Universidad de Valladolid

Programa de doctorado en matemáticas

TESIS DOCTORAL

Statistical Distances For Model Validation And Clustering. Applications To Flow Cytometry And Fair Learning.

Presentada por Hristo Inouzhe Valdes
para optar al grado de
Doctor por la Universidad de Valladolid

Dirigida por:
Prof. Eustasio del Barrio Tellado
Prof. Carlos Matrán Bea

To my family for their infinite patience and support. To my friends and loved ones for their attention and interest.

Contents

Introduction (English)

This thesis has been developed at the University of Valladolid and IMUVA within the framework of the project *Sampling, trimming, and probabilistic metric techniques. Statistical applications* whose main researchers are Carlos Matrán Bea and Eustasio del Barrio Tellado. Among the lines of research associated with the project are: model validation, Wasserstein distances and robust cluster analysis. It is precisely the work carried out in these fields that gives rise to chapters 1,2 and 4 of this report.

The work done in the field of fair learning with Professor Jean-Michel Loubes, frequent collaborator with Valladolid's team, during the international stay at the Paul Sabatier University of Toulouse, is the basis of Chapter 3 of this report.

Therefore, this thesis is an exposition of the problems and results obtained in the different fields previously mentioned. Due to the diversity of topics, we have decided to base chapters on the works published or submitted to the present date, and therefore each chapter has a structure relatively independent of the others. In this way Chapter 1 is based on the works [del Barrio et al., 2019e,del Barrio et al., 2019d], Chapter 2 is based on the work [del Barrio et al., 2019c], Chapter 3 on the work [del Barrio et al., 2019b] and Chapter 4 shows results of a work in progress.

In this introduction our objective is to present the main challenges we have faced, as well as to briefly present our most relevant results. On the other hand, each chapter will have its own introduction where we will delve into the topics discussed below. With this in mind, our intention is that the reader will have a general idea of what he or she will find in each chapter and in this way will have the necessary information to face the more technical discussions that will be found there.

Due to the diversity of topics dealt with in this report, we propose a non-linear reading. We suggest that the reader, after reading a section of the Introduction, moves to the corresponding chapter. In this way the reader will have the relevant information more at hand and will be able to follow better the exposition in each chapter. If on the other hand there is a sequential reading of the document, we apologize in advance for some repetitions and reiterations, which nevertheless seem to us to contribute positively to the understanding of this work.

Approximate validation of models

It is a well-known fact that classic goodness-of-fit tests are excessively rigid for large samples. This means that very often the model will be rejected for large datasets, although most subsamples with sizes smaller than a certain size would not result in rejection. The poor behaviour of these procedures has been interpreted by some authors as an indicator

of model falsity, to the point of stating, as in [Liu and Lindsay, 2009], that for any data generating mechanism there is a sample size from which model failure is obvious. This difficulty has been approached by different authors, coinciding in the consideration of relaxations of the null hypothesis, considering extended but equally useful models that also tend to involve a gain in robustness [Hodges and Lehmann, 1954, Munk and Czado, 1998, Álvarez-Esteban et al., 2012].

From a different but somewhat complementary point of view, we have Box's famous phrase "all models are false, but some are useful", which even under the model's falsehood paradigm leads us to consider some measure of to what extent the model would be useful or preferable to others. The works [Davies, 1995, Davies, 2018] address these questions from the perspective that a model is useful as long as it is capable of generating samples similar to the data at hand.

Over the course of successive projects, the proposals within this quasi-validation or approximate model validation scheme can be found in [Álvarez-Esteban et al., 2008, Álvarez-Esteban et al., 2012]. The approach starts from a contamination model, in which two probabilities, P_1 and P_2, would be similar at level α if they admitted a simultaneous decomposition:

$$P_1 = (1-\alpha)P_0 + \alpha Q_1, P_2 = (1-\alpha)P_0 + \alpha Q_2 \tag{1}$$

where P_0, Q_1 and Q_2 are arbitrary probabilities. For small values of α, this would guarantee that the samples obtained from these probabilities would be practically homogeneous, since most of the data would come from the same P_0.

The similarity model (1) can be characterized in terms of total variation distance, where P_1 and P_2 are defined in the σ-algebra $\mathfrak{A}$ over Ω, as

$$d_{VT}(P_1, P_2) = \sup_{A \in \mathfrak{A}} |P_1(A) - P_2(A)| \leq \alpha,$$

but, as is well known, this distance is only appropriate in statistical problems with discrete support. A more general characterization needs some extra tools like generalized trimmings and contamination neighbourhoods.

Generalized trimmings of a probability were introduced in [Gordaliza, 1991]. A probability $\tilde{P} \in \mathbb{R}$ is a trimming of level $\alpha \in [0,1)$ of P when there is a function w such that $0 \leq w \leq 1$ and $\tilde{P}(B) = \frac{1}{1-\alpha}\int_B w(x)P(dx)$ for all $B \in \beta$. Equivalently, it must be absolutely continuous with respect to P and with Radon-Nykodim derivative bounded by $\frac{1}{1-\alpha}$. We will denote the set of α-trimmings of a probability distribution P as $R_\alpha(P)$:

$$R_\alpha(P) = \left\{\tilde{P} \in \mathcal{P} : \tilde{P} \ll P,\ \tfrac{d\tilde{P}}{dP} \leq \tfrac{1}{1-\alpha}\ P\text{-a.s.}\right\}. \tag{2}$$

In his seminal work [Huber, 1964], Huber introduced the contamination neighbourhoods of a probability, making them one of the pillars of robust statistics. An (α-) contamination neighbourhood of a probability distribution P_0 is the following set of probability distributions

$$\mathcal{V}_\alpha(P_0) = \{(1-\alpha)P_0 + \alpha Q : Q \in \mathcal{P}\}, \tag{3}$$

where $\mathcal{P}$ is the set of all probability distributions in the space. Although it can be defined in a completely general way, in the first chapter $\mathcal{P}$ will be the set of probabilities defined in the (Borel) sets, β, of the real line $\mathbb{R}$. If F and F_0 are distribution functions, we will use

$R_\alpha(F)$ and $\mathcal{V}_\alpha(F_0)$, with the same meanings as before, but defined in terms of distribution functions.

Thus, given an "ideal" model P_0, the neighbourhood includes those probability distributions that are a product of a distortion by bulge or rounding errors: given a value $\alpha \in [0,1)$, a probability P of $\mathcal{V}_\alpha(P_0)$ would generate samples with approximately $(1-\alpha)\times 100$ percent of the data coming from P_0. Alternatively, such a sample could be conveniently "trimmed" to get an authentic sample of the model. In fact, even P_0 could be obtained from an appropriate trimming of P.

A fundamental fact is that contamination neighbourhoods are related to trimmings (see [Álvarez-Esteban et al., 2011]) through

$$P \in \mathcal{V}_\alpha(P_0) \Longleftrightarrow P_0 \in R_\alpha(P). \tag{4}$$

In this way, model (1) can be expressed as $P_1, P_2 \in \mathcal{V}_\alpha(P_0)$ or equivalently for an appropriate distance in the space of probabilities

$$0 \leq d(R_\alpha(P_1), R_\alpha(P_2)) = \inf_{\tilde{P}\in R_\alpha(P_1), \tilde{Q}\in R_\alpha(P_2)} d(\tilde{P},\tilde{Q}) \leq d(P_0,P_0) = 0.$$

A particular case is when we want to check whether $P \in \mathcal{V}_\alpha(P_0)$, i.e. to see that $d(P_0, R_\alpha(P)) = 0$.

The problem with these characterizations is that we do not know either α or the actual contaminated distribution P. In reality we usually have an approximation $\hat{P}$ to P; normally $\hat{P}$ is the empirical distribution, and our goal is to look for statistical evidence, based on $\hat{P}$, for or against the hypothesis $P \in \mathcal{V}_\alpha(P_0)$. For this task we use a metric, d, in the space $\mathcal{P}$ and consider $d(P_0, R_\alpha(\hat{P}))$ as an estimator of $d(P_0, R_\alpha(P))$.

The L_2 Wasserstein distance

$$\mathcal{W}_2^2(P_1,P_2) = \inf_{\pi\in\Pi(P_1,P_2)} \int \|x-y\|^2 d\pi(x,y) = \inf\left\{E\|X-Y\|^2, \mathcal{L}(X)=P_1, \mathcal{L}(Y)=P_2\right\},$$

where $\Pi(P_1,P_2)$ is the set of probabilities in $\Omega\times\Omega$ with first marginal P_1 and second marginal P_2 and $\mathcal{L}(X)$ is the law of X, is the choice of metric used in [Álvarez-Esteban et al., 2011].

Our proposal is based on the same principles, but using the Kolmogorov (or L_∞) distance between the distribution functions, explicitly,

$$d_K(F,G) = \sup_{x\in\mathbb{R}} |F(x)-G(x)|, \tag{5}$$

(We will use the notation $\|F-G\|$ and $d_K(F,G)$ interchangeably). After all, the Kolmogorov-Smirnov test is probably the best-known and most widely used of the goodness-of-fit contrasts, so it is a privileged framework for developing these ideas.

In particular we have studied the properties of the plug-in estimator $d_k(F_0, R_\alpha(F_n))$, where F_n is the empirical distribution function based on a sample of n independent random variables with common distribution F (Proposition 1.2). We have also developed an algorithm for the efficient computation of the estimator (Theorem 1.4). We have developed a test with exponentially small error probabilities, based on the previous statistic, to contrast $H_0 : d_k(F_0, R_\alpha(F)) = 0$ against the alternative $d_k(F_0, R_\alpha(F)) > \rho$ (Theorem

1.6). As a main contribution we highlight Theorem 1.9, a central limit theorem where we show that $\sqrt{n}\,(d_K(F_0, R_\alpha(F_n)) - d_K(F_0, R_\alpha(F))$ converges to the supreme of a Gaussian process.

As an application of our theoretical results it is worth noting our estimator of $\alpha^* = \min\{\alpha : F \in \mathcal{V}_\alpha(F_0)\}$, the minimum level of contamination that would allow the underlying distribution to be included in the original model, which is more general than the one introduced in [Rudas et al., 1994] since it is not restricted to the multinomial case. We also provide confidence bounds for the size of the sub-samples that would be compatible with the model, generalizing the point-wise estimate given in [Lindsay and Liu, 2009].

Optimal-transport approach to flow cytometry

Recent advances on the optimal transport problem have resulted in relevant applications in the analysis of clustering procedures. These seek to divide a series of data into clusters (groups), thus producing what is known as partitions or clusterings. Analysis procedures compare different clusterings (cluster validation) and attempt to produce summary partitions that in some optimal way represent a set of partitions (consensus clustering). In particular our interest is going to focus on procedures based on optimal transport, so we briefly present some fundamental results that we will use later.

Following [Villani, 2009], let $\mathcal{P}(\Omega)$, be a probability space in Ω. For $\mu, \nu \in \mathcal{P}(\Omega)$, let $\Pi(\mu, \nu)$ be the set of probabilities π in $\Omega \times \Omega$ with first marginal μ and second marginal ν. The optimal transport cost between the two measures is defined as

$$C(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int c(x, y) d\pi(x, y) \tag{6}$$

where $c(x, y)$ is the cost of transporting a unit of mass from x to y. A probability π that reaches the minimum in (6) is called an optimal pairing, and has an associated random variable (X, Y) with a joint distribution π. When μ and ν are discrete, that is, $\mu = \sum_{k=1}^{n} p_k \delta_{x_k}$ and $\nu = \sum_{l=1}^{m} q_l \delta_{y_l}$, with $x_k, y_l \in \Omega$, the optimal transport problem can be posed as the following linear programming problem (see [Bertsimas and Tsitsiklis, 1997])

$$C(\mu, \nu) = \sum_{k=1}^{n} \sum_{l=1}^{m} w^*_{kl} c(x_k, y_l), \tag{7}$$

where (w^*_{kl}) are the solutions to the optimal-transport linear program

$$\begin{array}{lll} \text{minimize} & \sum_{k=1}^{n} \sum_{l=1}^{m} w_{kl} c(x_k, y_l) & \\ \text{subject to} & w_{kl} \geq 0, & 1 \leq k \leq n, 1 \leq l \leq m \\ & \sum_{l=1}^{m} w_{kl} = p_k, & 1 \leq k \leq n \\ & \sum_{k=1}^{n} w_{kl} = q_l, & 1 \leq l \leq m \\ & \sum_{k=1}^{n} \sum_{l=1}^{m} w_{kl} = 1. & \end{array}$$

For (Ω, d) a Polish metric space and $p \in [1, \infty)$, the $p-$Wasserstein distance between μ and ν is defined as

$$\mathcal{W}_p^p(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int d^p(x, y) d\pi(x, y) = \inf \left\{ E d^p(X, Y), \mathcal{L}(X) = \mu, \mathcal{L}(Y) = \nu \right\}, \tag{8}$$

where $\mathcal{L}(X)$ is the law of X.

We will make use of the Wassserstein distance in an important task such as the comparison of different clusterings or partitions. Let us say we have N datasets $X^i = \{X^i_j\}_{j=1}^{n_i} \subset \mathbb{R}^d$, where for each dataset we have a respective partition $\mathcal{C}^i = \{\mathcal{C}^i_k\}_{k=1}^{k_i}$ with $\mathcal{C}^i_k \subset X^i$ and $\mathcal{C}^i_k \cap \mathcal{C}^i_l = \emptyset$. We would like to know how similar are the partitions $\mathcal{C}^i$. One way to do this is the proposal in [Coen et al., 2010]. In this case we take $\Omega = \{x_{\mathcal{C}^1_1}, \dots, x_{\mathcal{C}^1_{k_1}}, \dots, x_{\mathcal{C}^N_1}, \dots, x_{\mathcal{C}^N_{k_N}}\}$ as the space formed by the abstract points $x_{\mathcal{C}^i_j}$ which just serve the purpose of referring to the associated cluster $\mathcal{C}^i_j$. To each partition $\mathcal{C}^i$ there are also associated weights $p^i = \{p^i_1, \dots, p^i_{k_i}\}$. For example, these weights can correspond to the proportion of points in the cluster $\mathcal{C}^i_k$ with respect to the total number of points in $\mathcal{C}^i$. Therefore each clustering $\mathcal{C}^i$ has an associated discrete distribution $\mu^i = \sum_{k=1}^{k_i} p^i_k \delta_{x_{\mathcal{C}^i_k}}$.

On the one hand the following distance is defined

$$d_{OT}(\mathcal{C}^i, \mathcal{C}^j) = C(\mu^i, \mu^j) = \sum_{k=1}^{k_i}\sum_{l=1}^{k_j} w^*_{kl} c(x_{\mathcal{C}^i_k}, x_{\mathcal{C}^j_l}) = \sum_{k=1}^{k_i}\sum_{l=1}^{k_j} w^*_{kl} d(\mathcal{C}^i_k, \mathcal{C}^j_l), \tag{9}$$

where (w^*_{kl}) is defined as in (7). That is, we are solving a discrete optimal transport problem in which the cost between the points associated to the clusters, $c(x_{\mathcal{C}^i_k}, x_{\mathcal{C}^j_l})$, is defined as, $d(\mathcal{C}^i_k, \mathcal{C}^j_l)$, the distance that exists between the respective clusters for a certain metric d. Another auxiliary distance called naïf is defined as

$$d_{NT}(\mathcal{C}^i, \mathcal{C}^j) = \sum_{k=1}^{k_i}\sum_{l=1}^{k_j} p^i_k p^j_l c(x_{\mathcal{C}^i_k}, x_{\mathcal{C}^j_l}) = \sum_{k=1}^{k_i}\sum_{l=1}^{k_j} p^i_k p^j_l d(\mathcal{C}^i_k, \mathcal{C}^j_l). \tag{10}$$

The *similarity distance* is defined as the quotient

$$d_S(\mathcal{C}^i, \mathcal{C}^j) = \frac{d_{OT}(\mathcal{C}^i, \mathcal{C}^j)}{d_{NT}(\mathcal{C}^i, \mathcal{C}^j)}. \tag{11}$$

It is important to note that $0 \leq d_S \leq 1$, where $d_S = 0$ means that the partitions $\mathcal{C}^i, \mathcal{C}^j$ are represented by the same clusters with the same weights and $d_S = 1$ means that each cluster in $\mathcal{C}^i$ is transported proportionally to each of the clusters of $\mathcal{C}^j$. Therefore, values of d_S close to zero indicate a high similarity between the partitions involved and values close to 1 indicate very different partitions.

The similarity distance d_S, with an appropriate definition of distance between clusters, d, will be one of the fundamental tools we will use.

The other fundamental tool are the (2-)Wasserstein k-barycenters. Let us denote by $\mathcal{P}_2(\mathbb{R}^d)$ the set of probability measures in $\mathbb{R}^d$ with second finite moment and let us take in $p = 2$ in (8) and $d^2(x, y) = \|x - y\|^2$ for $x, y \in \mathbb{R}^d$, where $\|x - y\|$ refers to the Euclidean distance. In [del Barrio et al., 2019a] the notion of k-barycenter and trimmed k-barycenter were introduced, building on the Wasserstein barycenter concept introduced in [Agueh and Carlier, 2011, Boissard et al., 2015, Gouic and Loubes, 2017]. The k-barycenter of probabilities $\{\mu_1, \dots, \mu_n\}$ in $\mathcal{P}_2(\mathbb{R}^d)$, with weights $\lambda_1, \dots, \lambda_n$ is any k-set $\{\bar{\mu}_1, \dots, \bar{\mu}_k\}$ in $\mathcal{P}_2(\mathbb{R}^d)$ such that for any $\{\nu_i, \dots, \nu_k\} \subset \mathcal{P}_2(\mathbb{R}^d)$ we have

$$\sum_{i=1}^{n} \lambda_i \min_{j \in \{1,\dots,k\}} \mathcal{W}_2^2(\mu_i, \bar{\mu}_j) \leq \sum_{i=1}^{n} \lambda_i \min_{j \in \{1,\dots,k\}} \mathcal{W}_2^2(\mu_i, \nu_j). \tag{12}$$

An α-trimmed k-barycenter for $\{\mu_1, \ldots, \mu_n\}$ with weights as before is any k-set $\{\bar{\mu}_1, \ldots, \bar{\mu}_k\}$ with weights $\bar{\lambda} = (\bar{\lambda}_1, \ldots, \bar{\lambda}_n) \in \Lambda_\alpha(\lambda)$ such that

$$\sum_{i=1}^{n} \bar{\lambda}_i \min_{j\in\{1,\ldots,k\}} \mathcal{W}_2^2(\mu_i, \bar{\mu}_j) = \min_{\{\nu_1,\ldots,\nu_k\}\subset\mathcal{P}_2(\mathbb{R}^d),\lambda^*\in\Lambda_\alpha(\lambda)} \sum_{i=1}^{n} \lambda_i^* \min_{j\in\{1,\ldots,k\}} \mathcal{W}_2^2(\mu_i, \nu_j), \tag{13}$$

where $\Lambda_\alpha(\lambda) = \{\lambda^* = (\lambda_1^*, \ldots, \lambda_n^*) : 0 \leq \lambda_i^* \leq \lambda_i/(1-\alpha), \sum_{i=1}^n \lambda_i^* = 1\}$.

