor such that $\mathbf{\Sigma}(j, \cdot, \cdot)$ is the covariance matrix of the j-th Gaussian.

The incomplete-data likelihood function is $L(\theta; \mathbf{x}) = \mathbf{P}(\mathbf{X}_1 = \mathbf{x}_1) \cdot ... \cdot \mathbf{P}(\mathbf{X}_n = \mathbf{x}_n) = \prod_{i=1}^{n} \sum_{j=1}^{k} \mathbf{P}(\mathbf{X}_i = \mathbf{x}_i | Z_i = j) \mathbf{P}(Z_i = j) = \prod_{i=1}^{n} \sum_{j=1}^{k} \tau_j f(\mathbf{x}_i; \mu_j, \Sigma_j)$, recalling that

$$f(\mathbf{x}; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} \sqrt{\det \Sigma}} e^{-\frac{1}{2}\langle \mathbf{x}-\mu, \Sigma^{-1}(\mathbf{x}-\mu) \rangle}$$

is the Gaussian density function.

The complete-data likelihood function is

$L(\theta; \mathbf{x}, \mathbf{z}) = p(\mathbf{x}, \mathbf{z}|\theta) = \prod_{i=1}^{n} \prod_{j=1}^{k} \left[f\left(\mathbf{x}_i; \mu_j, \Sigma_j\right) \tau_j \right]^{I_{(z_i=j)}}$, i. e.

$$L(\theta; \mathbf{x}, \mathbf{z}) = \exp \log \left(\prod_{i=1}^{n} \prod_{j=1}^{k} \left[f\left(\mathbf{x}_i; \mu_j, \Sigma_j\right) \tau_j \right]^{I_{(z_i=j)}} \right) =$$

$$\exp \sum_{i=1}^{n} \sum_{j=1}^{k} I_{(z_i=j)} \log \left(\tau_j \frac{1}{(2\pi)^{d/2} \sqrt{\det \Sigma_j}} e^{-\frac{1}{2}\left\langle \mathbf{x}_i - \mu_j, \Sigma_j^{-1}\left(\mathbf{x}_i - \mu_j\right)\right\rangle} \right) =$$

$$\exp \sum_{i=1}^{n} \sum_{j=1}^{k} I_{(z_i=j)} \left[\log \tau_j - \frac{d}{2} \log 2\pi - \frac{1}{2} \log \det \Sigma_j - \frac{1}{2} \left\langle \mathbf{x}_i - \mu_j, \Sigma_j^{-1}\left(\mathbf{x}_i - \mu_j\right)\right\rangle \right]$$

The algorithm for estimating the parameters, called Expectation-Maximization algorithm, is essentially an algorithm meant to find maximum likelihood estimators, and it does so alternating two steps: expectation (computing the expected log-likelihood) and maximization (maximizing such expectation), updating the estimate of parameters at each iteration.

We suppose we have an initial estimate of parameters $\theta^{(0)}$.

E step Given the current estimate $\theta^{(t)}$, we define

$$T_{ji}^{(t)} = \mathbf{P}\left(Z_i = j | \mathbf{X}_i = \mathbf{x}_i; \theta^{(t)}\right) = \frac{\mathbf{P}\left(\mathbf{X}_i = \mathbf{x}_i; \theta^{(t)} | Z_i = j\right) \mathbf{P}\left(Z_i = j\right)}{\mathbf{P}\left(\mathbf{X}_i = \mathbf{x}_i; \theta^{(t)}\right)}$$
$$= \frac{f\left(\mathbf{x}_i; \mu_j^{(t)}, \Sigma_j^{(t)}\right) \tau_j^{(t)}}{\sum_{j=1}^{k} \tau_j^{(t)} f\left(\mathbf{x}_i; \mu_j^{(t)}, \Sigma_j^{(t)}\right)}$$

These are called the membership probabilities: $T_{ji}^{(t)}$ encodes the probability that the i-th observation comes from the j-th Gaussian, given that the parameters are $\theta^{(t)}$.

The expected log-likelihood, computed with respect to the probability dis-

tribution of $\mathbf{Z}$ given $\mathbf{X} = \mathbf{x}$ and the current estimate $\theta^{(t)}$, is

$$Q\left(\theta|\theta^{(t)}\right) = \mathbf{E}_{\mathbf{Z}|\mathbf{X}=\mathbf{x};\theta^{(t)}}\left(\log L\left(\theta;\mathbf{x},\mathbf{Z}\right)\right) =$$

$$\mathbf{E}_{\mathbf{Z}|\mathbf{X}=\mathbf{x};\theta^{(t)}}\left(\sum_{i=1}^{n}\sum_{j=1}^{k} I_{(Z_i=j)}\left[\log\tau_j - \frac{d}{2}\log 2\pi - \frac{1}{2}\log\det\Sigma_j - \frac{1}{2}\left\langle \mathbf{x}_i-\mu_j, \Sigma_j^{-1}\left(\mathbf{x}_i-\mu_j\right)\right\rangle\right]\right)$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{k} T_{ji}^{(t)}\left[\log\tau_j - \frac{d}{2}\log 2\pi - \frac{1}{2}\log\det\Sigma_j - \frac{1}{2}\left\langle \mathbf{x}_i-\mu_j, \Sigma_j^{-1}\left(\mathbf{x}_i-\mu_j\right)\right\rangle\right]$$

M step The likelihood principle requires that we find θ maximizing $Q\left(\theta|\theta^{(t)}\right)$. τ and (μ_j, Σ_j) for each j may be found independently, since they all appear in separate linear terms.

$\tau^{(t+1)} = \arg\max_\tau \left\{\sum_{j=1}^{k}\left(\sum_{i=1}^{n} T_{ji}^{(t)}\right)\log\tau_j\right\}$, so that

$$\tau_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} T_{ji}^{(t)}$$

and similarly
$\left(\mu_j^{(t+1)}, \Sigma_j^{(t+1)}\right) = \arg\max_{(\mu_j,\Sigma_j)} \sum_{i=1}^{n} T_{ji}^{(t)}\left[-\frac{1}{2}\log\det\Sigma_j - \frac{1}{2}\left\langle \mathbf{x}_i-\mu_j, \Sigma_j^{-1}\left(\mathbf{x}_i-\mu_j\right)\right\rangle\right]$, so that

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{n} T_{ji}^{(t)}\mathbf{x}_i}{\sum_{i=1}^{n} T_{ji}^{(t)}}$$

$$\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^{n} T_{ji}^{(t)}\left(\mathbf{x}_i-\mu_j^{(t+1)}\right)\left(\mathbf{x}_i-\mu_j^{(t+1)}\right)^t}{\sum_{i=1}^{n} T_{ji}^{(t)}}$$

The iterative process stops if $Q(\theta^{(t+1)}|\theta^{(t+1)}) \leq Q\left(\theta^{(t)}|\theta^{(t)}\right) + \varepsilon$ for ε below some preset threshold.

To show convergence of EM algorithm, we need to prove Jensen's inequality (a classical result in probability theory) first.

Proposition 2.2. ***Jensen's inequality***

Let X be an integrable random variable, $g : \mathbb{R} \to \mathbb{R}$ be a convex function such that $g(X)$ is also an integrable random variable. Then

$$\mathbf{E}(g(X)) \geq g(\mathbf{E}(X))$$

Proof Since g is convex, for any point x_0 the graph of g lies entirely above its tangent at the point x_0:

$$g(x) \geq g(x_0) + b(x - x_0) \forall x \in \mathbb{R}$$

where b is the slope of the tangent. Setting $x = X$ and $x_0 = \mathbf{E}(X)$, the inequality reads

$$g(X) \geq g(\mathbf{E}(X)) + b(X - \mathbf{E}(X))$$

By taking the expected value of both sides of the inequality and using monotonicity and linearity of expected value, we obtain

$$\mathbf{E}(g(X)) \geq \mathbf{E}(g(\mathbf{E}(X))) + b(\mathbf{E}(X) - \mathbf{E}(\mathbf{E}(X))) = g(\mathbf{E}(X))$$

This proves the thesis. ■

Of course the thesis could be stated equivalently with concave functions: if g is concave, then $\mathbf{E}(g(X)) \leq g(\mathbf{E}(X))$.

In general, EM algorithm is not guaranteed to converge to a global maximum of the likelihood. However, it has the important property that the likelihood is guaranteed to increase at each iteration, as it is proved in the paper [3] which proposed the algorithm.