In a way k-barycentres can be understood as an extension of k-means to the abstract space of probabilities with second finite moment, since we can rewrite (12) as follows

$$\min_{\mathfrak{S}} \sum_{j=1}^{k} \sum_{\mu_i\in\mathfrak{S}_j} \lambda_i \mathcal{W}_2^2(\mu_i, \bar{\mu}_j) \tag{14}$$

where $\mathfrak{S} = \{\mathfrak{S}_1, \ldots, \mathfrak{S}_k\}$ is a partition of $\{\mu_1, \ldots, \mu_n\}$ and $\bar{\mu}_j$ is the barycentre of the elements in $\mathfrak{S}_j$. Also, trimmed k-barycenters can be considered an extension of trimmed k-means. As mentioned in [del Barrio et al., 2019a], for families of location-scale absolutely continuous distributions in $\mathcal{P}_2(\mathbb{R}^d)$ efficient calculations can be performed. A notable example is the family of Gaussian multivariate distributions.

Another important task in the world of clustering, and which will be relevant in our work, is that of *consensus clustering* or *metaclustering*, that is, the search for a partition that optimally summarizes a set of different partitions. A popular way of doing cluster analysis is based on models of mixtures, usually multivariate Gaussians, which essentially assigns to each cluster a distribution that characterizes it. Therefore we are going to be interested in consensus clustering with clustering results based on mixture models.

Suppose, in particular, that we are interested in clustering based on mixture models where distributions are absolutely continuous in $\mathcal{P}_2(\mathbb{R}^d)$ and form a location-scale family . This means that a set of partitions $\mathcal{C}^1, \ldots, \mathcal{C}^N$ can be characterized as $\mathcal{C}^i = \{\mu_j^i\}_{j=1}^{k_i} \subset \mathcal{P}_2(\mathbb{R}^d)$ for $i = 1, \ldots, N$. In these circumstances a way of doing consensus clustering was proposed in [del Barrio et al., 2019a] based on k-barycenters. In particular, we pool together all distributions that characterize the partitions obtaining $\mathcal{C} = \{\mu_1^1, \ldots, \mu_{k_1}^1, \ldots, \mu_1^N, \ldots, \mu_{k_N}^N\} = \{\mu_i\}_{i=1}^{n=k_1+\cdots+k_N}$. We set k, the number of clusters for the consensus (summary) partition and then the k-baycenter gives us $\bar{\mathcal{C}} = \{\bar{\mu}_1, \ldots, \bar{\mu}_k\}$. Precisely $\bar{\mathcal{C}}$ is what we consider to be the consensus partition.

We propose an alternative but related way of consensus clustering. The main advantage is that we do not need to specify, k, the number of groups we want for the summary partition as this assignment is done automatically. The idea is simple, based on $\mathcal{C}$, we build a distances matrix $W_{i,j} = \mathcal{W}_2(\mu_i, \mu_j)$, $i, j = 1, \ldots, n$. Once we have the distances matrix we can use hierarchical clustering to get a partition for $\mathcal{C}$ given by $\mathfrak{S} = \{\mathfrak{S}_1, \ldots, \mathfrak{S}_k\}$. Now, taking the (1-)barycentre of the elements in each $\mathfrak{S}_i$ we get $\bar{\mathcal{C}} = \{\bar{\mu}_1, \ldots, \bar{\mu}_k\}$, which we consider to be the summary partition. There are popular and robust ways of hierarchical clustering where the number of clusters is determined automatically. In particular, we are going to use density-based hierarchical clustering methods through as DBSCAN (see [Ester et al., 1996]) and HDBSCAN (see [Campello et al., 2013]).

Once the basic tools have been presented, we will talk briefly about the methodology we introduce and its application to the world of immunology through flow cytometry. Flow cytometry is based on 'quantitative measurements with a large number of variables

obtained from the study of light scattering and fluorescence properties of hundreds of thousands of individual cells in each analyzed sample' (see [Aghaeepour et al., 2013]). These quantitative measurements allow the analysis and classification of individual cells providing various applications. For example, as mentioned in [Saeys et al., 2016], flow cytometry is used to identify and quantify immune cell populations allowing the monitoring of the immune status of patients or the detection of relevant biomarkers by comparing cytometrics from different groups.

Flow cytometry data have high technical and biological variability. Biological variability is due to intrinsic differences between individuals such as health status, age, gender, etc. Technical variability appears through the use of different experimental adjustments, variation of conditions during experiments or the use of different measuring devices (flow cytometers).

One of the main components of flow cytometry is *gating*, i.e. the assignment of individual cells (an entry in the measurements) to well-established cell types. Among the most commonly used proposals for automatic gating is supervised classification. However, the high variability present in the data obtained by means of flow cytometers makes the effective application of supervised techniques not an easy task.

Note that a classified cytometry, i.e. one that has been subjected to a gating procedure, is equivalent to having a partition in the sense used above. Bearing this in mind, we propose the following methodology for performing supervised classification in flow cytometry. First, we will obtain, through the use of d_S, a partition of a database of classified cytometries $\mathcal{C} = \{\mathcal{C}^1, \ldots, \mathcal{C}^N\}$, which we will denote by $\mathfrak{T} = \{\mathfrak{T}_1, \ldots, \mathfrak{T}_s\}$ where $\mathfrak{T}_i \subseteq \mathcal{C}$. For each $\mathfrak{T}_i$ we want to get a prototype (or summary partition) $\mathcal{T}^i$ that optimally represents the clusterings in $\mathfrak{T}_i$. We will do this using the consensus clustering tools introduced above. Through some unsupervised procedure we will get a partition $\mathcal{C}^u$ for a new cytometry that we want to classify. We will assign $\mathcal{C}^u$ to the prototype $\mathcal{T}^*$ closer to d_S. Finally, we will use the prototype $\mathcal{T}^*$ or the partitions in $\mathfrak{T}_*$ to do supervised classification on the cytometry of interest.

The idea behind the procedure is the following: as the intrinsic variability of the data is high it should be apparent in the database. Therefore it is desirable to group the database into groups whose elements are more homogeneous, resulting in $\mathfrak{T}$. As the partitions within $\mathfrak{T}_i$ would be similar to each other, a consensus of these partitions, given by $\mathcal{T}^i$, should be a good representative to learn from. Again, given the natural variability of flow cytometry, a new cytometry might not resemble some of the prototypes in $\mathcal{T}$, so it is natural to assign this cytometry to the most similar prototype or group of most similar cytometries and only use these for later supervised learning. In case the new cytometry does not resemble any of the prototypes it would not be advisable to use supervised learning. In that case exploratory techniques would be more recommendable.

Precisely the most detailed presentation and practical implementation of this methodology, as well as experiments in real data and comparisons with other state of the art methods is what we propose as the second chapter of this work. In particular, we present the main functionalities of *optimalFlow*, an R package that we have developed to facilitate the use of our methods. We have also made a package, *optimalFlowData*, with the data used for our experiments. Both are available as free software in gitHub and have been submitted to BioConductor.

Attraction-Repulsion clustering

As mentioned previously, cluster analysis or clustering is the task of dividing a set of objects in such a way that elements in the same group or cluster are more similar, according to some dissimilarity measure, than elements in different groups. To achieve this task there are two main types of algorithms: partitioning algorithms, which try to split the data into k groups that usually minimize some optimality criteria, or agglomerative algorithms, which start with single observations and merge them into clusters according to some dissimilarity measure. Such methods have been investigated in a large amount of literature, hence we refer to [Hennig et al., 2015] and references therein for an overview.

Supervised and unsupervised classification procedures are increasingly more influential in people's life since they are used in credit scoring, article recommendation, risk assessment, spam filtering or sentencing recommendations in courts of law, among others. Hence controlling the outcome of such procedures, in particular ensuring that some variables, which should not be taken into account due to moral or legal issues, are not playing a role in the classification of the observations, has become an important field of research known as *fair learning*. We refer to [Lum and Johndrow, 2016, Chouldechova, 2017, Besse et al., 2018] or [Friedler et al., 2018] for an overview of such legal issues and mathematical solutions to address them. The main concern is to detect whether decision rules, learnt from variables X, are biased with respect to a subcategory of the population driven by some variables called protected or sensitive variables. Such variables induce a bias in the observations, and are correlated to other observations. Hence avoiding this effect cannot be achieved by the naive solution of ignoring such protected attributes. Indeed, if the the data at hand reflects a real world bias, machine learning algorithms can pick on this behaviour and emulate it.

Recently, concerns about fairness have received an increasing attention, resulting into two main strategies to address it in the field of classification. One course of action is to transform the data in order to avoid correlation between the set of sensitive attributes and the rest of the data [Feldman et al., 2015, del Barrio et al., 2018]. Another way is to modify the objective functions of the algorithms in a way that eliminates or reduces the unfairness [Zafar et al., 2017, Kehrenberg et al., 2018].

In this work we consider the problem of fair clustering. Suppose we observe data that includes information about attributes that we know or suspect that are biased with respect to the protected class. If the biased variables are dominant enough, a standard clustering on the unprotected data will result in some biased clusters, therefore, if we take some actions based on this partition, we will incur in biased decisions. Ideally, a *fair clustering* would be the situation in which, in the partition of the data, the proportions of the protected attributes are the same in each cluster (hence, the same as the proportions in the whole dataset). This notion of fairness is close to the disparate impact doctrine ([Feldman et al., 2015]) adapted to the clustering setting. The idea is to guarantee that a decision taken with respect to a particular cluster in the partition will not affect disproportionately some group of the population (codified by the protected class), since every group is represented in every cluster according to its proportion in the whole data. With our approach to fair clustering we try to avoid or mitigate these situations by reducing homogeneity with respect to the sensitive classes of the groups, but without imposing too hard fairness constraints such as group proportions that may result in an

excessive reduction of the information conveyed by the data.

A possible approach for fair clustering was presented in [Chierichetti et al., 2017] based on the idea of constrained k-center and k-median clustering. The authors proposed a model where data are partitioned into two groups, codified by red and blue, where disparate impact is avoided by maintaining *balance* between the proportion of points in the two categories. Their goal is to achieve a partition of the data where the respective objective function is minimized while balance in each cluster is maintained over a given treshold. Their approach initially designed for data with two different protected classes has led to the following extensions. The k-center problem was tackled in [Rösner and Schmidt, 2018, Bercea et al., 2018], the k-median in [Backurs et al., 2019] while k-means is further studied in [Bercea et al., 2018, Bera et al., 2019, Schmidt et al., 2018]. These extensions impose constraints related to some minimum and/or maximum value for the proportions of the protected classes in each cluster. Yet, the constraints may result in an excessive modification of the clustering problem with a subsequent loss of geometrical information contained in the data.

In Chapter 3, we propose an alternative to the constrained clustering approach for obtaining fair partitions by incorporating some perturbations in the original dissimilarities of the data in order to favour heterogeneity with respect to the protected variable. For this, we present a new methodology inspired by particle physics and based on *attraction-repulsion dissimilarities*. These new dissimilarities aim at increasing the separation between points with the same values of the protected class and/or decreasing the separation between points with different values of the protected class. Hence they favour the formation of clusters that are more heterogeneous in the protected variable since they reduce the distance between points with different class values, and increase the distance between points with the same class values, leading to a gain in fairness.

Note that we do not need to find *fairlets* ([Chierichetti et al., 2017]), partitions of the data into small fair clusters, or *core-sets* ([Schmidt et al., 2018]), small subsets of the data for which solving the fair clustering problem gives a reasonable approximation to the fair clustering problem in the whole dataset. Our method is more flexible than previous ones and enables to control better the trade-off between fairness and geometrical information of the data. The proposed dissimilarities depend on parameters that the practitioner can control and therefore he or she can impose bigger tendency to fairness. Guidance in the task of choosing the parameters is presented in Section 3.5 and discussed through some synthetic examples in Section 3.6.1. Moreover, *attraction-repulsion dissimilarities* can be combined with some common clustering techniques via an embedding, in particular, with multidimensional scaling (Section 3.2). Agglomerative hierarchical clustering is well suited for the use of dissimilarities, hence, in Section 3.3, we show how to adapt our proposals in a computationally efficient way when using this type of clustering. The proposed attraction-repulsion dissimilarities can also be adapted to the kernel-trick extension, applied to the unprotected variables X, leading to non linear separation in X with a penalization for heterogeneity w.r.t. S as shown in Section 3.4 and 3.6.1.

A stability heuristic for selecting the number of clusters

An essential problem in clustering (cluster analysis) is the determination of the "right" or optimal number of clusters present in the analysed data. This is, and probably will remain, an open problem that has received many different approaches. Among the difficulties we can highlight: the definition of what optimal means, the presence of a huge amount of different clustering procedures, the granularity (zoom in or zoom out) with which we want to analyse the data, etc...

Since determining the number of clusters is crucial, different approaches have been proposed. Extensive discussion can be found in [Hennig et al., 2015] and [Xu et al., 2016]. For example, when using the popular k-means algorithm, the usual procedure for finding the optimal number of cluster is to repeat the k-means procedure for different values of k, the number of clusters. Then, some criteria for selecting the appropriate value of k is used. On the other hand, a fairly simple and used heuristic is to plot intra-cluster variability against the number of clusters k, and select as an appropriate value the "elbow" where making bigger k does not reduce significantly the intra-cluster variability. Some recent interesting proposals when using model-based clustering are given in [Fritz et al., 2013, Cerioli et al.,2018].

An atractive alternative is based on stability. As stated in the Conclusion of Chapter 28 in [Hennig et al., 2015] "over the last decade, a number of resampling schemes have been proposed to detect the correct number of clusters for a given data set or assess the stability of complete partitions or single clusters. For a long time, these methods were computationally too expensive to be used by practitioners on a regular basis in everyday work. With the advent of multicore computers as standard desktops or laptops, cluster model diagnostics by resampling is feasible in acceptable computing time even on standard hardware." Usually, stability is understood as how similar, given a particular similarity criteria, are clusterings of different subsamplings or bootstrapings of the data. In this way a number of clusters that produces stable (similar) partitions is considered to capture relevant structure in the data, and hence, can be considered as the correct or at least more informative number of clusters.

We will be interested precisely in a notion of stability for a clustering methodology known as classification maximum likelihood (cml) (see [Scott and Symons, 1978, McLachlan, 1982]). In particular we will be interested in the restricted version of cml where all clusters are considered to have the same weight. Given a data set $\{x_1, \ldots, x_n\} \in \mathbb{R}^d$, cml looks to maximize

$$\sum_{i=1}^{k} \sum_{x \in \mathcal{C}_i} \log \varphi(x; m_i, S_i) \tag{15}$$

over a partition of the data $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ and the parameters m_i, S_i of the multivariate normal density φ. Clustering based on cml is part of the family of mixture model based clustering procedures, and in particular we will be interested in mixtures of multivariate gaussian distributions. As stated in [Celeux and Govaert, 1993], "loosely speaking, the [well known] mixture approach is aimed to maximize the likelihood over the mixture parameters, whereas the classification approach is aimed to maximize the likelihood over the mixture parameters and over the identifying labels of the mixture component origin

for each point." From another point of view, restricted cml clustering can be viewed as a generalization of k-means for clusters with arbitrary elliptical shape.

An efficient algorithm for doing cml is given by *tclust* introduced in [García-Escudero et al., 2008] (see also equation (2.3)). Using *tclust* with equal weights for all clusters, with no restrictions for the covariance matrices and with no trimming gives a local maximum for the (restricted) cml objective function (15).

Our interest in doing clustering in the cml setting is due to a surprising pattern we have noticed during numerical experiments. It seems that there is a quantifiable instability that appears when using cml for dividing a single dense cluster into k disjoint clusters. In fact, we think that we have captured the behaviour of this instability in Conjecture 4.1. This kind of instability does not appear in a comparable method as k-means as is shown in [Tarpey et al., 1995].

Precisely this unstable behaviour inspires a heuristic for selecting the optimal number of clusters for model based clustering procedures based on the stability of appropriate Wasserstein k-barycenters (see (14) above). We also make use of other well established stability criteria for comparison. In particular, this leads us to suggest that from the point of view of stability it is advisable not to use k-means. The above mentioned heuristic and experiments supporting the previous conjecture are given in Section 4.4 of Chapter 4.

Introducción (Castellano)

Esta tesis ha sido desarrollada en la Universidad de Valladolid y el IMUVA en el marco del proyecto *Técnicas de remuestreo, de recorte, y métricas probabilísticas. Aplicaciones estadísticas* que tiene como investigadores principales a Carlos Matrán Bea y a Eustasio del Barrio Tellado. Entre las líneas de investigación asociadas al proyecto cabe destacar: la validación de modelos, las Distancias de Wasserstein y el Análisis Cluster Robusto. Precisamente el trabajo realizado en estos campos es el que da lugar a los capítulos 1,2 y 4 de esta exposición.

El trabajo realizado en el campo del aprendizaje justo (fair learning) junto al profesor Jean-Michel Loubes, frecuente colaborador con el equipo de Valladolid, durante la estancia internacional en la universidad Paul Sabatier de Toulouse, es la base del Capítulo 3 de esta memoria.

Por lo tanto, esta tesis es una exposición de los problemas y resultados obtenidos en los distintos campos previamente mencionados. Debido a la diversidad de los temas, hemos decidido basar los capítulos en los trabajos publicados o sometidos a fecha de hoy, de manera que cada capítulo cuenta con una estructura relativamente independiente de los demás. De esta manera el Capitulo 1 está basado en los trabajos [del Barrio et al., 2019e, del Barrio et al., 2019d], el Capítulo 2 en el trabajo [del Barrio et al., 2019c], el Capítulo 3 en el trabajo [del Barrio et al., 2019b] y el Capítulo 4 muestra los resultados de un trabajo en progreso.

En esta introducción nuestro objetivo es presentar los principales retos a los que nos hemos enfrentado, así como exponer brevemente nuestros resultados más relevantes. Por otro lado, cada capítulo contará con su propia introducción donde se profundizará en los temas tratados más abajo. Con esto nuestra intención es que el lector tenga una idea general de lo que va a encontrar en cada capítulo y de esta manera cuente con la información necesaria para afrontar las discusiones más técnicas que se encontrará ahí.