Proposition 2.3. *$f(x; \theta_j) \geq f(x; \theta_{j-1})$ for every x, for every j, if θ_{j-1} is obtained from θ_j following the presented steps.*

Proof The conditional probability formula

$$f(z|x;\theta) = \frac{f(x,z|\theta)}{f(x;\theta)}$$

implies that

$$\log f(x;\theta) = \log f(x,z|\theta) - \log f(z|x;\theta)$$

If we multiply both sides by $f(z|x;\theta_{j-1})$, we obtain

$$f(z|x;\theta_{j-1}) \log f(x;\theta) = f(z|x;\theta_{j-1}) \log f(x,z|\theta) - f(z|x;\theta_{j-1}) \log f(z|x;\theta)$$

Then, we can integrate over the support of Z, S_Z:

$$\int_{S_Z} f(z|x;\theta_{j-1}) \log f(x|\theta) dz =$$
$$\int_{S_Z} f(z|x;\theta_{j-1}) \log f(x,z|\theta) dz - \int_{S_Z} f(z|x;\theta_{j-1}) \log f(z|x;\theta) dz$$

Since the log-likelihood $\log f(x|\theta)$ does not depend on z and $\int_{S_Z} f(z|x;\theta_{j-1}) = 1$ we have

$$\log f(x|\theta) = \int_{S_Z} f(z|x;\theta_{j-1}) \log f(x,z|\theta) dz - \int_{S_Z} f(z|x;\theta_{j-1}) \log f(z|x;\theta) dz$$

Moreover,

$$Q(\theta, \theta_{j-1}) = \int_{S_Z} f(z|x;\theta_{j-1}) \log f(x,z|\theta) dz$$

Therefore,

$$\log f(x|\theta) = Q(\theta, \theta_{j-1}) - \int_{S_Z} f(z|x;\theta_{j-1}) \log f(z|x;\theta) dz \tag{2.1}$$

Setting $\theta = \theta_{j-1}$, we obtain

$$\log f(x|\theta_{j-1}) = Q(\theta_{j-1}, \theta_{j-1}) - \int_{S_Z} f(z|x;\theta_{j-1}) \log f(z|x;\theta_{j-1}) dz \tag{2.2}$$

Subtracting (2.2) from (2.1), we get

$$\log f(x|\theta) - \log f(x|\theta_{j-1})$$
$$= Q(\theta, \theta_{j-1}) - Q(\theta_{j-1}, \theta_{j-1}) - \int_{S_Z} f(z|x;\theta_{j-1}) \log \frac{f(z|x;\theta)}{f(z|x;\theta_{j-1})} dz$$

By Jensen's inequality, we have

$$\int_{S_Z} f(z|x;\theta_{j-1}) \log \frac{f(z|x;\theta)}{f(z|x;\theta_{j-1})} dz \leq \log\left(\int_{S_Z} f(z|x;\theta_{j-1}) \frac{f(z|x;\theta)}{f(z|x;\theta_{j-1})} dz\right)$$
$$= \log\left(\int_{S_Z} f(z|x;\theta)\right) = \log(1) = 0$$

Therefore,

$$\log f(x|\theta) - \log f(x|\theta_{j-1}) \geq Q(\theta, \theta_{j-1}) - Q(\theta_{j-1}, \theta_{j-1})$$

which, for $\theta = \theta_j$, becomes

$$\log f(x|\theta_j) - \log f(x|\theta_{j-1}) \geq Q(\theta_j, \theta_{j-1}) - Q(\theta_{j-1}, \theta_{j-1})$$

But $Q(\theta_j, \theta_{j-1}) \geq Q(\theta_{j-1}, \theta_{j-1})$ because $\theta_j = \arg\max_\theta Q(\theta, \theta_{j-1})$. Therefore $\log f(x|\theta_j) - \log f(x|\theta_{j-1}) \geq 0$, i. e.

$$\log f(x|\theta_j) \geq \log f(x|\theta_{j-1}).\blacksquare$$

2.3. Numerical illustrations

In this section we will conduct numerical illustrations to see how the two algorithms work on Euclidean data (circular and non-circular): firstly we will look at K-Means algorithm, then at EM algorithm, lastly we will compare performances, concluding that the latter algorithm works better, as we expected.

We will apply the algorithms on planar data that we randomly generate through Python built-in functions. First we generate points that clearly

should be grouped into three and four globular centers.

```
x, y = make_blobs(n_samples=200, centers=3,
       cluster_std=0.4, random_state=0)
x, y = make_blobs(n_samples=200, centers=4,
       cluster_std=0.4, random_state=0)
```

Empolying the K-Means algorithm implemented in `sklearn.cluster` (Python) we observe the following results:

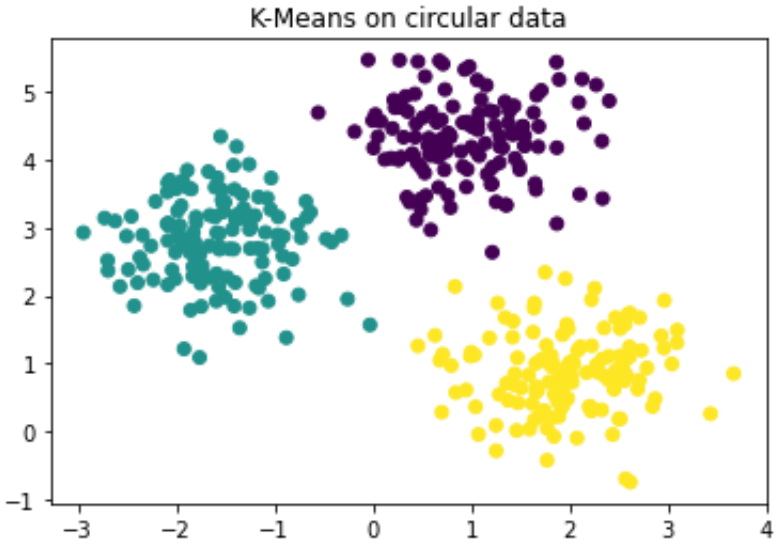

Figure 2.1: K-Means on circular data, 3 clusters

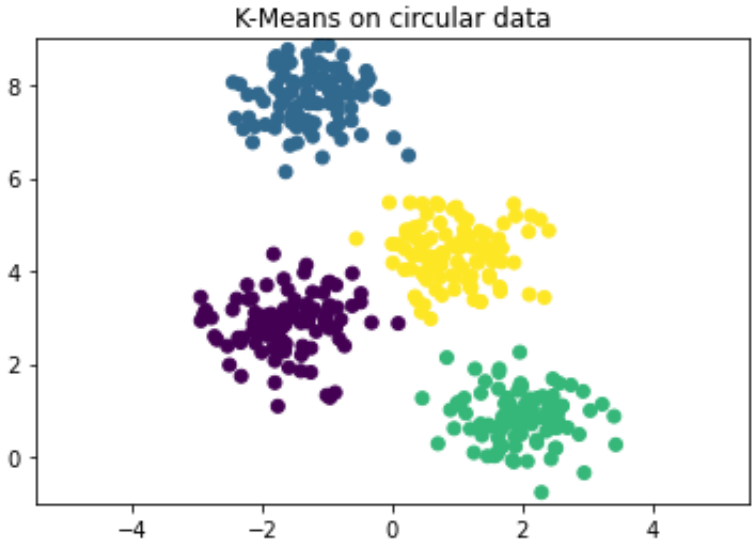

Figure 2.2: K-Means on circular data, 4 clusters

These results confirm what we said in the previous section: K-Means is a powerful tool to address clustering when points are circularly distributed because it uses the Euclidean distance to measure how far points are from the centers (and it does not take into consideration any confidence degree on how much it is likely that a point belongs to a cluster).
Therefore, we expect that it performs quite badly when it is required to cluster points on the plane that are elliptically distributed. Stretching the data obtained with the aforementioned function and employing again K-Means we observe

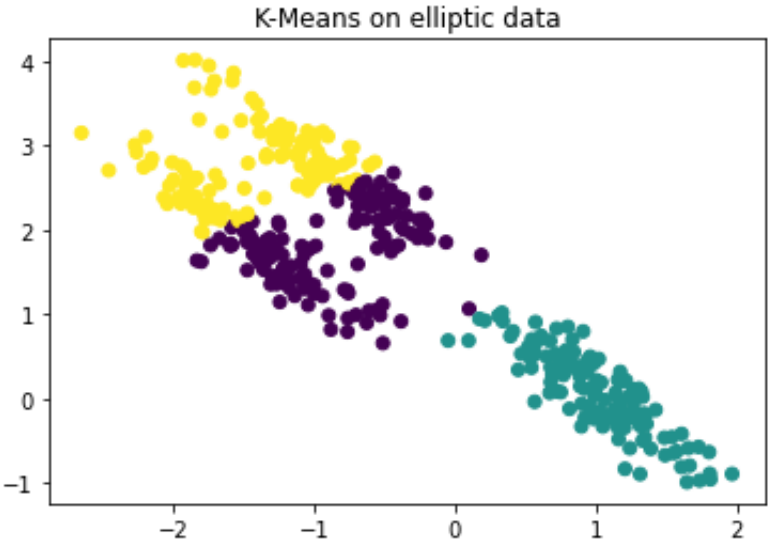

Figure 2.3: K-Means on elliptic data, 3 clusters

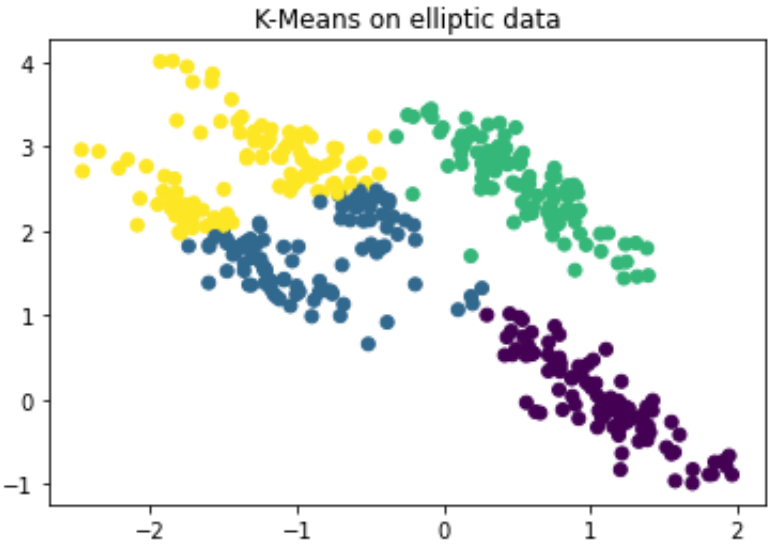

Figure 2.4: K-Means on elliptic data, 4 clusters

As we can see from the picture, when we randomly generate non-circular groups of points, cluster assignment is not satisfying at all: K-Means breaks eye-obvious elliptic clusters into circular clusters.

Using EM algorithm turns out to be the best alternative to cluster all types of data, because of the much more flexible model the algorithm rests on. It

involves a more complex definition of distance and also accounts for membership probabilities that indicate how likely it is that an observation comes from a certain distribution.
In the following pictures we observe that our implementation of Expectation-Maximization algorithm (see Appendix B for the code) has an excellent performance even when clustering non-circular data.

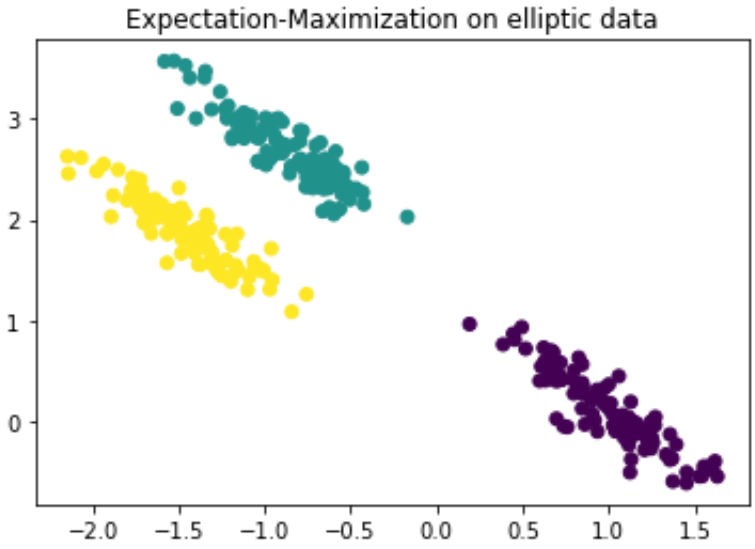

Figure 2.5: EM on elliptic data, 3 clusters

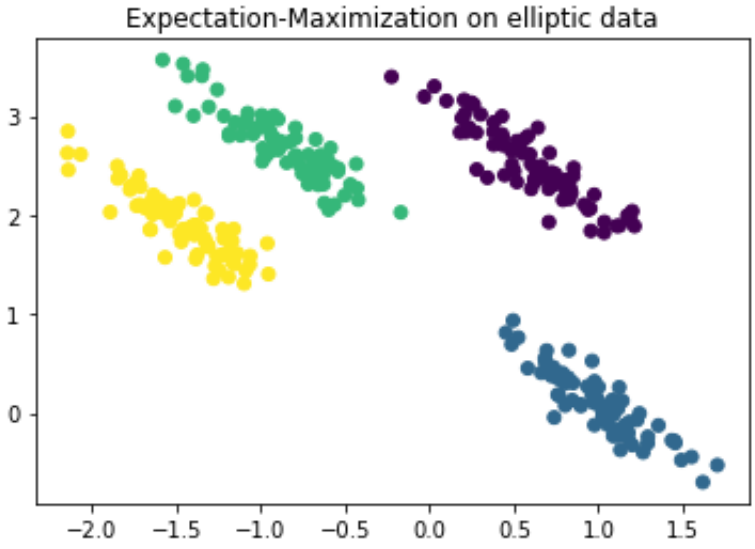

Figure 2.6: EM on elliptic data, 4 clusters

It is important to note that both algorithms may be initialized randomly: initial barycenters in K-Means may be chosen randomly among observations, while initial guesses of parameters in EM algorithm may be chosen randomly in their respective parameter spaces. This causes both algorithms to return different outputs based on random inputs.
The first draft of our implementation for EM used random initialization for parameters to be estimated, but we observed that it sometimes led to inaccurate clustering (even though the inaccuracy is limited to a few points). However, when it is initialized with K-Means output (which is still not precise), it performs really well.
We come to the conclusion that even if K-Means algorithm performs quite badly on elliptic-shaped points, its output is still the best we can initialize EM algorithm with and allows it to cluster observations perfectly.

3 | Clustering probability measures

3.1. K-Means algorithm in the Wasserstein space

As we have already hinted, the K-Means algorithm, meant to cluster Euclidean data, can be adapted to cluster probability measures: the Wasserstein distance shall be used in place of the Euclidean distance. Such algorithm has been first presented in [23]. However, we will see this algorithm has the same drawbacks as in the Euclidean case.
After presenting Wasserstein K-Means and its flaws, we will propose an extension of EM algorithm to the Wasserstein space.

The model for clustering upon which K-Means relies it the same as the one presented in the Euclidean case.
Let the set of given observations be $X = \{\mu_1, ..., \mu_n\}$, where $\mu_i \in \mathbb{W}_2(\mathbb{R})$ for $i = 1, ..., n$. We want to partition X in k clusters, which we will denote by $S_1, ..., S_k$.
We represent the cluster assignment with a matrix $C \in M_{\mathbb{R}}(n, k)$ whose element c_{ij} is 1 if the observation i belongs to cluster j, zero otherwise. Of course there is a one-to-one mapping between assignment matrices and partitions.

We denote with γ or G an ordered set of $\mathbb{W}_2(\mathbb{R})$ probability measures $(\gamma_1, ..., \gamma_k)$.
We say that a partition S and a set $(\gamma_1, ..., \gamma_k)$ solve the clustering problem

presented above if they minimize the loss function

$$L(\gamma, C) = \sum_{j=1}^{k} \sum_{\mu \in S_j} d^2(\mu, \gamma_j) = \sum_{j=1}^{k} \sum_{i=1}^{n} c_{ij} d^2(\mu_i, \gamma_j)$$

d denotes the 2-Wasserstein distance. γ_j has the meaning of mean of observations in the cluster S_j. Note that a set of means identifies a unique partition if we decide to assign each datum to the nearest cluster, and vice versa a partition identifies a unique set of means if we compute each γ_j according to the formula $\gamma_j = \arg\min_{\gamma \in \mathbb{W}_2(\mathbb{R})} \frac{1}{|S_j|} \sum_{i \in S_j} W_2^2(\mu_i, \gamma)$.

The K-Means algorithm reads as follows.
Given an initial centroid estimate $G_0 = (\gamma_1^{(0)}, ..., \gamma_k^{(0)})$, the algorithm proceeds alternating the following two steps:

1. given a centroid estimate $G_t = (\gamma_1^{(t)}, ..., \gamma_k^{(t)})$, each probability measure μ_i is assigned to the cluster S_j if and only if γ_j is the nearest centroid, according to the Wasserstein distance, to μ_i: such assignments are reported in a new assignment matrix C_t;

2. given an assignment matrix C_t, the centroid for each cluster is updated according to the rule $\gamma_j^{(t+1)} = \arg\min_{\gamma \in \mathbb{W}_2(\mathbb{R})} \frac{1}{|S_j^{(t)}|} \sum_{i \in S_j^{(t)}} W_2^2(\mu_i, \gamma)$: the updated centroids are collected in G_{t+1}.

This algorithm has the same drawbacks as in the Euclidean case: the cluster assignment is deterministic, and the clustering performed is not good when measures to be clustered are "non circular". In the section of numerical illustrations we will go through the meaning of "non-circular" and show some examples. The algorithm we develop in the following section seeks to solve such an issue.

3.2. Proposed extension of the EM algorithm to the Wasserstein space

We propose an algorithm, directly inspired to the Expectation-Maximization algorithm, able to handle the case of observations being probability measures on $\mathbb{R}$ belonging to $\mathbb{W}_2(\mathbb{R})$. It is not possible to define a density for a random variable taking values on the space of probability measures, hence we will look for an analogous of each single quantity appearing in the likelihood function, instead of directly defining a likelihood function.

Suppose we have the sample $\mu_1, ..., \mu_n$ from the random measures $\mathfrak{F}_1, ..., \mathfrak{F}_n : \Omega \to W_2(\mathbb{R})$, and suppose these measures are a "mixture" of $k < n$ measures with expected value $\gamma_{m,j}$ and variance $\mathfrak{S}_j^2$, $j = 1, ..., k$, in the sense that if $\mathbf{Z} : \Omega \to \mathbb{R}^n$ is a discrete random vector, with support on $\{1, ..., k\}^n$, such that $\mathbf{P}(Z_i = j) = \tau_j$ for $i = 1, ..., n$,$j = 1, ..., k$, then the random measure $\mathfrak{F}_i$ conditioned to $Z_i = j$ is a random measure with Fréchet mean $\gamma_{m,j}$ and Fréchet variance $\mathfrak{S}_j^2$.