Debido a la diversidad de temas tratados en esta memoria proponemos una lectura no lineal. Su- gerimos al lector que tras leer una sección de la Introducción se pase a la lectura del correspondiente capítulo. De esta manera el lector tendrá la información relevante más a mano y podrá seguir mejor la exposición en cada capítulo. Si por otro lado se realiza una lectura secuencial del documento, nos disculpamos de antemano por algunas repeticiones y reiteraciones, que no obstante nos parece que contribuyen positivamente a la comprensión de este trabajo.

Validación aproximada de modelos

Es un hecho bien conocido que los procedimientos clásicos de contraste de bondad de ajuste son excesivamente rígidos para muestras grandes. Esto significa que muy a menudo el modelo será rechazado para conjuntos grandes de datos, aunque la mayoría de las submuestras con tamaños inferiores a cierto tamaño no darían lugar a rechazo. El comportamiento pobre de estos procedimientos ha sido interpretado por algunos autores como un indicador de falsedad de los modelos, hasta el punto de plantear, como en [Liu and Lindsay, 2009], que para cualquier mecanismo generador de datos existe un tamaño muestral a partir del cual el fallo del modelo resulta obvio. Esta dificultad se ha abordado por diferentes autores, coincidiendo en la consideración de la relajación de la hipótesis nula, considerando modelos ampliados, pero igualmente útiles que además suelen comportar una ganancia en robustez [Hodges and Lehmann, 1954, Munk and Czado, 1998, Álvarez-Esteban et al., 2012].

Desde un punto de vista diferente, pero en cierto modo complementario, tenemos la célebre frase de Box "todos los modelos son falsos, pero algunos son muy útiles", que incluso bajo el paradigma de falsedad del modelo nos lleva al interés de considerar alguna medida de hasta qué punto el modelo sería útil o de preferencia respecto a otros. Los trabajos [Davies, 1995, Davies, 2018] inciden en estas cuestiones desde la perspectiva de que un modelo es útil en tanto en cuanto sea capaz de generar muestras similares a los datos obtenidos.

A lo largo de sucesivos proyectos, las propuestas dentro de este esquema de casi-validación o validación aproximada de modelos se encuentran en [Álvarez-Esteban et al., 2008, Álvarez-Esteban et al., 2012]. El enfoque parte de un modelo de contaminación, en el que dos probabilidades, P_1 y P_2, serían similares al nivel α si admitieran una descomposición simultánea:

$$P_1 = (1-\alpha)P_0 + \alpha Q_1, P_2 = (1-\alpha)P_0 + \alpha Q_2 \tag{1}$$

donde P_0, Q_1 y Q_2 son probabilidades arbitrarias. Para valores pequeños de α, esto garantizaría que las muestras obtenidas de estas probabilidades serían prácticamente homogéneas, puesto que la mayor parte de los datos provendrían de la misma P_0.

El modelo de similaridad (1) puede caracterizarse en términos de la distancia en variación total, donde P_1 y P_2 están definidas en la σ-álgebra $\mathfrak{A}$ sobre Ω, como

$$d_{VT}(P_1, P_2) = \sup_{A \in \mathfrak{A}} |P_1(A) - P_2(A)| \leq \alpha,$$

pero, como es bien conocido, esta distancia sólo es apropiada en problemas estadísticos con soporte discreto. Una caracterización más general necesita de algunas herramientas extra como los recortes generalizados y los entornos de contaminación.

Los recortes generalizados de una probabilidad fueron introducidos en [Gordaliza, 1991]. Una probabilidad $\tilde{P} \in \mathbb{R}$ es un recorte de nivel $\alpha \in [0, 1)$ de P cuando existe una función w tal que $0 \leq w \leq 1$ y $\tilde{P}(B) = \frac{1}{1-\alpha} \int_B w(x)P(dx)$ para todo conjunto $B \in \beta$. De manera equivalente, tiene que ser absolutamente continua con respecto a P y con derivada de Radon-Nykodim acotada por $\frac{1}{1-\alpha}$. Denotaremos el conjunto de α-recortes de una distribución de probabilidad P como $R_\alpha(P)$:

$$R_\alpha(P) = \left\{\tilde{P} \in \mathcal{P} : \tilde{P} \ll P, \, \tfrac{d\tilde{P}}{dP} \leq \tfrac{1}{1-\alpha} \, P\text{-a.s.}\right\}. \tag{2}$$

En su trabajo seminal [Huber, 1964], Huber introdujo los entornos de contaminación de una probabilidad, convirtiéndolos en uno de los pilares de la Estadística Robusta. Un (α-) entorno de contaminación de una distribución de probabilidad P_0 es el siguiente conjunto de distribuciones de probabilidad

$$\mathcal{V}_\alpha(P_0) = \{(1-\alpha)P_0 + \alpha Q : Q \in \mathcal{P}\}, \tag{3}$$

donde $\mathcal{P}$ es el conjunto de todas las distribuciones de probabilidad en el espacio. Aunque se pueda definir de manera completamente general, en el primer capítulo $\mathcal{P}$ será el conjunto de probabilidades definidas en los conjuntos (de Borel), β, de la recta real $\mathbb{R}$. Si F y F_0 son funciones de distribución, usaremos $R_\alpha(F)$ y $\mathcal{V}_\alpha(F_0)$, con los mismos significados que antes, pero definidos en termino de las funciones de distribución.

De esta manera, dado un modelo "ideal" P_0, el entorno incluye aquellas distribuciones de probabilidad que son producto de una distorsión por errores de bulto o de redondeo: dado un valor $\alpha \in [0, 1)$, una probabilidad P de $\mathcal{V}_\alpha(P_0)$ generaría muestras con aproximadamente $(1-\alpha)\times 100$ porciento de los datos provenientes de P_0. Alternativamente, una muestra así podría ser convenientemente "recortada" para obtener una autentica muestra del modelo. De hecho, incluso P_0 se podría obtener a partir de un recorte adecuado de P.

Un hecho fundamental es que los entornos de contaminación se relacionan con los recortes (ver [Álvarez-Esteban et al., 2011]) mediante

$$P \in \mathcal{V}_\alpha(P_0) \Longleftrightarrow P_0 \in R_\alpha(P). \tag{4}$$

De esta manera, el modelo (1) se puede expresar como $P_1, P_2 \in \mathcal{V}_\alpha(P_0)$ o de manera equivalente para una distancia apropiada en el espacio de probabilidades

$$0 \leq d(R_\alpha(P_1), R_\alpha(P_2)) = \inf_{\tilde{P}\in R_\alpha(P_1), \tilde{Q}\in R_\alpha(P_2)} d(\tilde{P}, \tilde{Q}) \leq d(P_0, P_0) = 0.$$

Un caso particular se da cuando queremos comprobar si $P \in \mathcal{V}_\alpha(P_0)$ o de manera equivalente ver que $d(P_0, R_\alpha(P)) = 0$.

El problema con estas caracterizaciones es que desconocemos tanto α como la verdadera distribución contaminada P. En la realidad se suele disponer de una aproximación $\hat{P}$ a P; normalmente $\hat{P}$ es la distribución empírica, y nuestro objetivo es buscar evidencia estadística, basada en $\hat{P}$ a favor o en contra de la hipótesis $P \in \mathcal{V}_\alpha(P_0)$. Para esta tarea recurrimos a la métrica, d, en el espacio $\mathcal{P}$ y consideramos $d(P_0, R_\alpha(\hat{P}))$ como un estimador de $d(P_0, R_\alpha(P))$.

La distancia L_2 de Wasserstein

$$\mathcal{W}_2^2(P_1, P_2) = \inf_{\pi\in\Pi(P_1,P_2)} \int \|x-y\|^2 d\pi(x,y) = \inf\left\{E\|X-Y\|^2, \mathcal{L}(X) = P_1, \mathcal{L}(Y) = P_2\right\},$$

donde $\Pi(P_1, P_2)$ es el conjunto de probabilidades en $\Omega \times \Omega$ con primera marginal P_1 y segunda marginal P_2, es la elección de métrica usada en [Álvarez-Esteban et al., 2011].

Nuestra propuesta parte de los mismos principios, pero recurriendo ahora a la distancia de Kolmogorov (o L_∞) entre las funciones de distribución, de manera explicita,

$$d_K(F, G) = \sup_{x\in\mathbb{R}} |F(x) - G(x)|, \tag{5}$$

(usaremos de manera intercambiable la notación $\|F - G\|$ y $d_K(F, G)$). Al fin y al cabo, el test de Kolmogorov-Smirnov es probablemente el más conocido y utilizado entre los contrastes de bondad de ajuste, por lo que supone un marco privilegiado para desarrollar estas ideas.

En particular hemos estudiado las propiedades del estimador plug-in $d_k(F_0, R_\alpha(F_n))$, donde F_n es la función de distribución empírica basada en una muestra de n variable aleatorias independientes con distribución común F (Proposición 1.2) . Además hemos desarrollado un algoritmo para el computo eficiente del estimador (Teorema 1.4). Hemos desarrollado un test con probabilidades de error exponencialmente pequeñas, basado en el estadístico anterior, para contrastar $H_0 : d_k(F_0, R_\alpha(F)) = 0$ contra la alternativa $d_k(F_0, R_\alpha(F)) > \rho$ (Teorema 1.6). Como contribución principal destacamos el Teorema 1.9, un teorema central del límite donde demostramos que $\sqrt{n}(d_K(F_0, R_\alpha(F_n)) - d_K(F_0, R_\alpha(F))$ converge al supremo de un proceso Gaussiano.

Como aplicación de nuestros resultados teóricos cabe destacar nuestro estimador para $\alpha^* = \min\{\alpha : F \in \mathcal{V}_\alpha(F_0)\}$, el mínimo nivel de contaminación que permitiría incluir la distribución subyacente en el modelo original, que es mas general que el estimador propuesto en [Rudas et al., 1994] ya que no se limita a modelos multinomiales. Asimismo, ofrecemos intervalos de confianza para el tamaño de las submuestras que serían compatibles con el modelo, extendiendo resultados de aproximación puntual presentados en [Lindsay and Liu, 2009].

Transporte óptimo aplicado a la citometría de flujo

Recientes avances en torno al problema de transporte óptimo han resultado en aplicaciones relevantes en el análisis de resultados de procedimientos cluster o de agrupación. Los procedimientos cluster buscan dividir una serie de datos en grupos disjuntos (clusters) produciendo así lo que se denomina como partición o clustering de los datos. Los procedimientos de análisis se encargan de comparar distintas particiones y de intentar producir particiones resumen que de alguna manera óptima representen a un conjunto de particiones (consensus clustering). En particular nuestro interés se va a centrar en procedimientos basados en transporte óptimo, por lo que presentamos brevemente algunos de los resultados fundamentales que usaremos más adelante.

Siguiendo [Villani, 2009], sea $\mathcal{P}(\Omega)$, el espacio de probabilidades en Ω. Para μ, ν en $\mathcal{P}(\Omega)$, sea $\Pi(\mu, \nu)$ el conjunto de probabilidades π en $\Omega \times \Omega$ con primera marginal μ y segunda marginal ν. El coste de transporte óptimo entre las dos medidas se define como

$$C(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int c(x, y) d\pi(x, y) \tag{6}$$

donde $c(x, y)$ es el coste de transportar una unidad de masa desde x a y. Una probabilidad π que alcanza el mínimo en (6) se denomina emparejamiento óptimo, y tiene asociada una variable aleatoria (X, Y) con distribución conjunta π. Cuando μ y ν son discretas, es decir, $\mu = \sum_{k=1}^{n} p_k \delta_{x_k}$ y $\nu = \sum_{l=1}^{m} q_l \delta_{y_l}$, con $x_k, y_l \in \Omega$, el problema de transporte óptimo se puede plantear como el sigueinte problema de programación lineal (vease [Bertsimas and Tsitsiklis, 1997])

$$C(\mu, \nu) = \sum_{k=1}^{n} \sum_{l=1}^{m} w^*_{kl} c(x_k, y_l), \tag{7}$$

donde (w_{kl}^*) son las soluciones al programa lineal de transporte óptimo

$$\begin{array}{lll} \text{minimizar} & \sum_{k=1}^{n}\sum_{l=1}^{m} w_{kl}c(x_k, y_l) & \\ \text{sujeto a} & w_{kl} \geq 0, & 1 \leq k \leq n, 1 \leq l \leq m \\ & \sum_{l=1}^{m} w_{kl} = p_k, & 1 \leq k \leq n \\ & \sum_{k=1}^{n} w_{kl} = q_l, & 1 \leq l \leq m \\ & \sum_{k=1}^{n}\sum_{l=1}^{m} w_{kl} = 1. & \end{array}$$

Para (Ω, d) un espacio métrico polaco y $p \in [1, \infty)$, la distancia $p-$Wasserstein entre μ y ν se define como

$$\mathcal{W}_p^p(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int d^p(x, y) d\pi(x, y) = \inf \left\{ E d^p(X, Y), \mathcal{L}(X) = \mu, \mathcal{L}(Y) = \nu \right\}, \quad (8)$$

donde $\mathcal{L}(X)$ es la ley de X.

Haremos uso de la distancia de Wassserstein en una tarea importante en el análisis cluster como la comparación de distintos clusterings o particiones. Digamos que tenemos N conjuntos de datos $X^i = \{X_j^i\}_{j=1}^{n_i} \subset \mathbb{R}^d$, donde para cada conjunto de datos tenemos la respectiva partición $\mathcal{C}^i = \{\mathcal{C}_k^i\}_{k=1}^{k_i}$ con $\mathcal{C}_k^i \subset X^i$ y $\mathcal{C}_k^i \cap \mathcal{C}_l^i = \emptyset$. Nos gustaría saber como de parecidas son las particiones $\mathcal{C}^i$. Una manera de hacerlo es la propuesta en [Coen et al., 2010]. En este caso tomamos $\Omega = \{x_{\mathcal{C}_1^1}, \ldots, x_{\mathcal{C}_{k_1}^1}, \ldots, x_{\mathcal{C}_1^N}, \ldots, x_{\mathcal{C}_{k_N}^N}\}$ formado por los puntos abstractos $x_{\mathcal{C}_j^i}$ que sirven para referirnos al cluster $\mathcal{C}_j^i$. Además, a cada partición, $\mathcal{C}^i$ se le asocian unos pesos $p^i = \{p_1^i, \ldots, p_{k_i}^i\}$. Por ejemplo, estos pesos pueden corresponderse con la proporción de puntos en el cluter $\mathcal{C}_k^i$ con respecto al número de puntos totales en $\mathcal{C}^i$. De esta manera a cada partición (clustering) $\mathcal{C}^i$ le corresponde una distribución discreta $\mu^i = \sum_{k=1}^{k_i} p_k^i \delta_{x_{\mathcal{C}_k^i}}$.

Por un lado se define la distancia

$$d_{OT}(\mathcal{C}^i, \mathcal{C}^j) = C(\mu^i, \mu^j) = \sum_{k=1}^{k_i}\sum_{l=1}^{k_j} w_{kl}^* c(x_{\mathcal{C}_k^i}, x_{\mathcal{C}_l^j}) = \sum_{k=1}^{k_i}\sum_{l=1}^{k_j} w_{kl}^* d(\mathcal{C}_k^i, \mathcal{C}_l^j), \quad (9)$$

donde (w_{kl}^*) se define como en (7). Es decir estamos resolviendo un problema de transporte óptimo discreto en el cual el coste entre los puntos asociados a los clusters, $c(x_{\mathcal{C}_k^i}, x_{\mathcal{C}_l^j})$, se define como, $d(\mathcal{C}_k^i, \mathcal{C}_l^j)$, la distancia que existe entre los respectivos clusters para cierta métrica d. Otra distancia auxiliar denominada naïf se define como

$$d_{NT}(\mathcal{C}^i, \mathcal{C}^j) = \sum_{k=1}^{k_i}\sum_{l=1}^{k_j} p_k^i p_l^j c(x_{\mathcal{C}_k^i}, x_{\mathcal{C}_l^j}) = \sum_{k=1}^{k_i}\sum_{l=1}^{k_j} p_k^i p_l^j d(\mathcal{C}_k^i, \mathcal{C}_l^j). \quad (10)$$

La *distancia de similaridad* se define como el cociente

$$d_S(\mathcal{C}^i, \mathcal{C}^j) = \frac{d_{OT}(\mathcal{C}^i, \mathcal{C}^j)}{d_{NT}(\mathcal{C}^i, \mathcal{C}^j)}. \quad (11)$$

Es importante notar que $0 \leq d_S \leq 1$, donde $d_S = 0$ quiere decir que las particiones $\mathcal{C}^i, \mathcal{C}^j$ están re-presentadas por los mismos clusters con los mismos pesos y $d_S = 1$ quiere decir que cada cluster en $\mathcal{C}^i$ es transportado de manera proporcional a cada uno de los clusters

de $\mathcal{C}^j$. Por lo tanto, valores de d_S cercanos a cero indican una alta similaridad entre las particiones involucradas y valores cercanos a 1 indican particiones muy diferentes.

La distancia de similaridad d_S, con una definición de distancia entre clusters, d, apropiada, será una de las herramientas fundamentales que usaremos.