The components of $\mathbf{Z}$ are called latent variables.

The quantile functions of the observed measures will be denoted by $F_1^{-1}, ..., F_n^{-1}$; the quantile functions of the expected values $\gamma_{m,j}$ will be denoted by $\Gamma_{m,1}^{-1}, ..., \Gamma_{m,k}^{-1}$; the quantile functions of the random measures will be denoted by $\mathbb{F}_1^{-1}[\omega], ..., \mathbb{F}_n^{-1}[\omega]$.

The objective is again to estimate the unknown parameters representing the mixing value and the means and variances of each random measure:

$$\theta = \left(\tau, \Gamma^{-1}, \mathfrak{S}^2\right)$$

where $\tau = (\tau_1, ..., \tau_k) \in \mathbb{R}^k$, Γ^{-1} indicates the collection of quantile functions of means $\left\{\Gamma_{m,1}^{-1}, ..., \Gamma_{m,k}^{-1}\right\}$ and $\mathfrak{S}^2 \in \mathbb{R}^k$ is a vector collecting the variances. Note that we will estimate the quantile functions of $\{\gamma_{m,j}\}_{j=1,...,k}$ instead

of directly estimating the measures, thanks to the isometric isomorphism between measures in $\mathbb{W}_2(\mathbb{R})$ and their quantile functions.

We rewrite each single term appearing in the Expectation-Maximization algorithm, following the statistical intuition underlying each term.

The function to be maximized in the Euclidean setting is

$$Q\left(\theta|\theta^{(t)}\right) = \sum_{i=1}^{n}\sum_{j=1}^{k} T_{ji}^{(t)}\left[\log\tau_j - \frac{d}{2}\log 2\pi - \frac{1}{2}\log\det\Sigma_j - \frac{1}{2}\left\langle \mathbf{x}_i - \mu_j, \Sigma_j^{-1}\left(\mathbf{x}_i - \mu_j\right)\right\rangle\right]$$

The last term, $\left\langle \mathbf{x}_i - \mu_j, \Sigma_j^{-1}\left(\mathbf{x}_i - \mu_j\right)\right\rangle$, is the squared Mahalanobis distance between $\mathbf{x}_i$ and a probability distribution with expected value μ_j and covariance matrix Σ_j. Such distance intuitively describes the distance of the observation $\mathbf{x}_i$ from a probability distribution and it may be rewritten, in the case of measures, using the Wasserstein distance.
Hence the natural analogue for $\left\langle \mathbf{x}_i - \mu_j, \Sigma_j^{-1}\left(\mathbf{x}_i - \mu_j\right)\right\rangle$ is the normalized Wasserstein distance

$$\frac{W_2^2\left(\mu_i, \gamma_{m,j}\right)}{\mathfrak{S}_j^2} = \frac{\int_0^1 \left(F_i^{-1}(u) - \Gamma_{m,j}^{-1}(u)\right)^2 du}{\mathbf{E}\left(\int_0^1 \left(\mathbb{F}_j^{-1}[\omega](t) - \Gamma_{m,j}^{-1}(t)\right)^2 dt\right)}$$

The term $\det\Sigma_j$ may be simply replaced by $\mathfrak{S}_j^2$, while $\log\tau_j$ does not need to be altered and $\frac{d}{2}\log 2\pi$ may be safely ignored, as it is irrelevant for the maximization.

To rewrite $T_{ji}^{(t)} = \frac{f\left(\mathbf{x}_i;\mu_j^{(t)},\Sigma_j^{(t)}\right)\tau_j^{(t)}}{\sum_{j=1}^{k}\tau_j^{(t)} f\left(\mathbf{x}_i;\mu_j^{(t)},\Sigma_j^{(t)}\right)}$, since we do not have a density in the Wasserstein setting, we need to find an analogous of each term appearing in the density function $f\left(\mathbf{x}_i;\mu_j,\Sigma_j\right) = \frac{1}{(2\pi)^{d/2}\sqrt{\det\Sigma_j}}\exp\left(-\frac{1}{2}\left\langle \mathbf{x}_i - \mu_j, \Sigma_j^{-1}\left(\mathbf{x}_i - \mu_j\right)\right\rangle\right)$.
According to what we said previously, we should write (up to an irrelevant

constant) $\frac{1}{\mathfrak{S}_j}\exp\left(-\frac{1}{2}\frac{W_2^2(\mu_i,\gamma_{m,j})}{\mathfrak{S}_j^2}\right)$, so that

$$T_{ji}^{(t)} = \frac{\tau_j^{(t)}\frac{1}{\mathfrak{S}_j^{(t)}}\exp\left(-\frac{1}{2}\frac{W_2^2\left(\mu_i,\gamma_{m,j}^{(t)}\right)}{\mathfrak{S}_j^{2,(t)}}\right)}{\sum_{j=1}^k \tau_j^{(t)}\frac{1}{\mathfrak{S}_j^{(t)}}\exp\left(-\frac{1}{2}\frac{W_2^2\left(\mu_i,\gamma_{m,j}^{(t)}\right)}{\mathfrak{S}_j^{2,(t)}}\right)}$$

As of the update steps, nothing changes regarding τ_j:

$$\tau_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^n T_{ji}^{(t)}$$

while the next estimate of $\Gamma_{m,j}^{-1}$ and $\mathfrak{S}_j^2$, previously $\mu_j^{(t+1)} = \frac{\sum_{i=1}^n T_{ji}^{(t)}\mathbf{x}_i}{\sum_{i=1}^n T_{ji}^{(t)}}$ and $\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^n T_{ji}^{(t)}\left(\mathbf{x}_i-\mu_j^{(t+1)}\right)\left(\mathbf{x}_i-\mu_j^{(t+1)}\right)^t}{\sum_{i=1}^n T_{ji}^{(t)}}$, should be

$$\Gamma_{m,j}^{-1,(t+1)} = \frac{\sum_{i=1}^n T_{ji}^{(t)}F_i^{-1}}{\sum_{i=1}^n T_{ji}^{(t)}}$$

$$\mathfrak{S}_j^{2,(t+1)} = \frac{\sum_{i=1}^n T_{ji}^{(t)}W_2^2\left(\mu_i,\gamma_{m,j}^{(t+1)}\right)}{\sum_{i=1}^n T_{ji}^{(t)}}$$

Note that the space of quantile functions is a vector space with respect to the usual addition and scalar multiplication defined for functions, so the above update is well defined.

3.2.1. Implementation of the EM algorithm for probability measures

In this section we propose a pseudocode implementing the changes above described in the classical Expectation-Maximization algorithm. We will denote by Q_t the value $Q(\theta^{(t)}|\theta^{(t)})$; the remaining notation has already been introduced.

Note that from now on we will suppose our observations are quantile functions $\left\{F_i^{-1}\right\}_{i=1}^n$ and not probability measures. This is justified by the already mentioned isometric isomorphism.

Algorithm 3.1 Expectation-Maximization for Wasserstein data

Input: observed quantile functions $\left\{F_i^{-1}\right\}_{i=1}^n$, number of clusters k, threshold ε

compute initial estimates for τ, Γ^{-1},$\mathfrak{S}^2$

set initial value $Q_0 << 0$

set $Q_1 > Q_0 + \varepsilon$

while $Q_{t+1} > Q_t + \varepsilon$ **do**

$Q_t = Q_{t+1}$

compute membership probabilities given current estimates for parameters:

for each j, i $T_{ji}^{(t)} = \dfrac{\tau_j^{(t)} \frac{1}{(2\pi)^{n/2}\mathfrak{S}_j^{(t)}} \exp\left(-\frac{1}{2}\frac{\left|\left|F_i^{-1}-\Gamma_{m,j}^{-1,(t)}\right|\right|^2}{\mathfrak{S}_j^{2,(t)}}\right)}{\sum_{l=1}^k \tau_l^{(t)} \frac{1}{(2\pi)^{n/2}\mathfrak{S}_l^{(t)}} \exp\left(-\frac{1}{2}\frac{\left|\left|F_i^{-1}-\Gamma_{m,l}^{-1,(t)}\right|\right|^2}{\mathfrak{S}_l^{2,(t)}}\right)}$

update estimates of parameters:

for each j $\tau_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^n T_{ji}^{(t)}$

for each j $\Gamma_{m,j}^{-1,(t+1)} = \frac{\sum_{i=1}^n T_{ji}^{(t)} F_i^{-1}}{\sum_{i=1}^n T_{ji}^{(t)}}$

for each j $\mathfrak{S}_j^{2,(t+1)} = \frac{\sum_{i=1}^n T_{ji}^{(t)} \left|\left|F_i^{-1}-\Gamma_{m,j}^{-1,(t)}\right|\right|^2}{\sum_{i=1}^n T_{ji}^{(t)}}$

compute expectation:

$Q_{t+1} = \sum_{i=1}^n \sum_{j=1}^k T_{ji}^{(t)} \left[\log \tau_j - \frac{1}{2}\log \mathfrak{S}_j^{2,(t+1)} - \frac{1}{2}\frac{\left|\left|F_i^{-1}-\Gamma_{m,j}^{-1,(t+1)}\right|\right|^2}{\mathfrak{S}_j^{2,(t)}}\right]$

end while

Output: τ, Γ^{-1}, $\mathfrak{S}^2$, T

The mentioned norm is the L^2 norm.