La otra herramienta fundamental son los k-baricentros de Wasserstein. Vamos a denotar por $\mathcal{P}_2(\mathbb{R}^d)$ el conjunto de medidas de probabilidad en $\mathbb{R}^d$ con segundo momento finito y tomaremos $p = 2$ en (8) y $d^2(x, y) = \|x - y\|^2$ para $x, y \in \mathbb{R}^d$ y $\|x - y\|$ la distancia Euclidea. En [del Barrio et al., 2019a] la noción de k-baricentro y de k-baricentro recortado fueron introducidas, construyendo sobre el concepto de baricentro de Wasserstein introducido en [Agueh and Carlier, 2011, Boissard et al., 2015, Gouic and Loubes, 2017]. El k-baricentro de las probabilidades $\{\mu_1, \ldots, \mu_n\}$ en $\mathcal{P}_2(\mathbb{R}^d)$ con pesos $\lambda_1, \ldots, \lambda_n$ es cualquier k-conjunto $\{\bar{\mu}_1, \ldots, \bar{\mu}_k\}$ en $\mathcal{P}_2(\mathbb{R}^d)$ tal que para cualquier $\{\nu_i, \ldots, \nu_k\} \subset \mathcal{P}_2(\mathbb{R}^d)$ tenemos que

$$\sum_{i=1}^{n} \lambda_i \min_{j\in\{1,\ldots,k\}} \mathcal{W}_2^2(\mu_i, \bar{\mu}_j) \leq \sum_{i=1}^{n} \lambda_i \min_{j\in\{1,\ldots,k\}} \mathcal{W}_2^2(\mu_i, \nu_j). \tag{12}$$

Un k-baricentro α-recortado de $\{\mu_1, \ldots, \mu_n\}$ con pesos como antes es cualquier k-conjunto $\{\bar{\mu}_1, \ldots, \bar{\mu}_k\}$ con pesos $\bar{\lambda} = (\bar{\lambda}_1, \ldots, \bar{\lambda}_n) \in \Lambda_\alpha(\lambda)$ tal que

$$\sum_{i=1}^{n} \bar{\lambda}_i \min_{j\in\{1,\ldots,k\}} \mathcal{W}_2^2(\mu_i, \bar{\mu}_j) = \min_{\{\nu_1,\ldots,\nu_k\}\subset\mathcal{P}_2(\mathbb{R}^d), \lambda^*\in\Lambda_\alpha(\lambda)} \sum_{i=1}^{n} \lambda_i^* \min_{j\in\{1,\ldots,k\}} \mathcal{W}_2^2(\mu_i, \nu_j), \tag{13}$$

donde $\Lambda_\alpha(\lambda) = \{\lambda^* = (\lambda_1^*, \ldots, \lambda_n^*) : 0 \leq \lambda_i^* \leq \lambda_i/(1-\alpha), \sum_{i=1}^n \lambda_i^* = 1\}$.

En cierta manera los k-baricentros se puede entender como una extensión de las k-medias al espacio abstracto de las probabilidades con segundo momento finito, ya que podemos reescribir (12) como

$$\min_{\mathfrak{S}} \sum_{j=1}^{k} \sum_{\mu_i\in\mathfrak{S}_j} \lambda_i \mathcal{W}_2^2(\mu_i, \bar{\mu}_j) \tag{14}$$

donde $\mathfrak{S} = \{\mathfrak{S}_1, \ldots, \mathfrak{S}_k\}$ es una partición de $\{\mu_1, \ldots, \mu_n\}$ y $\bar{\mu}_j$ es el baricentro de los elementos en $\mathfrak{S}_j$. Asimismo, los k-barycenters recortados se pueden considerar una extension de las k-medias recortadas. Como se menciona en [del Barrio et al., 2019a], para familias de localización-escala de distribuciones absolutamente continuas en $\mathcal{P}_2(\mathbb{R}^d)$ se pueden realizar cálculos eficientes. Como ejemplo notable cabe destacar a la familia de distribuciones multivariantes Gaussianas.

Otra tarea importante en el mundo del clustering, y que será relevante en nuestro trabajo, es la del *consensus clustering* o *metaclustering*, es decir, la búsqueda de una partición que resume de manera óptima un conjunto de particiones diferentes. Una forma popular de hacer análisis cluster es la basada en modelos de mezclas, habitualmente multivariantes Gaussianas, que esencialmente asigna a cada cluster una distribución que lo caracteriza. Por lo tanto vamos a estar interesados en hacer consensus clustering con resultados de clustering basado en modelo de mezclas.

Supongamos, en particular, que estamos interesados en el clustering basado en modelo de mezclas donde las distribuciones son una familia de localización-escala absolutamente continuas en $\mathcal{P}_2(\mathbb{R}^d)$. Esto quiere decir que un conjunto de particiones $\mathcal{C}^1, \ldots, \mathcal{C}^N$ se puede

caracterizar como $\mathcal{C}^i = \{\mu_j^i\}_{j=1}^{k_i} \subset \mathcal{P}_2(\mathbb{R}^d)$ para $i = 1, \ldots, N$. En estas circunstancias una manera de hacer consensus clustering fue propuesta en [del Barrio et al., 2019a] basada en el k-baricentro. En particular, agregamos todas las distribuciones que caracterizan las particiones obteniendo $\mathcal{C} = \{\mu_1^1, \ldots, \mu_{k_1}^1, \ldots, \mu_1^N, \ldots, \mu_{k_N}^N\} = \{\mu_i\}_{i=1}^{n=k_1+\cdots+k_N}$. Fijamos k, el número de clusters que que queremos que tenga la partición consenso (resumen) y entonces el k-baricentro nos da $\bar{\mathcal{C}} = \{\bar{\mu}_1, \ldots, \bar{\mu}_k\}$. Precisamente $\bar{\mathcal{C}}$ es lo que consideramos como la partición de consenso.

Nosotros proponemos una manera alternativa pero relacionada de hacer consensus clustering. La principal ventaja es que no necesitamos especificar el numero k de grupos que queremos que tenga la partición resumen ya que esta asignación se hace de forma automática. La idea es sencilla, basándonos en $\mathcal{C}$, construimos una matriz de distancias $W_{i,j} = \mathcal{W}_2(\mu_i, \mu_j)$, $i, j = 1, \ldots, n$. Una vez que tenemos la matriz de distancia podemos usar clustering jerárquico para obtener una partición $\mathfrak{S} = \{\mathfrak{S}_1, \ldots, \mathfrak{S}_k\}$ de $\mathcal{C}$. Ahora, tomando el (1-)baricentro de los elementos en cada $\mathfrak{S}_i$ obtenemos $\bar{\mathcal{C}} = \{\bar{\mu}_1, \ldots, \bar{\mu}_k\}$, que consideramos como la partición resumen. Hay maneras populares y robustas de hacer clustering jerárquico donde el número de clusters se determina de forma automática. En particular, vamos a usar métodos de clustering jerárquico basado en densidades a través de DBSCAN (ver [Ester et al., 1996]) y HDBSCAN (ver [Campello et al., 2013]).

Una vez presentadas las herramientas básicas, vamos a hablar brevemente de la metodología que introducimos y de su aplicación al mundo de la inmunología a través de la citometría de flujo. La citometría de flujo se basa en 'medidas cuantitativas con un gran número de variables provenientes del estudio de la dispersión de luz y de propiedades de fluorescencia de cientos de miles de células individuales en cada muestra analizada' (vease [Aghaeepour et al., 2013]). Estas medidas cuantitativas permiten el análisis y clasificación de células individuales proporcionando diversas aplicaciones. Por ejemplo, como se menciona en [Saeys et al., 2016], la citometría de flujo se usa para identificar y cuantificar poblaciones de células inmunes lo que permite monitorizar el estado inmune de los pacientes o la detección de biomarcadores relevantes mediante la comparación de citometrías de diferentes grupos.

Los datos procedentes de la citometría de flujo tienen una alta variabilidad técnica y biológica. La variabilidad biológica se debe a las diferencias intrínsecas entre individuos como el estado de salud, la edad, el género, etc...La variabilidad técnica aparece por el uso de distintos ajustes experimentales, por la variación de las condiciones durante los experimentos o por el uso de distintos aparatos de medición (citómetros de flujo).

Uno de los componentes principales de la citometría de flujo es el *gating*, es decir la asignación de células individuales (una entrada en las mediciones) a tipos de células bien establecidos. Entre las propuestas más usadas para realizar gating de manera automática se encuentra la clasificación supervisada. Sin embargo la alta variabilidad presente en los datos obtenidos mediante los citometros de flujo hace que la aplicación eficaz de técnicas supervisadas no sea una tarea sencilla.

Nótese que una citometría clasificada, es decir que ha sido sometida a un proceso de gating, es equivalente a tener una partición en el sentido usado más arriba. Teniendo en cuenta esto, proponemos la siguiente metodología para realizar clasificación supervisada en citometrías de flujo. Primero, obtendremos, mediante el uso de d_S, una partición de una base de datos de citometrías clasificadas $\mathcal{C} = \{\mathcal{C}^1, \ldots, \mathcal{C}^N\}$, que denotaremos por $\mathfrak{T} = \{\mathfrak{T}_1, \ldots, \mathfrak{T}_s\}$ donde $\mathfrak{T}_i \subseteq \mathcal{C}$. Para cada $\mathfrak{T}_i$ queremos obtener un prototipo (o par-

tición resumen) $\mathcal{T}^i$ que represente de manera óptima los clusterings en $\mathfrak{T}_i$. Haremos esto usando las herramientas de consensus clustering introducidas más arriba. Mediante algún procedimiento no supervisado obtendremos una partición $\mathcal{C}^u$ de una nueva citometría a la que queremos clasificar. Asignaremos $\mathcal{C}^u$ al prototipo $\mathcal{T}^*$ mas cercano con respecto a d_S. Por último, usaremos el prototipo $\mathcal{T}^*$ o las particiones en $\mathfrak{T}_*$ para hacer clasificación supervisada sobre la citometría de interés.

La idea detrás del procedimiento es la siguiente: como la variabilidad intrínseca de los datos es elevada esto se debería notar en la base de datos. Por lo tanto es deseable agrupar la base de datos en grupos cuyos elementos sean más homogéneos, dando lugar a $\mathfrak{T}$. Como las particiones dentro de $\mathfrak{T}_i$ serían parecidas entre si, un consenso de estas particiones, dado por $\mathcal{T}^i$, debería ser buen represente para aprender sobre él. Otra vez, dada la variabilidad natural de las citometrías de flujo, una nueva citometría podría no parecerse a algunos de los prototipos de $\mathcal{T}$, por lo que es natural asignar esta citometría al prototipo más similar o al grupo de citometrías más similares y usar solamente estos para el posterior aprendizaje supervisado. En caso de que la nueva citometría no se pareciera a ninguno de los prototipos no sería recomendable usar aprendizaje supervisado. En ese caso técnicas exploratorias serían más recomendables.

Precisamente la presentación más detallada y la implementación práctica de esta metodología, así como experimentos en datos reales y comparaciones con otros métodos de última generación es lo que proponemos como segundo capítulo de este trabajo. En particular, presentamos las principales funcionalidades de *optimalFlow*, un paquete de R que hemos desarrollado para facilitar el uso de nuestros métodos. Asimismo, hemos hecho un paquete, *optimalFlowData*, con los datos usados para nuestros experimentos. Ambos están disponibles como software libre en gitHub y han sido sometidos a BioConductor.

Clustering de atracción-repulsión

Como se mencionó anteriormente, el análisis cluster o clustering es la tarea de dividir un conjunto de objetos de tal manera que los elementos en el mismo grupo o cluster son más similares, de acuerdo con alguna medida de disimilaridad, que los elementos en diferentes grupos. Esto se puede conseguir principalmente mediante dos tipos de algoritmos: los algoritmos de partición, que tratan de dividir los datos en k grupos que habitualmente minimizan algunos criterios de optimización, o los algoritmos aglomerativos, que comienzan con observaciones individuales y las fusionan en clusters de acuerdo con alguna medida de disimilitud. Estos métodos han sido investigados ampliamente en la literatura, por lo que nos referimos a [Hennig et al., 2015] y sus referencias para una visión general.

Los procedimientos de clasificación supervisada y no supervisada son cada vez más influyentes en la vida cotidiana, ya que se utilizan en el scoring crediticio, la recomendación de artículos, la evaluación de riesgos, el filtrado de spam o las recomendaciones de sentencia en los tribunales de justicia, entre otros. Por lo tanto, el control del resultado de estos procedimientos, en particular asegurar que algunas variables, que no deben tenerse en cuenta debido a cuestiones morales o jurídicas, no desempeñan un papel en la clasificación de las observaciones, se ha convertido en un importante campo de investigación conocido como *fair learning*. Nos referimos a [Lum and Johndrow, 2016, Chouldechova, 2017, Besse et al., 2018] o [Friedler et al., 2018] para obtener una visión general de los

problemas legales y de las soluciones matemáticas para resolverlos. La tarea fundamental es detectar si las reglas de decisión, aprendidas de las variables X, están sesgadas con respecto a una subcategoría de la población impulsada por algunas variables llamadas variables protegidas o sensibles. Tales variables inducen un sesgo en las observaciones y están correlacionadas con otras observaciones. Por lo tanto, evitar este efecto no puede lograrse con la solución naïf de ignorar tales atributos protegidos. De hecho, si los datos en cuestión reflejan un sesgo del mundo real, los algoritmos de aprendizaje automático pueden captar este comportamiento y emularlo.

Recientemente, la preocupación por la equidad o justicía ha recibido una atención cada vez mayor, lo que ha dado lugar a dos estrategias principales para abordarla en el ámbito de la clasificación. Una manera es transformar los datos para evitar la correlación entre el conjunto de atributos sensibles y el resto de los datos [Feldman et al., 2015, del Barrio et al., 2018]. Otra forma es modificar las funciones objetivo de los algoritmos de manera que se elimine o reduzca la injusticia [Zafar et al., 2017, Kehrenberg et al., 2018].

En este trabajo consideramos el problema del clustering justo. Supongamos que observamos datos que incluyen información sobre atributos que conocemos o sospechamos que están sesgados con respecto a la clase protegida. Si las variables sesgadas son lo suficientemente dominantes, una agrupación (clustering) estándar en los datos desprotegidos resultará en algunos clusters (grupos) sesgadas, por lo tanto, si tomamos acciones basadas en esta partición, incurriremos en decisiones sesgadas. Idealmente, un *clustering justo* sería aquel en el que, en la partición de los datos, las proporciones de los atributos protegidos fueran las mismas en cada cluster (por lo tanto, las mismas que las proporciones en todo el conjunto de datos). Esta noción de equidad se acerca a la doctrina del impacto dispar ([Feldman et al., 2015]) adaptada al problema de clustering. La idea es garantizar que una decisión tomada con respecto a un determinado cluster en la partición no afecte desproporcionadamente a algún grupo de la población (codificado por la clase protegida), ya que cada grupo está representado en cada cluster según su proporción en el conjunto total de los datos. Con nuestro enfoque de agrupación justa tratamos de evitar o mitigar estas situaciones reduciendo la homogeneidad con respecto a las clases sensibles de los grupos, pero sin imponer restricciones de equidad demasiado fuertes, como en las proporciones de los grupos, que pueden resultar en una reducción excesiva de la información transmitida por los datos.

En [Chierichetti et al., 2017] se presentó un posible enfoque al clustering justo basado en la idea de clustering restringido basado en k-centros y k-medianas. Los autores propusieron un modelo en el que los datos se dividen en dos grupos, codificados en rojo y azul, y en el que se evita el impacto desigual manteniendo un balance entre la proporción de puntos en las dos categorías. Su objetivo es lograr una partición de los datos en la que se minimice la función objetivo respectiva mientras se mantiene el balance en cada cluster por debajo de un umbral determinado. Su enfoque inicialmente diseñado para datos con dos clases protegidas diferentes ha llevado a las siguientes extensiones. El problema del k-centro se abordó en [Rösner and Schmidt, 2018, Bercea et al., 2018], el de la k-mediana en [Backurs et al., 2019] mientras que las k-medias se estudia más a fondo en [Bercea et al., 2018, Bera et al., 2019, Schmidt et al., 2018]. Estas extensiones imponen restricciones relacionadas con algún valor mínimo y/o máximo para las proporciones de las clases protegidas en cada cluster. Sin embargo, las restricciones pueden dar lugar a una modificación excesiva del problema de agrupamiento con la consiguiente pérdida de

información geométrica contenida en los datos.

A continuación, proponemos una alternativa al enfoque de clustering restringido para la obtención de particiones justas, incorporando algunas perturbaciones en las disimilaridades originales de los datos para favorecer la heterogeneidad respecto a la variable protegida. Para ello, presentamos una nueva metodología inspirada en la física de partículas y basada en *disimilaridades de atracción-repulsión*. Estas nuevas disimilaridades tienen como objetivo aumentar la separación entre puntos con los mismos valores de la clase protegida y/o disminuir la separación entre puntos con diferentes valores de la clase protegida. Por lo tanto, favorecen la formación de conglomerados más heterogéneos en la variable protegida, ya que reducen la distancia entre puntos con diferentes valores de clase y aumentan la distancia entre puntos con los mismos valores de clase, lo que conduce a un aumento de la justicia.

Es importante notar que con nuestros procedimientos no necesitamos encontrar *fairlets* ([Chierichetti et al., 2017]), particiones de los datos en pequeños clusters justos, o *core-sets* ([Schmidt et al., 2018]), pequeños subconjuntos de datos para los cuales la resolución del problema de clustering justo da una aproximación razonable al problema de clustering justo en todo el conjunto de datos. Nuestro método es más flexible que los anteriores y permite controlar mejor el equilibrio entre la equidad y la información geométrica de los datos. Las disimilaridades propuestas dependen de parámetros que el usuario puede controlar y por lo tanto puede imponer una mayor tendencia a la equidad. Indicaciones sobre como elegir los parámetros libres se dan en la Sección 3.5 y se discuten a través de algunos ejemplos sintéticos en la Sección 3.6.1. Además, las disimilaridades de atracción-repulsión pueden combinarse con algunas técnicas comunes de clustering a través de un embedding, en particular, con un escalado multidimensional (Sección 3.2). El clustering jerárquico es adecuada para el uso de disimilaridades, por lo tanto, en la Sección 3.3, mostramos cómo adaptar nuestras propuestas de una manera computacionalmente eficiente cuando se utiliza este tipo de agrupación. Las disimilaridades de atracción-repulsión también pueden adaptarse a la extensión kernel-trick, aplicada a las variables desprotegidas X, lo que lleva a una separación no lineal en X con una penalización por heterogeneidad con respecto a S como se muestra en la Sección 3.4 y 3.6.1.

Heurística basada en estabilidad para la selección del número de clusters

Un problema esencial en el clustering (análisis cluster) es la determinación del número "correcto" u óptimo de clusters (grupos) presentes en los datos analizados. Este es, y probablemente seguirá siendo, un problema abierto que ha recibido muchos enfoques diferentes. Entre las dificultades podemos destacar: la definición de lo que significa óptimo, la presencia de una gran cantidad de procedimientos de clustering diferentes, la granularidad (zoom in o zoom out) con la que queremos analizar los datos, etc...

Debido a que la determinación del número de clusters es crucial se han propuesto diferentes enfoques para resolver el problema. Se puede encontrar una extensa discusión en [Hennig et al., 2015] y [Xu et al., 2016]. Por ejemplo, cuando se utiliza el popular algoritmo de k-medias, la técnica habitual para encontrar el número óptimo de clusters es repetir el procedimiento de k-medias para diferentes valores de k, el número de clus-

ters. Posteriormente se utilizan algunos criterios para seleccionar el valor apropiado de k. Por otro lado, una heurística simple y bastante usada es representar gráficamente la variabilidad intra-cluster contra el número de clusters k, y seleccionar como valor apropiado el "codo" donde hacer más grande k no reduce significativamente la variabilidad intra-cluster. Algunas propuestas interesantes y recientes sobre el usando clustering basada en modelos se presentan en [Fritz et al., 2013, Cerioli et al.,2018].

Una alternativa atractiva se basa en la estabilidad. Como se indica en la Conclusión del Capítulo 28 en [Hennig et al., 2015] "durante la última década, se han propuesto varios esquemas de remuestreo para detectar el número correcto de clusters para un conjunto de datos dado o para evaluar la estabilidad de particiones completas o clusters individuales. Durante mucho tiempo, estos métodos fueron computacionalmente demasiado costosos para ser utilizados de forma regular en el trabajo diario. Con la llegada de los ordenadores multinúcleo como los ordenadores de sobremesa o los portátiles estándar, el diagnóstico de procedimientos de agrupación mediante remuestreo es factible en un tiempo de cálculo aceptable, incluso en hardware estándar". Por lo general, la estabilidad se entiende como lo similares que son, dado un criterio particular de similitud, los agrupamientos de diferentes submuestras o bootstrapings de los datos. De esta manera, se considera que un número de clústeres que produce particiones estables (similares) captura la estructura relevante en los datos, y por lo tanto, puede ser considerado como el número correcto o al menos más informativo de clústeres.

Nos interesará precisamente una noción de estabilidad para una metodología de clustering conocida como clasificación de máxima verosimilitud (cml de su siglas en inglés) (ver [Scott and Symons, 1978, McLachlan, 1982]). En particular, nos interesará la versión restringida de cml, en la que se considera que todos los clusters tienen el mismo peso. Dada una colección de datos, el cml trata de maximizar la función

$$\sum_{i=1}^{k} \sum_{x \in \mathcal{C}_i} \log \varphi(x; m_i, S_i) \tag{15}$$

sobre la partición de los datos $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ y los parámetros m_i, S_i de la densidad normal multivariante φ. El clustering basado en cml forma parte de la familia de procedimientos de clustering basados en modelos de mezclas y, en particular, nos interesarán las mezclas de distribuciones Gaussianas multivariantes. Como se indica en [Celeux and Govaert, 1993], "en términos generales, el enfoque de mezclas [bien conocido] tiene por objeto maximizar la probabilidad sobre los parámetros de la mezcla, mientras que el enfoque de clasificación tiene por objeto maximizar la probabilidad sobre los parámetros de la mezcla y sobre las etiquetas de identificación del origen del componente de la mezcla para cada punto". Desde otro punto de vista, el clustering cml restringido puede ser entendido como una generalización del k-medias para clusters con forma elíptica arbitraria.

Un algoritmo eficiente para hacer clustering cml está dado por *tclust* introducido en [García-Escudero et al., 2008] (ver también ecuación (2.3)). Usando *tclust* con pesos iguales para todos los clusters, sin restricciones para las matrices de covarianza y sin recorte, obtenemos un máximo local para la función objetivo cml (restringida) (15).

Nuestro interés por hacer clustering basado en cml se debe a un patrón sorprendente que hemos observado durante los experimentos numéricos que hemos realizado. Parece ser que hay una inestabilidad cuantificable que aparece cuando se usa cml para dividir

un solo cluster denso en k clusters disjuntos. De hecho, pensamos que hemos capturado el comportamiento de esta inestabilidad en la Conjetura 4.1. Este tipo de inestabilidad no aparece en un método comparable como es el k-medias como se muestra en [Tarpey et al., 1995].

Precisamente este comportamiento inestable inspira una heurística para seleccionar el número óptimo de clusters para los procedimientos de clustering basados en modelos que se sustenta en la estabilidad de k-baricentros de Wasserstein apropiados (ver (14) más arriba). Para comparar usaremos otros criterios de estabilidad bien establecidos. En particular, esto nos lleva a sugerir que desde el punto de vista de la estabilidad es aconsejable no utilizar k-medias. La heurística y los experimentos que apoyan la conjetura anterior, arriba mencionados, se dan en la Sección 4.4 del Capítulo 4.

1

A Kolmogorov approach to approximate validation of models

1.1 Introduction

Often, some feature of a predominant population is clearly different from that of another minority population, simply because of its different eating or cultural habits. In either of these situations, a data sample of that feature taken from the general population will include data that do not come from and do not look like those arising from the predominant one. Consequently, the statistical inference on the main population should be made taking into account the presence of atypical data. As a first ingredient, to address this goal, we resort to a suggestive model introduced in [Huber, 1964], becoming one of the very basis of Robust Statistics, recall the (α-)contamination neighbourhood (CN) of a probability distribution P_0 defined in (3). We must note the use of particular contamination models in different statistical problems, stressing its role on the False-Discovery-Rate (FDR) setting (as considered e.g. in [Genovese and Wasserman, 2004]). We briefly comment on the relation of our approach with that in Section 1.5.

Of course, if an 'outlying label' were available for the data coming from the contaminating distribution, Q, removing the labeled data would produce a legitimate sample from P_0. The relevant fact is that CN's are related to trimmings (4). This relation allows us to work with trimmings, instead of CN's, taking advantage of the underlying meaning of trimming and its mathematical properties. If F and F_0 are distribution functions (d.f.'s in the sequel), we will also use $R_\alpha(F)$ and $\mathcal{V}_\alpha(F_0)$, with the same meanings as before, but defined in terms of d.f.'s.

The natural absence of an outlying label has been traditionally substituted by more or less orthodox trimming criteria, including the oldest consisting in trimming just the extreme values, carrying out the analysis with the remaining data. Recently, mainly in conection with two-sample problems (see e.g. [Álvarez-Esteban et al., 2008, Álvarez-Esteban et al., 2011, Álvarez-Esteban et al., 2012, Álvarez-Esteban et al., 2016]), optimal trimmings have been introduced as the nearest ones to the original model, according to some probability distance or dissimilarity measure. This role will be played here by the Kolmogorov (or L_∞-) distance between d.f.'s on the real line defined in (5). Recall that we will use the notation $\|F - G\|_\infty$, $\|F - G\|$ and $d_K(F, G)$ indistinguishably trough this

chapter.

In this chapter, we develop a robust hypothesis testing procedure based on the previous considerations. Moreover, under the paradigm of a false-model world, we use the elements involved in the procedure to suggest some tools for comparing models or for determining the usefulness of particular models.

The use of CN's, through their connection with trimmings, leads to consider $\mathcal{V}_\alpha(F_0)$ to be the 'reasonable' model. Notice that (see Example 1.1), this approach differs from that based just on d_K-neighbourhoods of F_0, which would have a different meaning (see [Owen, 1995] for this and other classic approaches). As relation (1.5) shows, (4) is also equivalent to $d_K(F_0, R_\alpha(F)) = 0$, giving to the 'trimmed Kolmogorov distance' functional

$$d_K(F_0, R_\alpha(F)) := \min_{\tilde{F} \in R_\alpha(F)} d_K(F_0, \tilde{F}), \tag{1.1}$$

and to the plug-in estimator $d_K(F_0, R_\alpha(F_n))$, a main role into our analysis. (Here F_n is the empirical d.f. based on a sample of n independent random variables with common d.f. F). In particular, we address the possibilities of testing $H_0 : d_K(F_0, R_\alpha(F)) = 0$ vs. $H_1 : d_K(F_0, R_\alpha(F)) > 0$, where 'reasonable' is controlled by the trimming level α. Related null hypotheses have already been considered making use of different probability metrics or different neighbourhoods. In [Álvarez-Esteban et al., 2011, Álvarez-Esteban et al., 2012], the L_2-Wasserstein distance is used in a two-sample version. Previous approaches based on particular trimming procedures were considered in [Munk and Czado, 1998] and [Álvarez-Esteban et al., 2008]. The Kolmogorov-Smirnov test is probably the most widely used goodness of fit test, therefore the d_K-metric provides a privileged setting to develop our approach. Notice that we provide existence and characterization of (a particular) minimizer, and even a result on directional differentiability.

As shown in [Barron, 1989], for any distance d dominating the total variation distance, testing the null hypothesis $P = P_0$ vs. the alternative $d(P, P_0) \geq \rho$ (> 0), makes generally unachievable to get exponential bounds for the involved errors. The test provided in Section 3 has exponentially small error probabilities for testing the null $H_0 : d_K(F_0, R_\alpha(F)) = 0$ (equivalently, $H_0 : F \in \mathcal{V}_\alpha(P_0)$) against the alternative $d_K(F_0, R_\alpha(F)) > \rho$. The test is uniformly consistent (type I and type II error probabilities tend to 0 uniformly) for detecting alternatives $d_K(F_0, R_\alpha(F)) > \eta_n/\sqrt{n}$ with $\eta_n/\sqrt{n} \to 0$ if $\eta_n \to \infty$.

Also, in Section 1.4.1, we provide asymptotic theory for $d_K(F_0, R_\alpha(F_n))$ for inferential purposes. It includes an extension of Theorem 2 in [Raghavachari, 1995] for flexible null hypotheses.

The second main goal in this chapter is to provide tools to compare different models when the null hypothesis is rejected. Under the model falseness paradigm, [Davies, 1995,Davies, 2018] introduce the idea of adequacy region (for a data set) as the set of probabilities in a model whose samples would typically look like the actual data. Also [Rudas et al., 1994] proposes the very natural concept of index of fit, namely, the contamination level necessary to make the random generator of the data a contaminated member of the model. The proposal in [Rudas et al., 1994], as well as its modification in [Liu and Lindsay, 2009], deal with multinomial models. In our setup we consider the trimmed Kolmogorov (tK) index of fit, α^*, defined by

$$\alpha^* = \min\{\alpha : \ d_K(F_0, R_\alpha(F)) = 0\} = \min\{\alpha : F \in \mathcal{V}_\alpha(F_0)\}. \tag{1.2}$$

This is the minimum contamination level α for which F is a contaminated version of F_0. This works in a very general setup, since we impose no constraints on F and F_0. This is in contrast with the methodology involved in the control of FDR, which takes advantage of the dominated contamination model. With the methodology developed here, it is fairly easy to calculate the empirical version of α^* for a particular data set. Using our asymptotic theory for $d_K(F_0, R_\alpha(F_n))$ we propose a consistent estimator for α^* in Section 4. We also provide comparisons with some methodologies developed in the FDR setting (as considered in [Meinshausen and Rice, 2006]) for estimating the proportion of false null hypotheses.

A related approach for comparing the quality of different models to describe the data is based on credibility indices, as introduced in [Lindsay and Liu, 2009]. Given a goodness of fit procedure, the credibility index allows comparison between models based on the minimal sample size n^* for which subsamples of size n^* of the original data (of size n) reject the null hypothesis 50% of times. The idea behind this index is that for large samples, goodness of fit tests will very likely reject the null hypothesis, while often for smaller sub-samples the null would not be rejected. Of course, these credibility indices have to be estimated from the data. The proposal in [Lindsay and Liu, 2009] is to use subsampling to perform this estimation. However, the accuracy of the subsampling approximation is limited to small (as compared to the complete sample) subsample sizes. Here we show how our asymptotic theory for $d_K(F_0, R_\alpha(F_n))$ can provide further information about the credibility indices.

Summarizing, this chapter addresses the analysis and applications of $d_K(F_0, R_\alpha(F))$, the 'trimmed Kolmogorov distance'. Section 1.2 is devoted to collect the mathematical bases and provide a fast algorithm for computation on sample data. The analysis of the proposed testing procedure is carried in Section 1.3. In Section 1.4 we show how to apply this test to credibility analysis and develop some results about the tK-index of fit and the related acceptance regions. The basis for that approach relies on the CLT for the trimmed Kolmogorov distance (see Theorem 1.9). Section 1.5 includes some relations with the FDR setting and comparisons between several estimators of the contamination index α. In Section 1.6 we illustrate the previous techniques to compare descriptive models over simulated and real data examples. In the last section we briefly discuss the results. Finally, the proofs of the main results are given in Section 1.7.

For convenience of the reader and clarity of exposition we provide a guideline of the main results. Proposition 1.1 gives some nice properties of the set of trimmings in the topology induced by the Kolmogorov distance. Proposition 1.2 shows consistency of the trimmed Kolmogorov distance. Lemma 1.3 provides a characterization of the trimmed Kolmogorov distance using quantile functions and is essential in the prove of Theorem 1.4. Theorem 1.4 is one of the main results of the chapter and it shows the existence of a trimming that achieves the trimmed Kolmogorov Distance and hence provides tools for theoretical and numerical computation. Proposition 1.5 shows how to build an Uniformly Exponentially Consistent and flexible test of goodness of fit using the trimmed Kolmogorov distance and Theorem 1.6 is a convenient consequence. Theorem 1.7 and Proposition 1.8 are concerned with the rate of convergence of the trimmed Kolmogorov distance. The main result of the chapter is given in Theorem 1.9 which describes the convergence of $\sqrt{n}\,(d_K(F_0, R_\alpha(F_n)) - d_K(F_0, R_\alpha(F))$ to the supreme of a Gaussian process. Theorem 1.4 plays an important role in the obtention of our CLT result.

1.2 Trimming and Kolmogorov distance

We keep the notation used in the Introduction and notice that the set $R_\alpha(F)$ can be also characterized, as showed in [Álvarez-Esteban et al., 2008] (Proposition 2.2 in [Álvarez-Esteban et al., 2011] gives a more general result), in terms of the set of α-trimmed versions of the uniform law $U(0,1)$. Let $\mathcal{C}_\alpha$ be the set of absolutely continuous functions $h : [0,1] \mapsto [0,1]$, such that $h(0) = 0, h(1) = 1$, with derivative h' verifying $0 \leq h' \leq \frac{1}{1-\alpha}$ a.e.. Then, the composition of the functions h and F: $F_h = h \circ F$ gives the useful parameterization

$$R_\alpha(F) = \{F_h : h \in \mathcal{C}_\alpha\}. \tag{1.3}$$

The set $R_\alpha(F)$ is convex and also well behaved w.r.t. weak convergence of probabilities and widely employed probability metrics (see Section 2 in [Álvarez-Esteban et al., 2011]). We show that $R_\alpha(F)$ keeps several nice properties under d_K.

Proposition 1.1. *For $\alpha \in [0,1)$, if F, G with or without suffixes are d.f.'s:*

(a) $R_\alpha(F)$ is compact w.r.t. d_K.

(b) $d_K(F_0, R_\alpha(F)) = \min_{\tilde{F} \in R_\alpha(F)} \|\tilde{F} - F_0\| = \min_{h \in \mathcal{C}_\alpha} \|h \circ F - F_0\|$.

(c) $|d_K(G_1, R_\alpha(F_1)) - d_K(G_2, R_\alpha(F_2))| \leq d_K(G_1, G_2) + \frac{1}{1-\alpha} d_K(F_1, F_2)$.

(d) If $d_K(F_n, F) \to 0$, then:

(d1) for every $\tilde{F} \in R_\alpha(F)$, there exist $\tilde{F}_n \in R_\alpha(F_n), n \in \mathbb{N}$ such that $d_K(\tilde{F}_n, \tilde{F}) \to 0$.

(d2) if $\tilde{F}_n \in R_\alpha(F_n), n \geq 1$, then there exists some d_K-convergent subsequence $\{\tilde{F}_{n_k}\}$. If $\tilde{F}$ is the limit of such a subsequence, necessarily $\tilde{F} \in R_\alpha(F)$.

(d3) if, additionally, $\{G_n\}$ is any sequence of d.f.'s such that $d_K(G_n, G) \to 0$, then $d_K(G_n, R_\alpha(F_m)) \to d_K(G, R_\alpha(F))$ as $n, m \to \infty$.

Proposition 1.1 guarantees the existence of optimal L_∞-approximations to every distribution function F_0 by α-trimmed versions of F:

$$\text{There exists } \tilde{F} \in R_\alpha(F) \text{ such that } \|F_0 - \tilde{F}\| = d_K(F_0, R_\alpha(F)). \tag{1.4}$$

It also shows, through (4), that for $\alpha \in [0,1)$

$$F \in \mathcal{V}_\alpha(F_0) \text{ if and only if } d_K(F_0, R_\alpha(F)) = 0. \tag{1.5}$$

Moreover, by convexity of $R_\alpha(F)$, the set of optimally trimmed versions of F associated to problem (1.4) is also convex. However, guarantying uniqueness of the minimizer (as it holds w.r.t. L_2- Wasserstein metric by Corollary 2.10 in [Álvarez-Esteban et al., 2011]) is not possible here. An additional consequence of Proposition 1.1 is the continuity of $d_K(F_0, R_\alpha(F))$ in F_0 and F.

By Polya's uniform convergence theorem, if F and G are continuous and $\{F_n\}, \{G_n\}$ are sequences of d.f.'s which, respectively, weakly converge to F, G, then they also converge in the d_K-sense, therefore $d_K(G_n, R_\alpha(F_m)) \to d_K(G, R_\alpha(F))$ holds. Mention apart, by its statistical interest, merits the the following consistency result, which is straightforward from Glivenko-Cantelli theorem and item (d3) above.

Proposition 1.2 (Consistency of trimmed Kolmogorov distance)**.** *Let $\alpha \in [0,1)$ and $\{F_n\}$ be the sequence of empirical d.f.'s based on a sequence $\{X_n\}$ of independent random variables with distribution function F. If $\{G_n\}$ is any sequence of distribution functions d_K-approximating the d.f. G (i.e. $d_K(G_n, G) \to 0$), then:*

$$d_K(G_n, R_\alpha(F_m)) \to d_K(G, R_\alpha(F)), \quad \text{as } n, m \to \infty, \quad \text{with probability one.}$$

While in other contexts the roles played by discarding contamination (by trimming) and the distance under consideration seem to be clear, here the nature of Kolmogorov distance can lead to a distorted picture. To give some light on these roles, we include a very simple example based on uniform laws that allows explicit computations. We also must note that (as commented in [Álvarez-Esteban et al., 2012]) contamination neighbourhoods have been extended in several ways; notably Rieder's neighborhoods of a probability comprise contamination as well as total variation norm neighborhoods.

Example 1.1. Contamination vs d_K-based neighbourhoods. *Let us fix F_0 to be the $U(0,1)$ d.f. and consider the following scenarios for F*

i) F the d.f. of an $U(0, 1+\varepsilon)$ or an $U(-\varepsilon, 1)$ law. Then $d_K(F_0, F) = \frac{\varepsilon}{1+\varepsilon}$ and $d_K(F_0, R_\alpha(F)) = \frac{\varepsilon-\alpha}{1+(\varepsilon-\alpha)}$ if $0 \le \alpha \le \varepsilon$ (and 0 if $\alpha \ge \varepsilon$).

ii) F the d.f. of a $U(0, 1-\varepsilon)$ law. Then $d_K(F_0, F) = \varepsilon$ and $d_K(F_0, R_\alpha(F)) = \varepsilon$ for every $0 \le \alpha < 1$.