An example of initial estimates for parameters is the following: initialize the vector τ as the uniform discrete law (all components $\frac{1}{k}$), $\mathfrak{S}^2$ as the identity matrix, Γ^{-1} as the set of quantile functions provided by the K-Means

Wasserstein algorithm on the data.

3.3. Numerical illustrations

In this section, we will conduct numerical illustrations to see how our algorithms work on random probability measures: firstly we will look at K-Means algorithm in the Wasserstein space; then at the EM algorithm; lastly we will compare performances.
We will test both algorithms on measures that we randomly generate through Python built-in functions (thanks to the isometric isomorphism that we examined in the first section, we will actually generate quantile functions).
In order to argue about what happens when the two algorithms are applied to random measures, we need to clarify what "non-circular measure" means, as it should be an analogue of elliptic data in the plane.
We will observe that K-means algorithm performs quite well on clustering "circular" probability measures, but very poorly on "non-circular" probability measures. Moreover, we will show that our extension for the EM algorithm in the Wasserstein space addresses the problem correctly.

To find an analogue of elliptic data in the space of measures, a reasonable intuition may be thinking about quantile functions that follow the same pattern in a sub-interval of $[0, 1]$ closer to 0 and then space apart in a sub-interval of $[0, 1]$ closer to 1.
An example of probability distribution following this pattern is the skew-normal distribution with a positive skewness parameter and a high variance on the right tail.
The skewness parameter γ_1 of a real random variable $X \in L^3$ is defined as

the third standardized moment:

$$\gamma_1 := \tilde{\mu}_3 = \mathbf{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{\mu_3}{\sigma^3}$$

It is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. If the skewness is positive, the right tail is longer and the mass is concentrated on the left of the point density function. If the skewness is negative, the left tail is longer and the mass is concentrated on the right of the point density function.

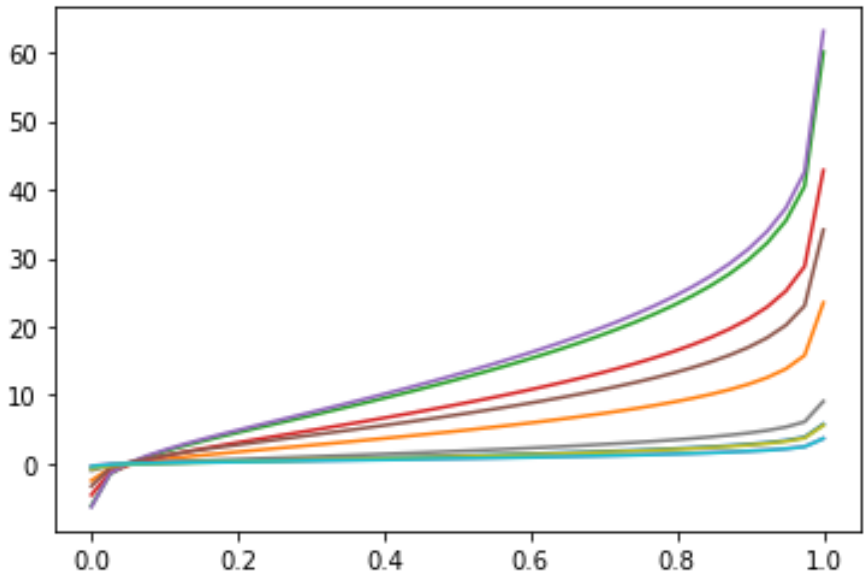

Figure 3.1: Skew normal distributions with positive skewness, quantile functions

To realize that the EM algorithm works better than the K-Means algorithm on elliptic measures, we show their performances on just two clusters. We sample the means of two skew-normal distributions from two normal distributions with means equal to 5 and -5 respectively, and both variances equal to 0.2. We sample the variances of the skew-normal distribution from a log-normal distribution with mean equal to 1 and variance equal to 2. Then, we generate skew-normal quantile functions with the previous sampled means and variances and skewness parameter equal to 5. A proper clustering algo-

rithm should group together measures with means sampled from the normal distribution with mean 5, and put into another cluster the ones from the -5 mean.

First, though, we observe that the K-Means algorithm performs well on "circular" measures such as normal distributions.

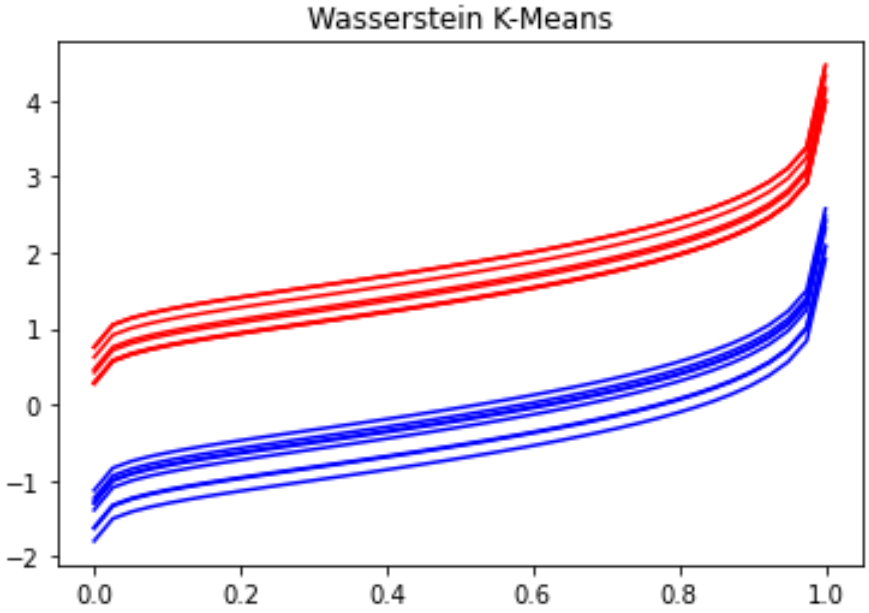

Figure 3.2: K-Means on circular measures, quantile functions

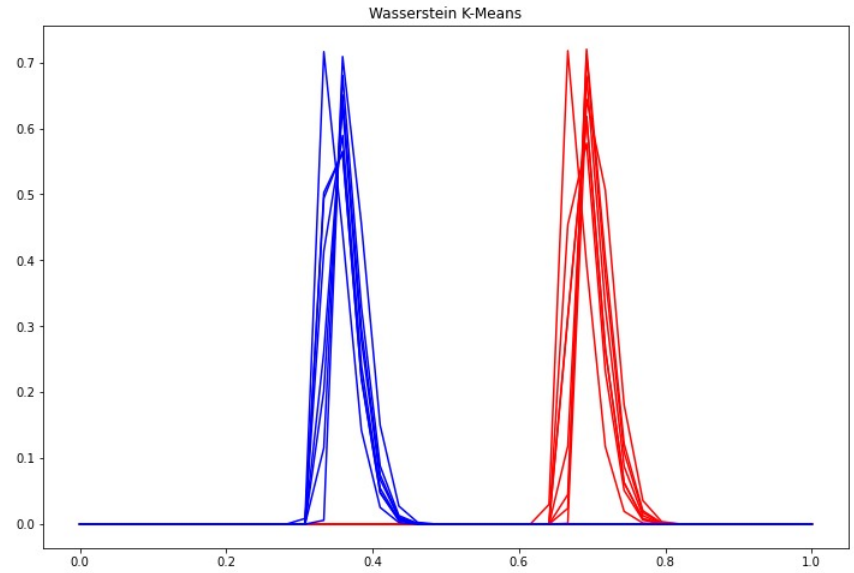

Figure 3.3: K-Means on circular measures, point density functions

However, the performance is very bad when K-Means is tested on "non-circular", in the sense explained above, probability measures, as it does not recognize the two distinct origins of measures.