In fact, the first situation involves a contamination of exact size ε of F_0, because $F = (1-\varepsilon)F_0 + \varepsilon F'$ where F' is the d.f. of an $U(1, 1+\varepsilon)$ or an $U(-\varepsilon, 0)$ law. In contrast, the second one does not fit in the contamination model at all. The following scenario includes inner contamination at the support of F_0, adding some complexity to the analysis:

iii) $F = (1-\varepsilon)F_0 + \varepsilon F'$, where F' is the d.f. of a $U(a,b)$ law with $0 < a < b < 1$. Then $d_K(F_0, F) = \varepsilon \sup\{a, 1-b\}$, and for $0 \le \alpha \le \varepsilon$: $d_K(F_0, R_\alpha(F)) = (\varepsilon - \alpha)\sup\{a, 1-b\}$, if $0 < a < b \le 1/2$ else $1/2 \le a < b < 1$. If $0 < a \le 1/2 < b < 1$, then for $0 < \alpha < \varepsilon_0 := \varepsilon\frac{|a+b-1|}{b-a}$, we would have $d_K(F_0, R_\alpha(F)) = (\varepsilon - \alpha)\sup\{a, 1-b\}$, while for $\varepsilon_0 \le \alpha \le \varepsilon$, defining $\gamma = |1/2 - \sup\{a, 1-b\}|$, we would have $d_K(F_0, R_\alpha(F)) = [1/2 - \gamma(\varepsilon-\alpha)/(\varepsilon-\varepsilon_0)](\varepsilon-\alpha)$.

The analysis above shows that the effect of optimal trimming according to the d_K-distance strongly depends on several factors. Notably, they include the presence or not of a contaminating part, but also its spread and relative position. □

Throughout this chapter we make frequent use of the quantile function. Given a d.f. F, we write F^{-1} for the associated quantile function. Recall that it is just the left-continuous inverse of the d.f. F, namely, $F^{-1}(t) := \inf\{x \mid t \le F(x)\}$. It allows a useful representation of the corresponding distribution because, if U is a uniformly distributed $U(0,1)$ random variable, $F^{-1}(U)$ has d.f. F. Moreover, if X has a continuous d.f. F, $F_0 \circ F^{-1}$ is easily seen to be the quantile function associated to the r.v. $Y = F_0(X)$. As we show next, under some regularity assumptions $d_K(F_0, R_\alpha(F))$ can be expressed in terms of the function $F_0 \circ F^{-1}$. This fact allows the practical computation of $d_K(F_0, R_\alpha(F_n))$ when F_n is an empirical d.f. based on a data sample $x_1, \ldots, x_n$, and even that of $d_K(F_0, R_\alpha(F))$ for theoretical distributions (see Example 1.2). Recall that then $F_n(x) := \frac{1}{n}\sum_{i=1}^n I_{(-\infty,x]}(x_i)$.

Lemma 1.3. *If F, F_0 are continuous d.f.'s and F is additionally strictly increasing then*

$$d_K(F_0, R_\alpha(F)) = \min_{h\in\mathcal{C}_\alpha} \|h - F_0 \circ F^{-1}\| \quad \textit{and} \quad d_K(F_0, R_\alpha(F_n)) = \min_{h\in\mathcal{C}_\alpha} \|h - F_0 \circ F_n^{-1}\|.$$

We include below a fundamental tool for our goals. It gives an explicit characterization of a solution of the corresponding optimization problem.

Theorem 1.4. *Assume $\Gamma : [0,1] \to [0,1]$ is a continuous nondecreasing function. Define $G(t) = \Gamma(t) - \frac{t}{1-\alpha}$, $U(t) = \sup_{t\le s\le 1} G(s)$, $L(t) = \inf_{0\le s\le t} G(s)$ and*

$$\tilde{h}_\alpha(t) = \max\left(\min\left(\tfrac{U(t)+L(t)}{2}, 0\right), \tfrac{-\alpha}{1-\alpha}\right).$$

Then $h_\alpha := \tilde{h}_\alpha + \frac{\cdot}{1-\alpha}$ is an element of $\mathcal{C}_\alpha$, and

$$\min_{h\in\mathcal{C}_\alpha} \|h - \Gamma\| = \|h_\alpha - \Gamma\| = \|\tilde{h}_\alpha - G\|.$$

Note that the assumption on Γ is always verified when $\Gamma = F_0 \circ F^{-1}$, and that taking right and left limits at 0 and 1, respectively, we can assume that $F_0 \circ F^{-1}$ is a nondecreasing (and left continuous) function from $[0,1]$ to $[0,1]$.

A key aspect in Theorem 1.4 is that, although not necessarily unique, h_α is an optimal trimming function in the sense described above. However, from the point of view of asymptotic theory, Theorem 1.4 is the key to our Theorem 1.9 in Section 1.4. Moreover, from a practical point of view, it yields a simple algorithm for the computation of $d_K(F_0, R_\alpha(F_n))$, as follows.

Assume $X_1, \ldots, X_n$ are i.i.d. observations from the continuous and strictly increasing d.f. F and assume that F_0 is continuous. From Lemma 1.3 and Theorem 1.4 we know that $d_K(F_0, R_\alpha(F_n)) = \|\tilde{h}_{\alpha,n} - G_n\|$, where $G_n(t) = H_n^{-1}(t) - \frac{t}{1-\alpha}$, H_n^{-1} is the empirical quantile function of the transformed data, $Y_i = F_0(X_i)$, $U_n(t) = \sup_{t\le s\le 1} G_n(s)$, $L_n(t) = \min_{0\le s\le t} G_n(s)$ and

$$\tilde{h}_{\alpha,n}(t) = \max\left(\min\left(\frac{U_n(t) + L_n(t)}{2}, 0\right), \frac{-\alpha}{1-\alpha}\right).$$

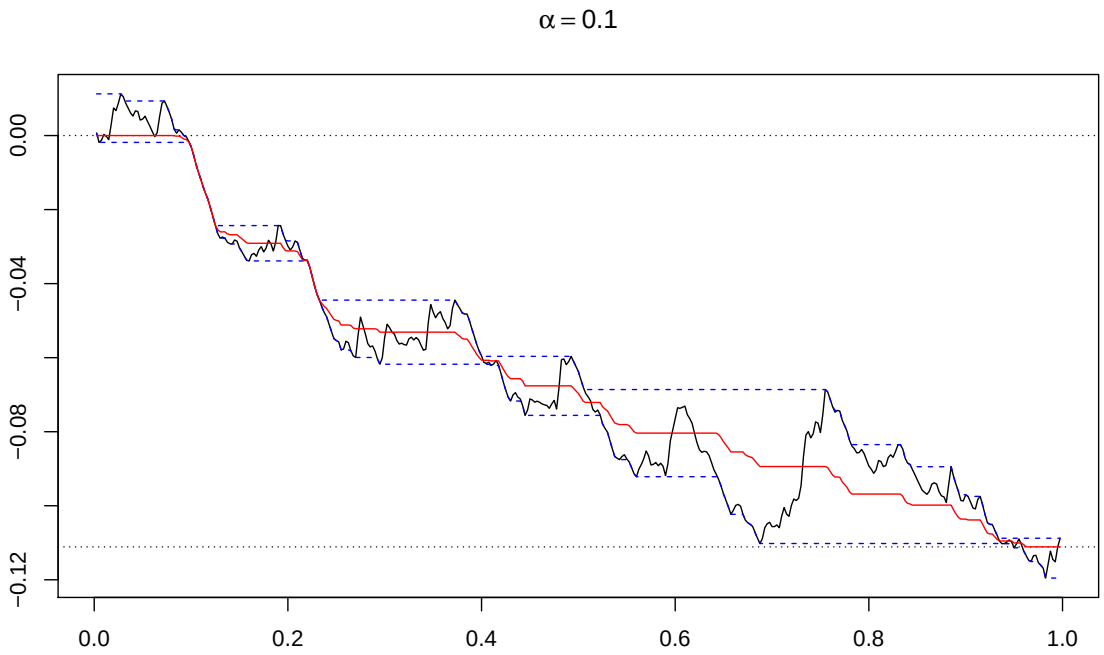
To gain some intuition we have plotted an example of $\tilde{h}_{\alpha,n}$ in Figure 1.1. Denote by $Y_{(1)} \le \cdots \le Y_{(n)}$ the ordered (transformed) sample. Note that $G_n(t) = Y_{(i)} - \frac{t}{1-\alpha}$ if $t \in (\frac{i-1}{n}, \frac{i}{n}]$, while $\tilde{h}_{\alpha,n}$ is a non-increasing function and this implies that

$$\|\tilde{h}_{\alpha,n} - G_n\| = \max_{1\le i\le n}\left(\max(G_n(\tfrac{i-1}{n}+) - \tilde{h}_{\alpha,n}(\tfrac{i-1}{n}), \tilde{h}_{\alpha,n}(\tfrac{i}{n}) - G_n(\tfrac{i}{n}))\right),$$

with $G_n(\frac{i-1}{n}+) = Y_{(i)} - \frac{i-1}{n(1-\alpha)}$, $G_n(\frac{i}{n}) = Y_{(i)} - \frac{i}{n(1-\alpha)}$. In Figure 1.1 we see that the maximum L_∞ distance between $\tilde{h}_{\alpha,n}$ and G_n is achieved around 0.7 and around 0.76. For the computation of $\tilde{h}_{\alpha,n}(\frac{i}{n})$ we note that $U_n(\frac{i}{n}) = \max_{i\le j\le n-1} G_n(\frac{j}{n}+)$ and $L_n(\frac{i}{n}) = \min_{1\le j\le i} G_n(\frac{j}{n})$ for $i = 1, \ldots, n-1$. Summarizing, we see that $d_K(F_0, R_\alpha(F_n))$ can be computed through the following algorithm.

Beyond this algorithm for the empirical case, Theorem 1.4 provides a simple way for the computation of theoretical trimmed Kolmorogov distances. We analyze the problem for the Gaussian model.

Figure 1.1: Representation of $\tilde{h}_{\alpha,n}$ in red for $\alpha = 0.1$ and $n = 400$ when we take $Y = F_0(X) \sim U(0,1)$. In black we have G_n, in dashed blue we have U_n and L_n. Black dashed lines represent the box constraints 0 and $-\alpha/(1-\alpha)$.



Algorithm 1 Computation of $d_K(F_0, R_\alpha(F_n))$

Input: $X_1, \dots, X_n, \alpha$

for $1 \le i \le n$ **do**
  $Y_i \leftarrow F_0(X_i)$
end for
$Y \leftarrow \{Y_1, \dots, Y_n\}$; $Y \leftarrow$ sort Y
for $0 \le i \le n-1$ **do**
  $g_{i+} \leftarrow Y_{(i+1)} - \frac{i}{n(1-\alpha)}$
end for
for $1 \le i \le n$ **do**
  $g_{i-} \leftarrow Y_{(i)} - \frac{i}{n(1-\alpha)}$
end for
for $1 \le i \le n-1$ **do**
  $u_i \leftarrow \max_{i \le j \le n-1} g_{j+}$
  $l_i \leftarrow \min_{1 \le j \le i} g_{j-}$
end for
$h_0 \leftarrow 0$; $h_n \leftarrow -\frac{\alpha}{1-\alpha}$
for $1 \le i \le n-1$ **do**
  $h_i \leftarrow \max(\min(0, \frac{u_i + l_i}{2}), -\frac{\alpha}{1-\alpha})$
end for
$d_K(F_0, R_\alpha(F_n)) \leftarrow \max_{1 \le i \le n} \big(\max(g_{(i-1)+} - h_{i-1}, h_i - g_{i-}) \big)$

Output: $d_K(F_0, R_\alpha(F_n))$

Example 1.2 (Trimmed Kolmogorov distances in the Gaussian model.)**.** *Consider the case $F_0 = \Phi$, $F = \Phi((\cdot - \mu)/\sigma)$, where Φ denotes the standard normal d.f., $\mu \in \mathbb{R}$ and $\sigma > 0$. Here we have $H^{-1}(t) := F_0 \circ F^{-1}(t) = \Phi(\mu + \sigma\Phi^{-1}(t))$. We note that $w(t) := (H^{-1})'(t) \le 1/(1-\alpha)$ if and only if $p(\Phi^{-1}(t)) \ge 0$, where*

$$p(x) = (\sigma^2 - 1)x^2 + 2\mu\sigma x + \mu^2 - 2\log((1-\alpha)\sigma).$$

To avoid cumbersome computations we focus on the cases $\sigma = 1$, $\mu \neq 0$ and $\mu = 0$, $\sigma \neq 1$.

If $\sigma = 1$ and $\mu > 0$ then p is linear with positive slope and we see that $w(t) \le 1/(1-\alpha)$ if and only if $t \ge t_0 = \Phi\big(-\frac{\mu}{2} + \frac{1}{\mu}\log(1-\alpha)\big)$. This means that $G(s) = H^{-1}(s) - s/(1-\alpha)$ is increasing in $[0, t_0]$ and decreasing in $[t_0, 1]$. Since, $H^{-1}(0) = G(0) = 0$, we have that, $\tilde{h}_\alpha(t) = 0$ for $t \in [0, t_1]$, where $t_1 \in (t_0, 1)$ is (the unique) solution to $G(t_1) = 0$, and $\tilde{h}_\alpha(t) = G(t)$ for $t \in [t_1, 1]$. We conclude that $d_K(R_\alpha(N(\mu,1)), N(0,1)) = G(t_0)$. The case $\mu < 0$ can be handled similarly to obtain

$$d_K(R_\alpha(N(\mu,1)), N(0,1)) = \Phi\big(\tfrac{|\mu|}{2} + \tfrac{1}{|\mu|}\log(1-\alpha)\big) - \tfrac{1}{1-\alpha}\Phi\big(-\tfrac{|\mu|}{2} + \tfrac{1}{|\mu|}\log(1-\alpha)\big), \quad \mu \neq 0.$$

We focus now on the case $\mu = 0$. If $\sigma^2 < 1$, p is a parabola with negative leading coefficient and discriminant $\Delta^2 = 8(\sigma^2 - 1)\log(\sigma(1-\alpha)) > 0$. Hence, $p(x)$ is positive for $x \in (x_a, x_b)$ with $x_a = -\frac{\Delta}{2(1-\sigma^2)}$, $x_b = \frac{\Delta}{2(1-\sigma^2)}$. Equivalently, $w(t) \le 1/(1-\alpha)$ if and only if $t_a := \Phi(x_a) \le t \le t_b := \Phi(x_b)$. This means that G is increasing in $[0, t_a)$, decreasing in $[t_a, t_b]$, increasing in $(t_b, 1]$, $G(0) = 0$ and $G(1) = -\alpha/(1-\alpha)$. Arguing

as above, we have $\tilde{h}_\alpha(t) = \min(G(t), 0)$ *for* $0 \leq t \leq \frac{1}{2}$, $\tilde{h}_\alpha(t) = \max(G(t), -\frac{\alpha}{1-\alpha})$ *for* $\frac{1}{2} \leq t \leq 1$, $\tilde{h}_\alpha(t_a) = 0$ *and* $\tilde{h}_\alpha(t_b) = \frac{-\alpha}{1-\alpha}$. *We conclude that* $d_K(R_\alpha(N(\mu, \sigma^2)), N(0,1)) = G(t_a) - \tilde{h}_\alpha(t_a) = \tilde{h}_\alpha(t_b) - G(t_b)$. *Hence,*

$$d_K(R_\alpha(N(0,\sigma^2)), N(0,1)) = \Phi\Big(\frac{-\sigma\frac{\Delta}{2}}{1-\sigma^2}\Big) - \frac{1}{1-\alpha}\Phi\Big(\frac{-\frac{\Delta}{2}}{1-\sigma^2}\Big), \quad \text{if } \sigma < 1.$$

If $1 \leq \sigma \leq 1/(1-\alpha)$ *then we have that* $w(t) \leq 1/(1-\alpha)$ *for all* t *and* $h_0 = H^{-1} \in \mathcal{C}_\alpha$. *In particular,* $d_K(R_\alpha(N(0,\sigma^2)), N(0,1)) = 0$.

Finally, we consider the case $\sigma > 1/(1-\alpha)$. *In this case* p *is positive for* $x \notin [x_a, x_b]$ *with* $x_a = -\frac{\Delta}{2(\sigma^2-1)}$, $x_b = \frac{\Delta}{2(\sigma^2-1)}$. *This means that* $(H^{-1})'(t) > \frac{1}{1-\alpha}$ *for* $t \in (t_a, t_b)$ *with* $t_a = \Phi(x_a), t_b = \Phi(x_b)$. *Therefore,* G *is decreasing in* $[0, t_a)$, *increasing in* $[t_a, t_b]$, *decreasing in* $(t_b, 1]$, $G(0) = 0$ *and* $G(1) = -\alpha/(1-\alpha)$. *Hence,* $\tilde{h}_\alpha(t) = \max(G(t), \frac{G(t)+G(t_b)}{2})$, $0 \leq t \leq t_a$, $\tilde{h}_\alpha(t) = \frac{G(t_a)+G(t_b)}{2}$, $t_a \leq t \leq t_b$, $\tilde{h}_\alpha(t) = \min(G(t), \frac{G(t_a)+G(t)}{2})$, $t_b \leq t \leq 1$. *In particular,* $d_K(R_\alpha(N(0,\sigma^2)), N(0,1)) = \tilde{h}_\alpha(t_a) - G(t_a) = G(t_b) - \tilde{h}_\alpha(t_b) = \frac{1}{2}(G(t_b) - G(t_a))$, *that is,*

$$d_K(R_\alpha(N(0,\sigma^2)), N(0,1)) = \Phi\Big(\frac{\sigma\frac{\Delta}{2}}{\sigma^2-1}\Big) - \frac{\Phi\Big(\frac{\frac{\Delta}{2}}{\sigma^2-1}\Big) - \frac{\alpha}{2}}{1-\alpha}, \quad \text{if } \sigma > \frac{1}{1-\alpha}.$$

To summarize, we have:

If $\sigma = 1$ *and* $\mu \neq 0$ *then*

$$d_K(R_\alpha(N(\mu,1)), N(0,1)) = \Phi\big(\tfrac{|\mu|}{2} + \tfrac{1}{|\mu|}\log(1-\alpha)\big) - \tfrac{1}{1-\alpha}\Phi\big(-\tfrac{|\mu|}{2} + \tfrac{1}{|\mu|}\log(1-\alpha)\big). \tag{1.6}$$

In the case $\mu = 0$*:*

$$d_K(R_\alpha(N(0,\sigma^2)), N(0,1)) = \begin{cases} \Phi\Big(\frac{-\sigma\frac{\Delta}{2}}{1-\sigma^2}\Big) - \frac{1}{1-\alpha}\Phi\Big(\frac{-\frac{\Delta}{2}}{1-\sigma^2}\Big), & \text{if } \sigma < 1 \\ 0, & \text{if } 1 \leq \sigma \leq 1/(1-\alpha) \\ \Phi\Big(\frac{\sigma\frac{\Delta}{2}}{\sigma^2-1}\Big) - \frac{\Phi\Big(\frac{\frac{\Delta}{2}}{\sigma^2-1}\Big) - \frac{\alpha}{2}}{1-\alpha}, & \text{if } \sigma > 1/(1-\alpha) \end{cases}$$

□

Relations (1.4) and (1.5) state the link between CN's and trimming, opening ways to approximately validating a model making use of trimming through the Kolmogorov distance. We end this section showing how CN's and approximate validation in a parametric model setting can be related. For that task we focus on what are the parameters in the model leading to distributions in $\mathcal{V}_\alpha(F_0)$. As pointed out in [Davies, 1995], we should just consider models able to generate data similar to our sample. Moreover, distributions in a CN have an intuitive appeal and, if α is small, we can expect to be handling reasonable models. For instance, if $F_0 \sim N(0,1)$ then we can calculate the tolerance region given by the subset of normal distributions belonging to $\mathcal{V}_\alpha(F_0)$ in an elementary fashion. This provides an approximate picture of the kind of distributions present in the CN of F_0. These tolerance regions for $\alpha = 0.05$ and $\alpha = 0.1$ are shown in Figure 1.2. Every combination of $(\tilde{\mu}, \tilde{\sigma})$ inside the green border is a normal distribution that belongs to $\mathcal{V}_{0.1}(N(0,1))$. The same is true for the red border and $\mathcal{V}_{0.05}(N(0,1))$.

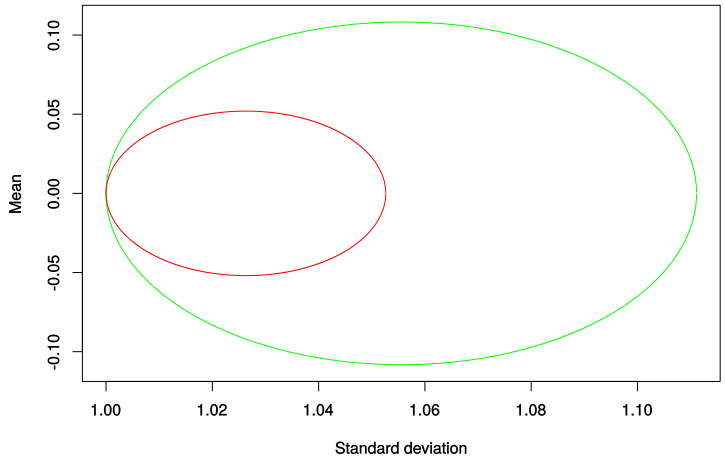


Figure 1.2: Plot of regions containing the parameters compatible with α-contamination neighbourhoods of $F_0 \sim N(0,1)$ for $\alpha = 0.05$ (red) and $\alpha = 0.1$ (green).

1.3 Hypothesis testing

To develop our approach for a testing procedure, throughout, $X_1, \ldots, X_n$ will be independent random variables with common d.f. F, and F_n will be the corresponding empirical d.f. The main result, following the principles in [Barron, 1989], concerns control of error probabilities: a test is uniformly consistent (UC) if both type I and type II error probabilities (EI and EII in the sequel) converge uniformly to 0 as the sample size, $n \to \infty$, and it is uniformly exponentially consistent (UEC) if the error probabilities are uniformly bounded by e^{-rn} for large n and some $r > 0$. To stress on the necessity of considering some separating zone between the null and the alternative, we include this previous slightly more general result.

Proposition 1.5. *Given $0 \leq \rho_1 < \rho_2$, for testing $H_0 : d_K(F_0, R_\alpha(F)) \leq \rho_1$ vs. $H_1 : d_K(F_0, R_\alpha(F)) > \rho_2$, for every $0 < \lambda < 1$ rejecting the null hypothesis when $d_K(F_0, R_\alpha(F_n)) > (1-\lambda)\rho_1 + \lambda\rho_2$ is an uniformly exponentially consistent (UEC) test.*

Proof. From Proposition 1.1 (c), we have the inequality $|d_K(F_0, R_\alpha(F_1)) - d_K(F_0, R_\alpha(F_2))| \leq \frac{1}{1-\alpha}\|F_1 - F_2\|$, thus for EI:

$$\begin{aligned}
&P_F\Big(d_K(F_0, R_\alpha(F_n)) > (1-\lambda)\rho_1 + \lambda\rho_2\Big) \\
&\leq P\Big(d_K(F_0, R_\alpha(F_n)) - d_K(F_0, R_\alpha(F)) > \lambda\rho_2 - \lambda\rho_1\Big) \\
&\leq P\Big(\frac{1}{1-\alpha}\sup_x |F_n(x) - F(x)| > \lambda(\rho_2 - \rho_1)\Big) \\
&= P\Big(\sup_x \sqrt{n}|F_n(x) - F(x)| > \sqrt{n}(1-\alpha)\lambda(\rho_2 - \rho_1)\Big) \\
&\leq 2e^{-2\lambda^2 n(1-\alpha)^2(\rho_2-\rho_1)^2}. \qquad (1.7)
\end{aligned}$$

Note that the last bound follows from the [Massart, 1990] version of the Dvoretsky-Kiefer-Wolfowitz inequality.

To handle EII (thus if $d_K(F_0, R_\alpha(F)) > \rho_2$), we have

$$\begin{aligned} P_F\Big(d_K(F_0, R_\alpha(F_n)) \le (1-\lambda)\rho_1 + \lambda\rho_2\Big) \\ &= P_F\Big(\rho_2 - d_K(F_0, R_\alpha(F_n)) \ge \rho_2 - ((1-\lambda)\rho_1 + \lambda\rho_2)\Big) \\ &\le P\Big(d_K(F_0, R_\alpha(F)) - d_K(F_0, R_\alpha(F_n)) > (1-\lambda)(\rho_2 - \rho_1)\Big) \\ &\le P\Big(\sup_x \sqrt{n}|F_n(x) - F(x)| > \sqrt{n}(1-\alpha)(1-\lambda)(\rho_2 - \rho_1)\Big) \\ &\le 2e^{-2(1-\lambda)^2 n(1-\alpha)^2(\rho_2-\rho_1)^2}. \end{aligned} \tag{1.8}$$

□

As an easy consequence, taking $\rho_1 = 0$ and $\rho = \rho_2$, we get:

Theorem 1.6. *Given $\rho > 0$, for testing*

$$H_0 : d_K(F_0, R_\alpha(F)) = 0 \quad \textit{vs.} \quad H_1 : d_K(F_0, R_\alpha(F)) > \rho, \tag{1.9}$$

for every $0 < \lambda < 1$ the critical region $d_K(F_0, R_\alpha(F_n)) > \lambda\rho$ defines an uniformly exponentially consistent (UEC) test.

Since the null hypothesis includes all the contamination versions (of α-level) of F_0, rejection means that the generator of the sample is far enough of any such a contaminated version. Theorem 1.6 guarantees that alternatives will be quickly detected when farness is measured through the d_K-distance.

In statistical practice, it could be wiser to change the alternative hypothesis and make it sample size dependent. That leads to consider tests of the form

$$H_{0,n} : d_K(F_0, R_\alpha(F)) = 0 \quad \text{vs.} \quad H_{1,n} : d_K(F_0, R_\alpha(F)) > \rho_n, \tag{1.10}$$

for $\rho_n = \rho(n) > 0$, and rejection when $d_K(F_0, R_\alpha(F_n)) > \lambda\rho_n$. For instance, taking $\rho_n = \eta_n/\sqrt{n} \to 0$, and $\eta_n \to \infty$ results in an uniformly consistent test. Uniform consistency is weaker than uniform exponential consistency, but it allows to detect, for example, alternatives at a distance $\log(n)/\sqrt{n}$. Also, we can consider λ as a tuning parameter which can help if we have some additional information or if we want more or less conservative tests with respect to EI and EII probabilities (of course, when $\rho = 0$, $\rho_1 = \rho_2$ or $\lambda = 0$ or $\lambda = 1$, some bounds are meaningless and we can not assure uniform consistency with the previous procedure). Alternatively, we may look for the smallest possible values for ρ_n, while still controlling EI and EII. From (1.7) and (1.8) note that if ρ_n is $o(n^{-1/2})$ we would lose the control of the errors, since $n\rho_n^2 \to 0$ as $n \to \infty$. This leads us to choose ρ_n as $O(n^{-1/2})$, or, fixing some $\rho > 0$:

$$\rho_n = \frac{\rho}{\sqrt{n}} \tag{1.11}$$

Now, if we fix $0 < \epsilon_1, \epsilon_2 < 1$, looking for a rejection threshold, $\lambda\rho_n$, for which

$$EI \le \epsilon_1 \quad \text{and} \quad EII \le \epsilon_2,$$

we get $2e^{-2\lambda^2(1-\alpha)^2\rho^2} = \epsilon_1$ and $\epsilon_1 e^{4\lambda-2} = \epsilon_2$. With a bit of algebra we get

$$\rho = \frac{1}{(1-\alpha)\lambda}\sqrt{\frac{1}{2}\log\frac{2}{\epsilon_1}}, \quad \lambda = \frac{1}{2} + \frac{1}{4}\log\frac{\epsilon_2}{\epsilon_1}, \tag{1.12}$$

imposing $\epsilon_1 e^{-2} < \epsilon_2 < \epsilon_1 e^2$, which gives the optimal boundary level

$$\rho_n = \frac{\rho}{\sqrt{n}} = \frac{1}{(1-\alpha)\lambda}\sqrt{\frac{1}{2n}\log\frac{2}{\epsilon_1}}. \tag{1.13}$$

Relations (1.12) and (1.13) summarize the balance among the different elements. Ideally, we look for small ρ_n, ϵ_1 and ϵ_2 but, paying the price for our demands, ρ_n grows as ϵ_1 gets smaller and as ϵ_2 gets more similar to ϵ_1. Therefore, we need to make sensible choices for ϵ_1 and ϵ_2. In Table 1.1 we show some examples of the mentioned behaviour. For instance, fixing $\epsilon_1 = 0.01$ and $\epsilon_2 = 0.05$ seems a sensible choice, giving a fairly low ρ_{1000} while keeping low error probabilities.

Table 1.1: Values associated to error bounds for $\alpha = 0.1$ and $N = 1000$.

EI	EII	λ	ρ_{1000}	EI	EII	λ	ρ_{1000}	EI	EII	λ	ρ_{1000}
0.1	0.5	0.90	0.048	0.05	0.25	0.90	0.053	0.01	0.05	0.90	0.063
0.1	0.1	0.50	0.086	0.05	0.05	0.50	0.095	0.01	0.01	0.50	0.114
0.1	0.02	0.10	0.440	0.05	0.01	0.10	0.489	0.01	0.002	0.10	0.586

An appealing goal would be to detect the 'true' contamination level, that is, the minimal level of trimming for which the postulated model would not be rejected. In this way we could, also, detect possible contaminations in the generating mechanism. To address this objective, we resort to the following result obtained in greater generality in [del Barrio and Matrán, 2013].

Theorem 1.7. *If $\alpha \in (0,1)$ and $\nu > 1$, then*

$$d_K(F, R_\alpha(F_n)) = o_P\Big(\frac{(\log n)^\nu}{n}\Big). \tag{1.14}$$

Therefore, if $F = (1-\alpha_0)F_0 + \alpha_0 G_0$ and we test for $\alpha > \alpha_0$, as $n \to \infty$, trimming α from F_n will eliminate the part of the sample coming from G_0, but also will affect the part of the sample coming from F_0. This fact and Proposition 1.2 lead to the following statement.

Proposition 1.8. *Let $\rho_n = O(n^{-1/2})$ and $\rho_n^{-1} = O(n^{1/2})$, and $\alpha > \alpha_0$. Then:*

$$\frac{d_K(F_0, R_\alpha(F_n))}{\rho_n} \to \begin{cases} \infty \text{ almost surely,} & \text{if } d_K(F_0, R_\alpha(F)) > 0 \\ 0 \text{ in probability,} & \text{if } F_0 \in R_{\alpha_0}(F). \end{cases} \tag{1.15}$$

This means that, for big enough samples, our testing procedure will be able to detect the *overtrimming* boundary, that is, the trimming level beyond which the trimmed sample is closer to the model than true random samples from that model. In Figure 1.3 we are able to appreciate this behaviour (see the caption for details). The frequency of rejecting

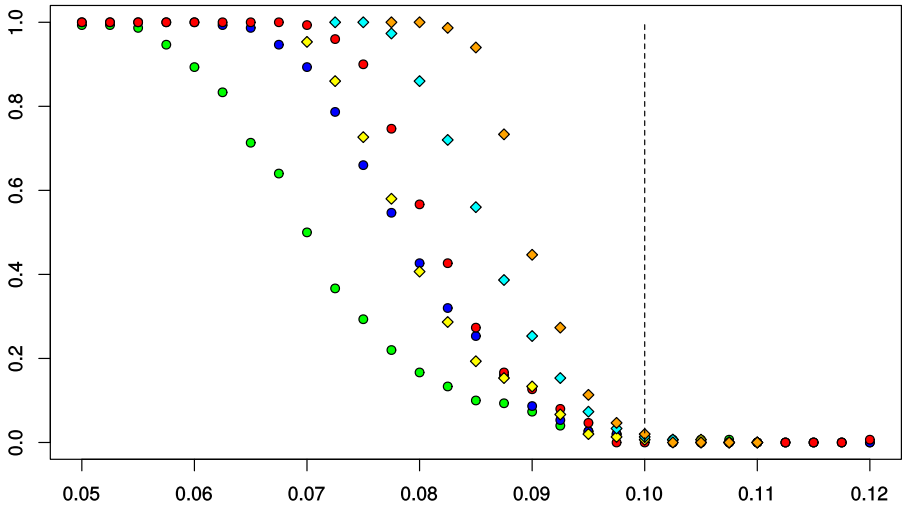


Figure 1.3: Round green (blue, red) dots represent the frequency of rejection (y label) for 150 independent samples of a generating mechanism $F_1 \sim 0.9N(0,1) + 0.1N(3,1)$ for sample sizes 2000 (4000, 6000) and a model $F_0 \sim N(0,1)$, as we vary the trimming level α (x label). Diamond yellow (cyan, orange) dots represent the rejection frequency for a generator $F_2 \sim 0.9N(0,1) + 0.1N(0, 0.1^2)$ for sample sizes 12500 (25000, 50000). The black dashed line represents the true contamination level which is 0.1, since $F_0 \in R_{0.1}(F_1)$ and $F_0 \in R_{0.1}(F_2)$. The error probabilities are fixed to $\epsilon_1 = \epsilon_2 = 0.05$.

the null, for both models, after trimming 0.11 or more is almost zero, the theoretical contamination being 0.1. We see that around 0.1 the models start dropping abruptly the rejection level, but that for the model contaminated with a $N(3,1)$ we need much less points to attain the expected behaviour than we need for the model contaminated with a $N(0, 0.1^2)$. In other words, the presence of a meaningful outlier contamination, even when trimming is allowed, disturbs more heavily the Kolmogorov distance than the presence of equally meaningful inlier contamination. In any case, these results suggest that it may be possible to find an estimator for the 'true' contamination level. We elaborate a little bit more about this in the next section.

1.4 A central limit theorem with applications

We divide this section in two subsections, respectively devoted to the presentation of results and to some of their applications. In particular, we stress on the extension of some of the applications that [Lindsay and Liu, 2009] and [Liu and Lindsay, 2009] explored just on multinomial models.

1.4.1 A central limit result

What follows is our main theoretical result which describes the asymptotic behaviour of the normalized difference between the empirical estimator and the theoretical trimmed Kolmogorov distance under some regularity assumptions. We recall from Section 2 that $d_K(F_0, R_\alpha(F))$ can be expressed in terms of $H^{-1} := F_0 \circ F^{-1}$. We need to introduce the following sets, with G, U, L and $\tilde{h}_\alpha$ standing for the same objects as in Theorem 1.4 in Section 2,

$$T_1 = \Big\{t \in [0,1] : G(t) = \|\tilde{h}_\alpha - G\|, \tfrac{1}{2}(U(t) + L(t)) \geq 0\Big\}, \tag{1.16}$$

$$T_2 = \Big\{t \in [0,1] : -\tfrac{\alpha}{1-\alpha} - G(t) = \|\tilde{h}_\alpha - G\|, \tfrac{1}{2}(U(t) + L(t)) \leq \tfrac{-\alpha}{1-\alpha}\Big\}, \tag{1.17}$$

$$T_3 = \Big\{(s,t) : 0 \leq s \leq t \leq 1, \tfrac{1}{2}(G(t) - G(s)) = \|\tilde{h}_\alpha - G\|, \tfrac{1}{2}(G(t) + G(s)) \in [-\tfrac{\alpha}{1-\alpha}, 0]\Big\}. \tag{1.18}$$

A look at Section 1.7.3 shows that $T_1 \cup T_2 \cup T_3 \neq \emptyset$ provided H^{-1} is continuous. We further denote $T_1^* = \{t \in T_1 : \frac{1}{2}(U(t)+L(t)) = 0\}$, $T_2^* = \{t \in T_2 : \frac{1}{2}(U(t)+L(t)) = -\frac{\alpha}{1-\alpha}\}$ and $T_3^* = \{(s,t) \in T_3 : \frac{1}{2}(G(t) + G(s)) \in \{-\frac{\alpha}{1-\alpha}, 0\}\}$. To avoid pathological examples we will assume that

$$T_1^* = \emptyset, \quad T_2^* = \emptyset, \quad T_3^* = \emptyset. \tag{1.19}$$

Our last regularity assumptions concern H, the d.f. of the random variable $F_0(X)$, where $X \sim F$. They allow the use of the strong approximation of the quantile process in the proof of the theorem (developed in the Appendix). We assume that H has a density, h supported in $[a, b]$ (note that, necessarily, $[a, b] \subset [0, 1]$) and either one of

$$h \text{ is positive and continuous on } [a, b], \tag{1.20}$$

$$h \text{ is positive and continuous on } (a, b); \text{ for some } \varepsilon > 0,\ T_1, T_2 \subset [\varepsilon, 1-\varepsilon],\ T_3 \subset [\varepsilon, 1-\varepsilon]^2. \tag{1.21}$$

Theorem 1.9. *Assume that F_0 and F are continuous d.f.'s, that F is strictly increasing and that the d.f. H associated to $H^{-1} = F_0 \circ F^{-1}$ satisfies (1.19) and either (1.20) or (1.21). Then,*

$$\sqrt{n}\,(d_K(F_0, R_\alpha(F_n)) - d_K(F_0, R_\alpha(F))) \\ \underset{w}{\to} \tfrac{1}{1-\alpha} \max\Big(\max_{t\in T_1} B(t), \max_{t\in T_2}(-B(t)), \max_{(s,t)\in T_3} \tfrac{1}{2}(B(t) - B(s))\Big),$$

where B is a Brownian bridge on $[0, 1]$.