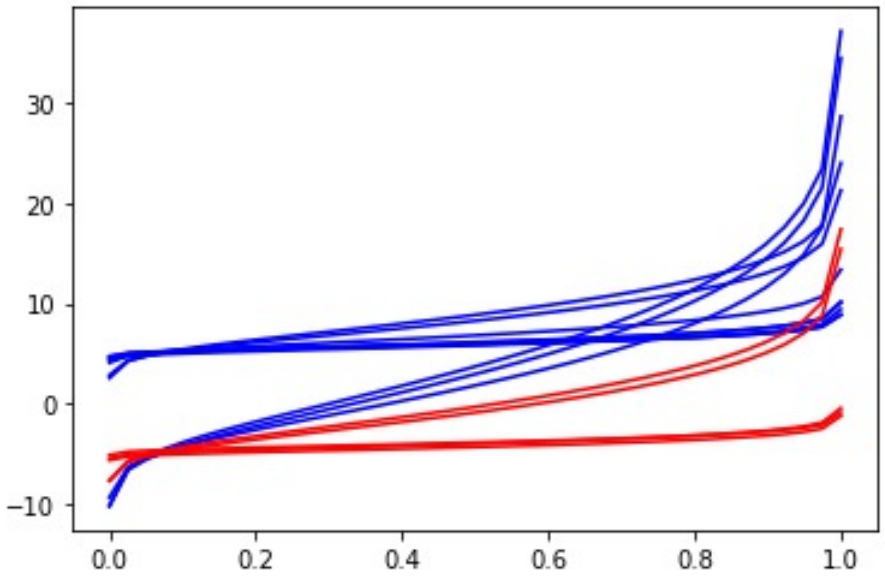

Figure 3.4: K-Means on elliptic measures, quantile functions

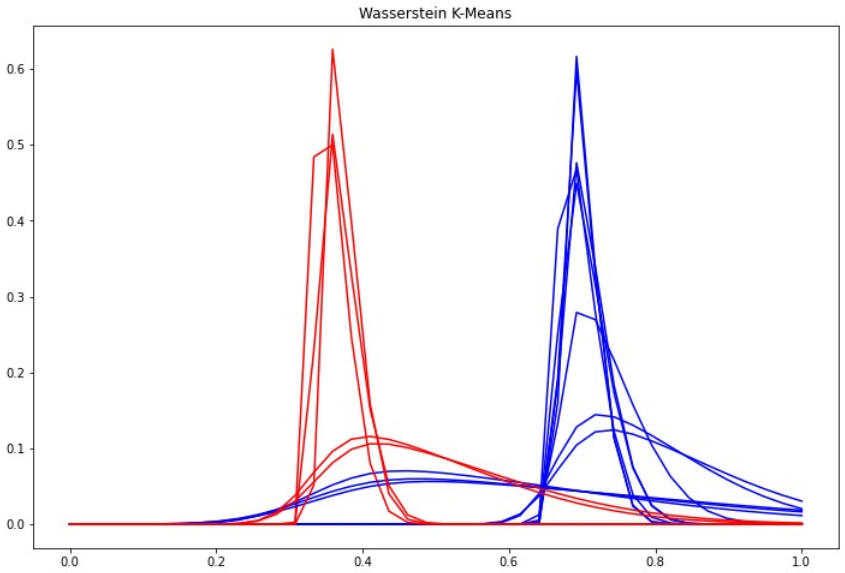

Figure 3.5: K-Means on elliptic measures, point density functions

We observe instead that the EM algorithm we developed is able to properly cluster "elliptic" measures as well, recognizing the two distinct clusters.

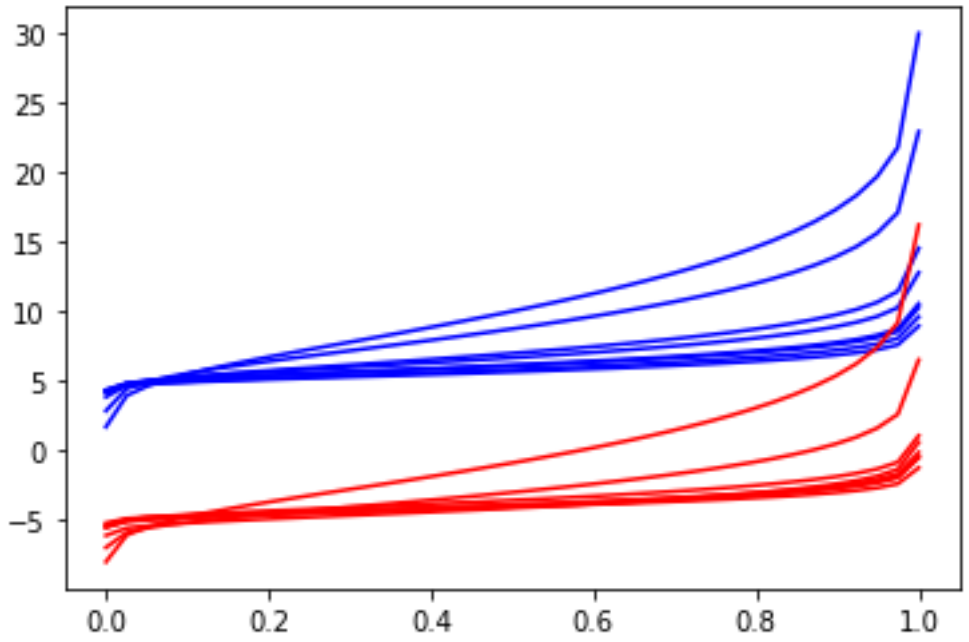

Figure 3.6: EM on elliptic measures, quantile functions; trial 1

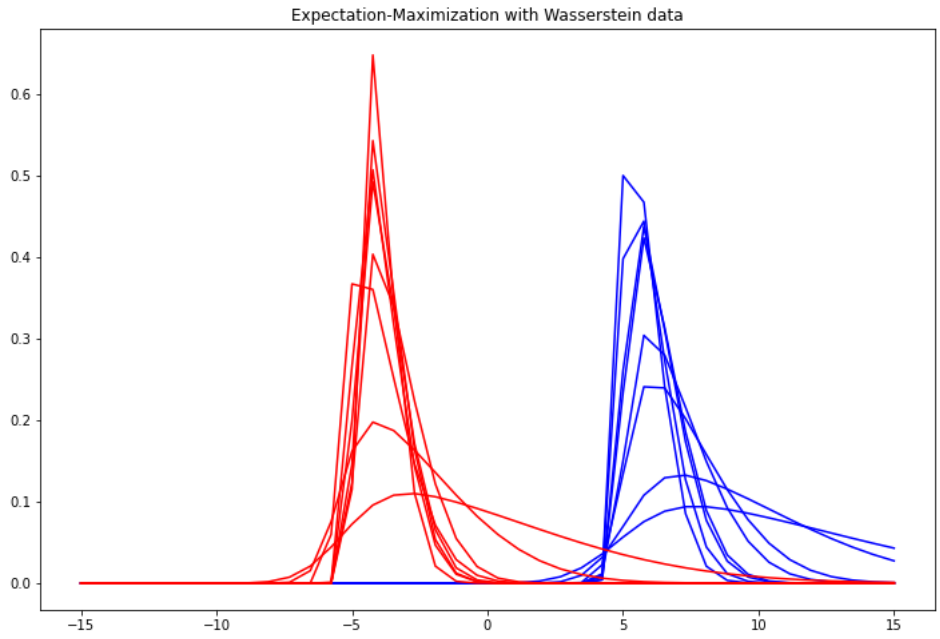

Figure 3.7: EM on elliptic measures, point density functions; trial 1

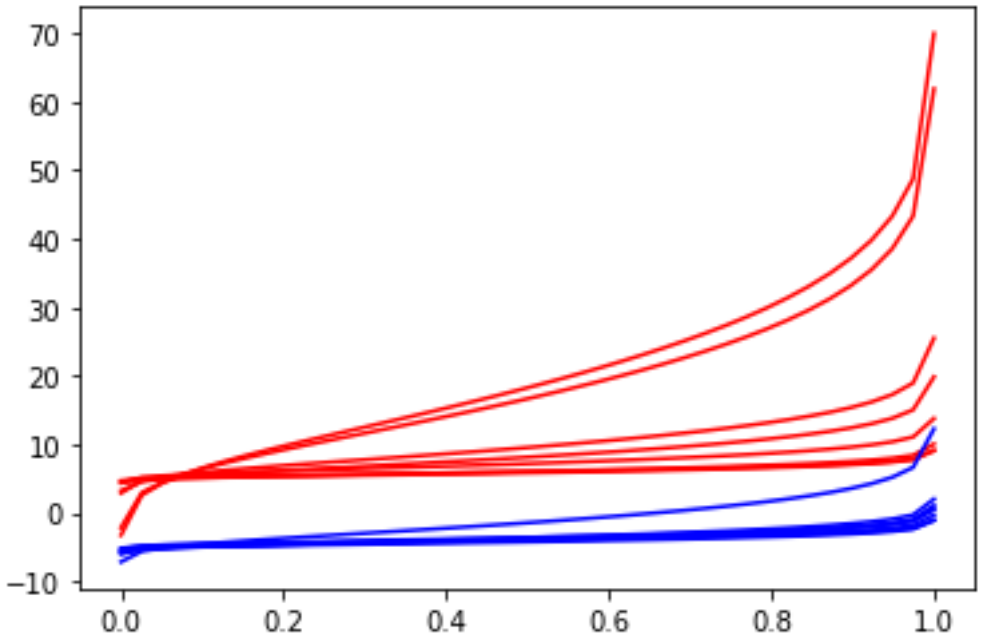

Figure 3.8: EM on elliptic measures, quantile functions; trial 2

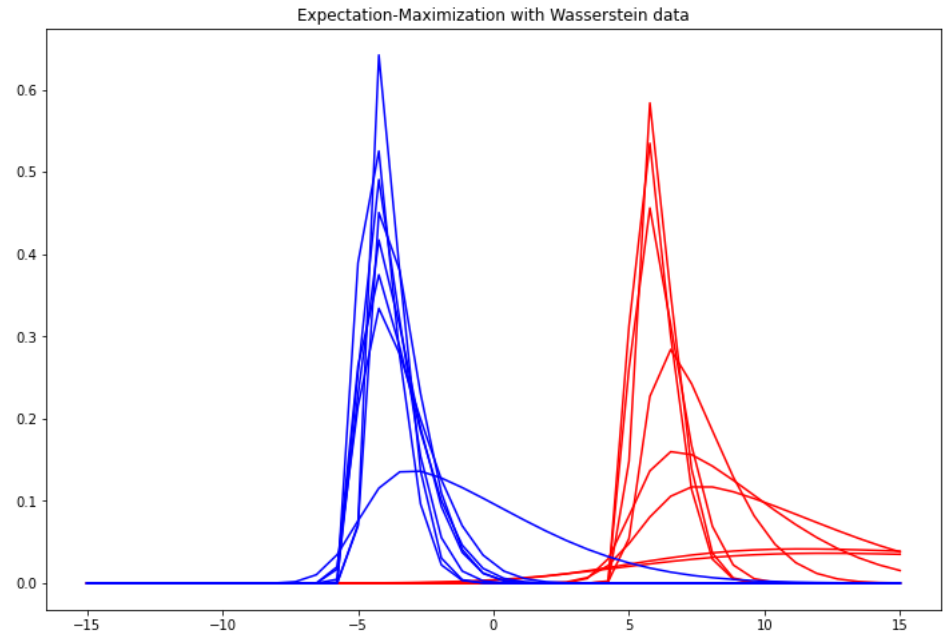

Figure 3.9: EM on elliptic measures, point density functions; trial 2

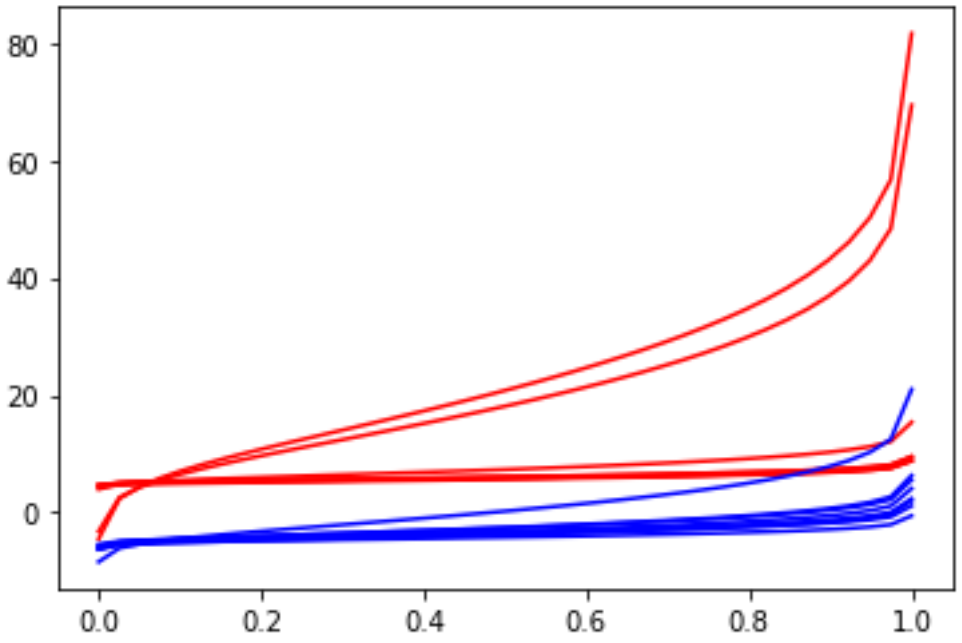

Figure 3.10: EM on elliptic measures, quantile functions; trial 3

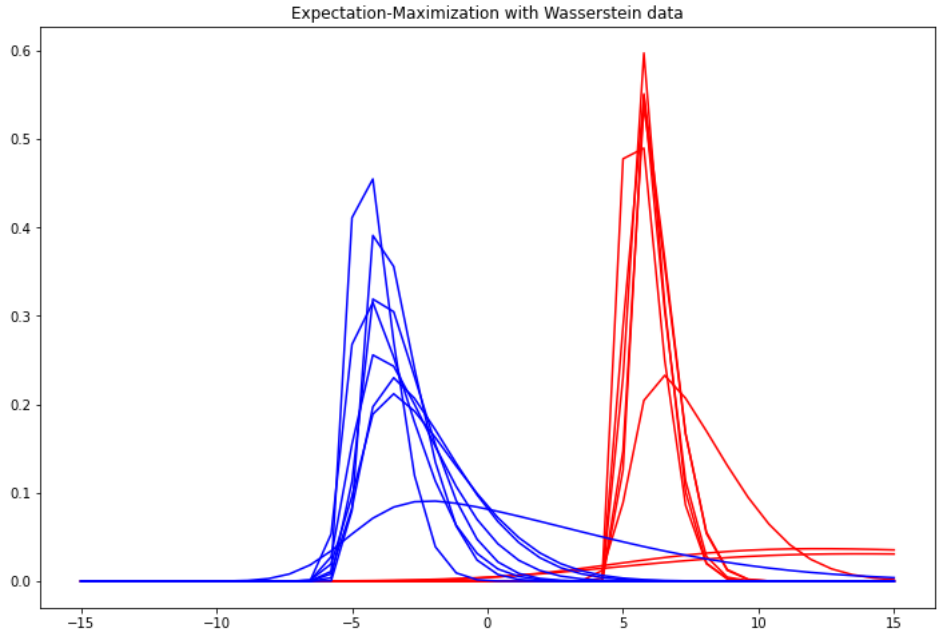

Figure 3.11: EM on elliptic measures, point density functions; trial 3

4 | Conclusions

Starting from clustering Euclidean data on the plane, we realized that the K-Means algorithm, in addition to providing a deterministic cluster assignment, struggles when it comes to cluster points that have a non-circular shape. With a probabilistic and much more flexible approach, the E-M algorithm solves these problems because it accounts for membership probabilities that describe the confidence an observation comes from a certain distribution; it is also able to cluster data with various shapes.

However, sometimes observed data are not points on the plane, but they come as realizations of random measures that we still need to be able to cluster. To make this task computationally feasible, we needed a theoretical result, that is the isometric isomorphism between the set of probability measures in $\mathbb{W}_2(\mathbb{R})$ and the set of left-continuous, non-decreasing functions defined on $(0, 1)$ and taking values in $\mathbb{R}$. This allows us to cluster random probability measures working with their quantile functions. We observed that K-Means extended to the Wasserstein space, just like in the Euclidean case, struggles to cluster "non-circular" measures. Inspired by the Euclidean case, we developed an algorithm based on the E-M algorithm which goes beyond the problem.

Despite the innovations and extensions of our work, what we did is intrinsically limited to the case of probability measures belonging to $\mathbb{W}_2(\mathbb{R})$. Indeed, without the isometric isomorphism between $\mathbb{W}_2(\mathbb{R})$, equipped with Wasserstein distance, and the space of quantile functions with the L^2 norm, we could not have treated probability random measures with their respective quantile

functions and used their properties to rewrite the Expectation-Maximization algorithm. If we were to deal with random measures belonging to $\mathbb{R}^2$ for example, we would not have the same isomorphism between the two functional spaces and it would be much harder to cluster these random measures. An alternative idea could be approximating quantile functions of probability measures on $\mathbb{R}^d$ with a sum of delta quantile functions, but this would not guarantee the uniqueness of the centroid quantile function.

Bibliography

[1] L. Ambrosio, A. Bressan, D. Helbing, A. Klar, E. Zuazua, and N. Gigli. A user's guide to optimal transport. *Modelling and Optimisation of Flows on Networks: Cetraro, Italy 2009, Editors: Benedetto Piccoli, Michel Rascle*, pages 1–155, 2013.

[2] G. Casella and R. L. Berger. Statistical lnference. *Duxbury Press*, 2002.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 1977.

[4] P. Embrechts and M. Hofert. A note on generalized inverses. *Mathematical Methods of Operations Research*, 77, no.3:423–432, 2013.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise.

[6] T. Hastie, R. Tibshirani, and G. Walther. Estimating the number of clusters in a data set via gap statistic. *Journal of the Royal Statistical Society, Series B*, 2001.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics, 2009.

[8] O. Kallenberg. *Foundations of modern probability*, volume 2. Springer, 1997.

[9] L. Kantorovich. On the translocation of masses. 1942.

[10] S. P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28, no.2, 1982.

[11] Q. Luan and J. Hamp. Automated regime detection in multidimensional time series data using sliced wasserstein k-means clustering.

[12] D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[13] G. Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie R. des Sci. de Paris*, 1781.

[14] A. Ng. Cs229 lecture notes.

[15] V. M. Panaretos and Y. Zemel. *An Invitation to Statistics in Wasserstein Space*. SpringerLink, 2020.

[16] A. Petersen and H.-G. Mueller. Wasserstein covariance for multiple random densities. *Biometrika*, 106, no.2:339–351, 2019.

[17] R. Rao, A. Moscovich, and A. Singer. Wasserstein k-means for clustering tomographic projections.

[18] F. Santambrogio. *Optimal Transport for Applied Mathematicians: Calculus of Variations, Pdes, and Modeling*. Birkhäuser, 2015.

[19] C. A. Sugar and G. M. James. Finding the number of clusters in a data set: An information-theoretic approach. *Journal of the American Statistical Association*, 2001.

[20] J. VanderPlas. *Python Data Science Handbook*. O'Reilly Media, 2016.

[21] C. Villani. *Topics in Optimal Transportation*. Springer Link, 2003.

[22] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[23] Y. Zhuang, X. Chen, and Y. Yang. Wasserstein k-means for clustering probability distributions. NEURIPS, 2022.

A | Appendix A

A.1. Absolutely continuous measures

To define absolute continuity for probability measures on generic separable Banach spaces we use Gaussian measures.

Definition Given a separable Banach space $\mathcal{X}$ and its topological dual $\mathcal{X}^*$, we call non-degenerate Gaussian measure a probability measure $\mu \in P(\mathcal{X})$ such that for each $T \in \mathcal{X}^* \setminus \{0\}$, $T\#\mu \in P(\mathbb{R})$ is a Gaussian measure with positive variance.

0 is the inverse element of addition in the vector space $\mathcal{X}^*$. With such definition we trace back to the usual old definition of Gaussian measure.

Definition Given a separable Banach space $\mathcal{X}$ and $A \subseteq \mathcal{X}$, we say that A is a Gaussian null set if $\nu(A) = 0$ for each non-degenerate Gaussian measure ν. A probability measure $\mu \in P(\mathcal{X})$ is said to be absolutely continuous if μ vanishes on all Gaussian null sets.

B | Appendix B

B.1. Expectation-Maximization algorithm to cluster Euclidean data on Python

The following code is our Python implementation of the expectation-maximization algorithm meant to cluster Euclidean data, i. e. n vectors in $\mathbb{R}^m$. The notation is very similar to the one used throughout the work:

A = $m \times n$ matrix of observations, each column is an observed vector
k = number of clusters
eps = preset threshold to stop algorithm (ε)
T = membership probabilities matrix
tau = mixture parameters ($\tau = (\tau_1, ..., \tau_k)$)
mu = $m \times k$ matrix of means, each column is the mean of a Gaussian component
sigma = $k \times m \times m$ tensor of covariance matrices of the Gaussian components
Q = expected likelihood given current estimate of parameters

The code is split in various functions: one to compute membership probabilities (i. e. T), one to update the estimates of parameters, one to compute the expected log-likelihood, and finally the body of the algorithm.
There are multiple possible choices for the initial estimates of parameters. We chose the uniform discrete law for tau, the identity matrix for each matrix in sigma, the K-Means final centroids for mu.

```
import numpy as np
import math
from scipy.stats import multivariate_normal as mvn
from sklearn.cluster import KMeans


#create color map for cluster assignment based on T
def colori(A,T): ##A is a nxd matrix with observations
    n=len(A)
    lab=np.zeros(n)
    for i in range(n):
        lab[i]=np.argmax(T[i,:])
    return lab

#define functions for quicker determinant and inverse computation
def det(A):
    d=np.linalg.det(A)
    return d

def inv(A):
    B=np.linalg.inv(A)
    return B

def memb_prob(A,mu,sigma,tau,k): #compute membership probabilities for
    data
  #given current estimate of parameters
    n=len(A[0])
    T=np.zeros((k,n))
    for l in range(k):
        for i in range(n):
            c=np.zeros(k)
            for h in range(k):
                c[h]=tau[h]*mvn.pdf(A[:,i],mu[:,h],sigma[h,:,:])
            T[l,i]=c[l]/sum(c)
    return T

def update(A,T): #update estimate of parameters in order to maximize
    likelihood
    k=len(T)
    n=len(A[0])
    m=len(A)
    tau=np.zeros(k)
    mu=np.zeros((m,k))
```

```
    sigma=np.zeros((k,m,m))

    for j in range(k):
        tau[j]=sum(T[j,:])/n

    for j in range(k):
        mu[:,j]=A.dot((T[j,:]).T)/sum(T[j,:])

    for j in range(k):
      s=np.zeros((m,m))
      for i in range(n):
          a = (A[:,i]-mu[:,j])[np.newaxis]
          s+=T[j][i]*(a.T).dot(a)
      sigma[j,:,:]=s/sum(T[j,:])

    return (tau,mu,sigma)

def exploglikelihood(A,T,tau,mu,sigma): #compute expected likelihood Q
    given
  #current estimate of parameters
       n=len(A[0])
       m=len(A)
       k=len(T)
       Q=0
       for i in range(n):
              for j in range(k):
                 Q+=T[j,i]*(np.log(tau[j])-0.5*np.log(det(sigma[j,:,:]))-
                     0.5*((A[:,i]-mu[:,j]).T).dot(inv(sigma[j,:,:]).dot(A
    [:,i]-mu[:,j]))-0.5*m*np.log(2*math.pi))
       return Q

def EMalgorithm(A,k,eps):

      m=len(A)
      n=len(A[0])

      #initialize parameters to be estimated

      mu=np.zeros((m,k))

      # kmeans initialization
      km = KMeans(3)
      km.fit(A.transpose())
      centers = km.cluster_centers_
```

```
84      mu=centers.transpose()
85
86      sigma=np.zeros((k,m,m))
87      for i in range(k):
88           sigma[i,:,:]=np.identity(m)
89
90      tau=np.ones(k)/k
91
92      T=np.zeros((k,n))
93
94      Q_old=-math.exp(20)
95      Q=eps+0.1+Q_old
96
97      while (Q>Q_old+eps): #stop criterion: stop when the likelihood stops
98      #increasing more than eps
99           Q_old=Q
100
101          #E step
102          T=memb_prob(A,mu,sigma,tau,k)
103
104          #M step
105          [tau,mu,sigma]=update(A,T)
106          Q=exploglikelihood(A,T,tau,mu,sigma)
107
108     return(tau,mu,sigma,T)
```

B.2. Expectation-Maximization algorithm to cluster probability measures on Python

We report the code implementing the proposed extension of the Expectation-Maximization algorithm.

Of course we do not know the analytic expression of the observed quantile functions $\left\{F_i^{-1}\right\}_{i=1}^n$: we will be provided with m samples from each function. The matrix collecting such discretization of $\left\{F_i^{-1}\right\}_{i=1}^n$ will be denoted with $invF$, belonging to the set of $m \times n$ matrices.

The set of quantile functions of the means of the k random measures has so far been denoted with Γ^{-1}; we will denote with G the $m \times k$ matrix collecting discretizations of such quantile functions.

So the notation is:

invF = matrix of quantile functions of observed measures
k = number of clusters
eps = preset threshold to stop algorithm (ε)
T = membership probabilities matrix
tau = mixture parameters ($\tau = (\tau_1, ..., \tau_k)$)
G = matrix of quantile functions means
sigma = vector of variances
Q = expected likelihood given current estimate of parameters

```
import numpy as np
import math
from sklearn.cluster import KMeans

def memb_prob(invF,G,sigma,tau,k): #compute membership probabilities for data
  #given current estimate of parameters
    n=len(invF[0])
    T=np.zeros((k,n))

    for l in range(k):
        for i in range(n):
            c=np.zeros(k)
            for h in range(k):
                a=math.exp((-(np.linalg.norm(invF[:,i]-G[:,h]))**2)/(2*sigma[h]))
                c[h]=tau[h]*(1/(2*math.pi*sigma[h]))*a #sum
        #of all weighed gaussian distribution evaluated at i-th observations
            T[l,i]=c[l]/sum(c)

    return T

def update(invF,T): #update estimate of parameters in order to maximize likelihood
    k=len(T)
    n=len(invF[0])
```

```
    m=len(invF)
    tau=np.zeros(k)
    G=np.zeros((m,k))
    sigma=np.zeros(k)

    for j in range(k):
       tau[j]=sum(T[j,:])/n

    for j in range(k):
       G[:,j]=invF.dot((T[j,:]).T)/sum(T[j,:])

    for j in range(k):
      s=0
      for i in range(n):
          s+=T[j][i]*(np.linalg.norm(invF[:,i]-G[:,j]))**2
      sigma[j]=s/sum(T[j,:])
    return (tau,G,sigma)

def exploglikelihood(invF,T,tau,G,sigma): #compute expected likelihood Q
    given
  #current estimate of parameters
       n=len(invF[0])
       m=len(invF)
       k=len(T)
       Q=0
       print(sigma)
       for i in range(n):
              for j in range(k):
                 Q+=T[j,i]*(np.log(tau[j])-0.5*np.log(sigma[j])-
                     0.5*((np.linalg.norm(invF[:,i]-G[:,j]))**2)/(sigma[j
    ])-0.5*m*np.log(2*math.pi))
       return Q

def EMWASSERSTEIN(invF,k,eps):

      m=len(invF)
      n=len(invF[0])

      #initialize parameters to be estimated
      G=np.zeros((m,k))
      km = KMeans(3)
      km.fit(invF.transpose())
      centers = km.cluster_centers_
      G=centers.transpose()
```

```
66
67      sigma=np.ones(k)
68
69      tau=np.ones(k)/k
70
71      T=np.zeros((k,n))
72
73      Q_old=-math.exp(20)
74      Q=eps+0.1+Q_old
75
76      while (Q>Q_old+eps): #stop criterion: stop when the likelihood stops
77      #increasing more than eps
78           Q_old=Q
79
80           #E step
81           T=memb_prob(invF,G,sigma,tau,k)
82
83           #M step
84           [tau,G,sigma]=update(invF,T)
85           Q=exploglikelihood(invF,T,tau,G,sigma)
86
87      return(tau,G,sigma,T)
```

List of Figures