The limit distribution in this result corresponds to the supremum of a Gaussian process. In fact, the index set for this process is often rather simple, consisting of only one or two points as we show in our next example.

Example 1.10. Trimmed Kolmogorov distances in the Gaussian model (cont.) *We revisit the cases studied in Example 1.2. Recall that $F_0 = \Phi$, $F = \Phi((\cdot - \mu)/\sigma)$ and $H^{-1}(t) = \Phi(\mu + \sigma\Phi^{-1}(t))$. Hence $H(x) = \Phi\Big(\frac{\Phi^{-1}(x)-\mu}{\sigma}\Big)$, $0 \leq x \leq 1$, which is supported in $[0, 1]$ and has a density which is positive and continuous on $(0, 1)$.*

In the case $\sigma = 1$ *and* $\mu > 0$ *the computations in Example 1.2 yield that* $T_1 = \{t_0\}$, $T_2 = \emptyset$, $T_3 = \emptyset$, *with* $t_0 = \Phi\big(-\frac{\mu}{2} + \frac{1}{\mu}\log(1-\alpha)\big)$. *Applying Theorem 1.9 we obtain that*

$$\sqrt{n}\big(d_K(R_\alpha(F_n), N(0,1))) - d_K(R_\alpha(N(\mu,1)), N(0,1))\big) \underset{w}{\to} N\big(0, \tfrac{t_0(1-t_0)}{(1-\alpha)^2}\big).$$

When $\mu = 0$ *and* $\sigma^2 < 1$, *writing* $x_a = -\frac{\Delta}{2(1-\sigma^2)}$, $x_b = \frac{\Delta}{2(1-\sigma^2)}$ *(with* $\Delta = (8(\sigma^2 - 1)\log(\sigma(1-\alpha)))^{1/2}$*),* $t_a = \Phi(x_a)$ *and* $t_b = \Phi(x_b)$ *we get* $T_1 = \{t_a\} = \{1 - t_b\}$, $T_2 = \{t_b\}$, $T_3 = \emptyset$ *and Theorem 1.9 yields*

$$\sqrt{n}\big(d_K(R_\alpha(F_n), N(0,1))) - d_K(R_\alpha(N(0,\sigma^2)), N(0,1))\big) \underset{w}{\to} \tfrac{1}{1-\alpha}\max\big(B(1-t_b), -B(t_b)\big),$$

with B *a Brownian bridge.*

Finally, if $\mu = 0$ *and* $\sigma > 1/(1-\alpha)$ *then* $T_1 = T_2 = \emptyset$, *while* $T_3 = \{(t_a, t_b)\}$, *with* $t_a = \Phi(x_a)$, $t_b = \Phi(x_b)$, $x_a = -\frac{\Delta}{2(\sigma^2-1)}$ *and* $x_b = \frac{\Delta}{2(\sigma^2-1)}$ *and we obtain*

$$\sqrt{n}\big(d_K(R_\alpha(F_n), N(0,1))) - d_K(R_\alpha(N(0,\sigma^2)), N(0,1))\big) \underset{w}{\to} N\big(0, \tfrac{(1-t_b)(t_b-\frac{1}{2})}{(1-\alpha)^2}\big).$$

□

The asymptotics showed in the previous example would allow to build asymptotic upper and lower confidence bounds for the Kolmogorov distance between the random generator of the data and the set of α-trimmings of the postulated normal model. In general, we would not be able to describe the sets T_i involved in the limit law, but Theorem 1.9 can be used to obtain conservative confidence bounds. Let $\beta \in (0, \frac{1}{2})$ be given and write $Z_\alpha(F, F_0)$ for the limiting random variable in Theorem 1.9. Recall that for a Brownian bridge and $0 \le s \le t \le 1$ we have $\mathrm{Var}(B(t)) = t(1-t)$ and $\mathrm{Var}(\frac{1}{2}(B(t) - B(s))) = \frac{1}{4}(t-s)(1-(t-s))$. The β-quantile of $Z_\alpha(F, F_0)$ must be lower bounded by the β-quantile of the centered Gaussian r.v.'s $\frac{1}{1-\alpha}B(t)$, $t \in T_1$, $\frac{1}{1-\alpha}(-B(t))$, $t \in T_2$ and $\frac{1}{2(1-\alpha)}(B(t) - B(s))$, $(s,t) \in T_3$ (recall that at least one of T_1, T_2, T_3 must be nonempty). From the last variance computation we see that any of these centered Gaussian r.v.'s has variance at most $\frac{1}{4(1-\alpha)^2}$, hence, a β-quantile lower bound is given by $\frac{\Phi^{-1}(\beta)}{2(1-\alpha)} = -\frac{\Phi^{-1}(1-\beta)}{2(1-\alpha)}$. Combining this with Theorem 1.9 we see that

$$\liminf P\left(\sqrt{n}\Big(d_K(F_0, R_\alpha(F_n)) - d_K(F_0, R_\alpha(F))\Big) \ge -\tfrac{\Phi^{-1}(1-\beta)}{2(1-\alpha)}\right) \ge 1 - \beta.$$

Hence,

$$d_K(F_0, R_\alpha(F_n)) + \frac{\Phi^{-1}(1-\beta)}{2\sqrt{n}(1-\alpha)} \tag{1.22}$$

is an upper confidence bound with asymptotic confidence level at least $1-\beta$ for $d_K(F_0, R_\alpha(F))$.

In order to get a simple and manageable lower bound for the Kolmogorov distance we need to pay attention to the worst cases inside the maximum of the limiting random variable in Theorem 1.9. This means that we have to study the cases $T_1 = [0, a]$, $T_2 = [b, 1]$,

$T_3 = [a, b]$ where $0 \leq a \leq b \leq 1$. We have the following inequalities

$$\begin{aligned} Z_\alpha(F_0, F) &= \tfrac{1}{1-\alpha} \max\left(\max_{t\in[0,a]} B(t), \max_{t\in[b,1]}(-B(t)), \max_{s\in[a,b],t\in[s,b]} \tfrac{1}{2}(B(t) - B(s)) \right) \\ &\leq \tfrac{1}{1-\alpha} \max\left(\max_{t\in[0,a]} |B(t)|, \max_{t\in[b,1]} | - B(t)|, \tfrac{1}{2}\left(\max_{t\in[a,b]} B(t) + \max_{s\in[a,b]} -B(s) \right) \right) \\ &\leq \tfrac{1}{1-\alpha} \max\left(\max_{t\in[0,a]} |B(t)|, \max_{t\in[b,1]} | - B(t)|, \max\left(\max_{t\in[a,b]} B(t), \max_{s\in[a,b]} -B(s) \right) \right) \\ &= \tfrac{1}{1-\alpha} \max_{t\in[0,1]} |B(t)|. \end{aligned}$$

Now, denoting $\Psi(x) = P\left(\max_{t\in[0,1]} |B(t)| \leq x\right)$ the d.f. of Kolmogorov's distribution, we have

$$\limsup P\left(\sqrt{n}\Big(d_K(F_0, R_\alpha(F_n)) - d_K(F_0, R_\alpha(F)) \Big) \leq \tfrac{\Psi^{-1}(1-\beta)}{(1-\alpha)} \right) \geq 1 - \beta.$$

Hence,

$$d_K(F_0, R_\alpha(F_n)) - \frac{\Psi^{-1}(1-\beta)}{\sqrt{n}(1-\alpha)} \tag{1.23}$$

is a lower confidence bound with asymptotic confidence level at least $1 - \beta$ for $d_K(F_0, R_\alpha(F))$.

In the following example we will show that the, arguably conservative, confidence bounds just obtained can be precise in practice. Of course, an efficient estimation of the sets T_i could improve the precision of the coverage bands, but our simulations show that the rate of convergence can make highly unstable the estimation. In fact, Theorem 3.1 in [Álvarez-Esteban et al., 2016] addressed a simpler but similar problem involving the supremum of the difference of two independent Brownian bridges on the set where two d.f.'s attain their greatest distance.

Example 1.11. Coverage rates for extreme cases. *The bounds (1.22) and (1.23) are conservative. Nontheless, there are extreme cases for which the bounds are (almost) optimal. We present several examples of such cases in Table 1.2. For different combinations of F_0, F and α, we give the observed coverage frequency of the confidence bounds (1.23) and (1.22). Figure 1.4 shows the he d.f.'s of some of these examples to get a better notion of the functions of interest. For simplicity in all the considered cases we fix $F \sim U(0, 1)$. Then we consider instances of F_0 for which, approximately, $T_1 \cup T_2 \cup T_3$ equals $[0, 1]$. For this, we fix $0 \leq a \leq b \leq 1$ and define the piecewise linear function*

$$F_0^{a,b}(t) = \begin{cases} t/(1-\alpha) + d_\alpha & t \in [-(1-\alpha)d_\alpha, t_0] \\ a & t \in [t_0, t_1] \\ (t - (1+q)\alpha)/(1-\alpha) & t \in [t_1, (a+b)/2] \\ (t + q\alpha)/(1-\alpha) & t \in [(a+b)/2, t_2] \\ b & t \in [t_2, t_3] \\ (t-\alpha)/(1-\alpha) - d_\alpha & t \in [t_3, 1 + (1-\alpha)d_\alpha] \end{cases}$$

where we take $q \in (0, 1)$ such that $d_\alpha = \frac{(2+q)\alpha}{2(1-\alpha)} < 1$ and define $t_0 = (1-\alpha)(a - d_\alpha), t_1 = (1-\alpha)a + (1+q)\alpha, t_2 = (1-\alpha)b - q\alpha, t_3 = (1-\alpha)(b + d_\alpha) + \alpha$ (Figure 1.4 depicts

$F_0^{0.01,0.99}$ and $F_0^{1/3,2/3}$ for $q = 0.1$ and $\alpha = 0.05$) . It is straightforward to check that $d_K(F_0^{a,b}, R_\alpha(F)) = d_\alpha$, and that $T_1 = [0, t_0]$, $T_2 = [t_3, 1]$ and $T_3 = [t_1, t_2]$. We note that T_1 becomes close to $[0, a]$, T_2 to $[b, 1]$ and T_3 to $[a, b]$ as $\alpha \to 0$.

For different extreme behaviour we take F_0 to be the d.f. of a $Beta(1, \beta_0)$ distribution with β_0 such that $f_0(1/2) = 1/(1-\alpha)$ (this is possible for $\alpha < 0.06148$). We obtain $d_K(F_0, R_\alpha(F)) = P(Beta(1, \beta_0) \leq 1/2) - (1/2)/(1-\alpha)$, $T_1 = \{1/2\}$ and $T_2 = T_3 = \emptyset$. Figure 1.4 includes the d.f. of $Beta(1, 1.637464)$ (corresponding to $\alpha = 0.05$). Finally, another extreme case follows by fixing $d_\alpha \in (0, 1)$ and defining

$$F_0^{0.5}(t) = \begin{cases} (1/(1-\alpha) + 2d_\alpha)t & t \in [0, 1/2] \\ ((1-2\alpha)/(1-\alpha) - 2d_\alpha)t + (\alpha/(1-\alpha) + 2d_\alpha) & t \in [1/2, 1]. \end{cases}$$

It is immediate that $d_K(F_0^{0.5}, R_\alpha(F)) = d_\alpha$, $T_1 = \{1/2\}$ and $T_2 = T_3 = \emptyset$. In Figure 1.4 we included the case for $d_\alpha = 0.1$.

Remark 1.12. *Notice that $F_0^{a,b}$ is not continuous in $(a+b)/2$ and is not differentiable in t_0, t_1, t_2 and t_3, also, $F_0^{0.5}$ is not differentiable in $1/2$. However, it is possible to modify these functions in such a way that from the point of view of simulation their behaviour becomes indistinguishable. This is why we keep the simple versions that give a better intuitive idea of what is happening.*

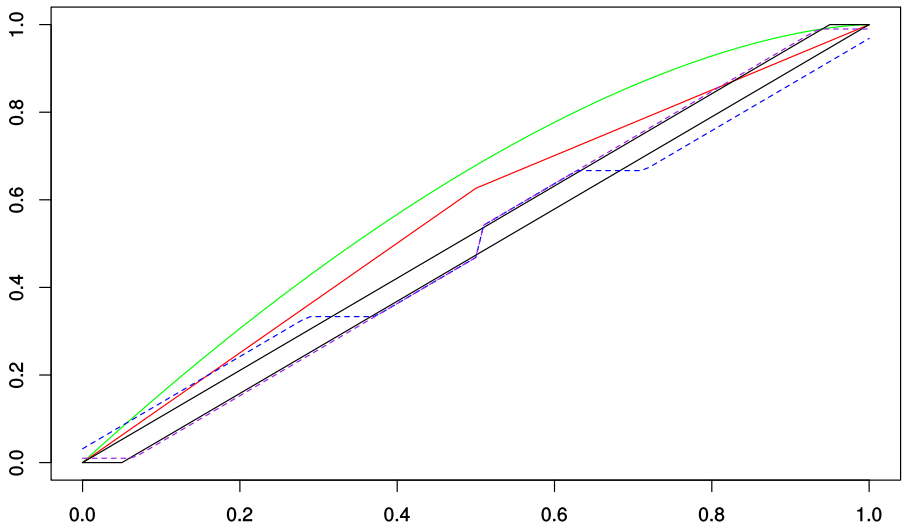


Figure 1.4: In green $Beta(1, 1637464)$; in red $F_0^{0.5}$ with $d_\alpha = 0.1$; in dashed blue $F_0^{1/3,2/3}$; in dashed purple $F_0^{0.01,0.99}$; in black the maximum and minimum (in the usual stochastic order sense) of the trimmings of $U(0, 1)$. We fix $\alpha = 0.05$ and $q = 0.1$.

Table 1.2: For the first two examples we fix $\alpha = 0.05$ and get $\beta_0 = 1.637464$. For $F_0^{0.5}$ we take $d_\alpha = 0.1$. For all the other examples $\alpha = 0.01$ and $q = 0.01$, where the first row indicates the values (a, b) for $F_0^{a,b}$. For each example, we generate $M = 1200$ samples of size N from $F \sim U(0, 1)$.

	$Beta(1, \beta_0)$	$F_0^{1/2}$	(0.01, 0.99)	(0.49, 0.51)	(1/3, 4/3)	(0.01, 0.5)	(0.6, 0.8)
N = 100	0.985	0.993	0.957	0.955	0.947	0.949	0.942
	0.998	1.000	1.000	1.000	1.000	1.000	1.000
N = 1000	0.988	0.989	0.981	0.958	0.968	0.970	0.968
	0.992	0.976	1.000	1.000	1.000	1.000	1.000
N = 5000	0.993	0.996	0.998	0.960	0.973	0.970	0.958
	0.980	0.963	1.000	1.000	1.000	1.000	1.000

1.4.2 Applications to credibility analysis

As already noted, for large enough sample sizes a classical goodness-of-fit test would reject the null hypothesis in (1.10) and yet we could be interested in knowing how well F_0 describes the generating mechanism behind the data. More about this idea of resemblance, understood as similarity between generated samples and the data can be found in [Davies, 1995]. In a similar spirit, a *model credibility index* was introduced in [Lindsay and Liu, 2009]. In short, for a fixed $\delta \in (0, 1)$, and a given test of fit to a model, the δ-credibility index is the sample size for which (for samples coming from the same random generator as the data) the model is rejected with probability δ. In the setting of the testing problem (1.10) with rejection rule $d_K(F_0, R_\alpha(F_n)) > \lambda\rho_n$, the credibility index is the sample size N_δ (note the dependence of α) for which

$$P\Big(d_K(F_0, R_\alpha(F_{N_\delta})) > \lambda\rho_{N_\delta}\Big) = P\left(d_K(F_0, R_\alpha(F_{N_\delta})) > \frac{\lambda\rho}{\sqrt{N_\delta}}\right) = \delta. \tag{1.24}$$

Since the underlying random generator is unknown, N_δ cannot be computed. Subsampling techniques were proposed in [Lindsay and Liu, 2009], considering the estimator $N_{\delta,subs}$ as the sample size such that when we take M subsamples of that size the rejection frequency of the null is δ. Drawbacks of this procedure include that it is accurate only when N_δ is small compared to the original sample size, that $N_{\delta,subs}$ can never be bigger than that sample size and that the procedure is computationally demanding. We will try to address these shortcomings, while still maintaining the nice intuitive interpretation associated to the credibility index.

We start by writing

$$\begin{aligned} &P\Big(d_K(F_0, R_\alpha(F_{N_\delta})) > \lambda\rho_{N_\delta}\Big) \\ &\quad = P\Big(\sqrt{N_\delta}\Big(d_K(F_0, R_\alpha(F_{N_\delta})) - d_K(F_0, R_\alpha(F))\Big) > \sqrt{N_\delta}\Big(\lambda\rho_{N_\delta} - d_K(F_0, R_\alpha(F))\Big)\Big) \\ &\quad = P\left(\sqrt{N_\delta}\Big(d_K(F_0, R_\alpha(F_{N_\delta})) - d_K(F_0, R_\alpha(F))\Big) > \lambda\rho - \sqrt{N_\delta}d_K(F_0, R_\alpha(F))\right) \end{aligned} \tag{1.25}$$

Using Theorem 1.9, asymptotically we can look for

$$P\Big(\max\big(\max_{t\in T_1}\tilde{B}(t),\max_{t\in T_2}-\tilde{B}(t),\tfrac{1}{2}\max_{(s,t)\in T_3}(\tilde{B}(t)-\tilde{B}(s))\big)>\lambda\rho-\sqrt{N_\delta}d_K(F_0,R_\alpha(F))\Big)\Big), \tag{1.26}$$

where we keep using N_δ for our approximation in the asymptotic regime of the original N_δ.

Next, we define a lower and an upper estimate for N_δ using the probability bounds for $Z_\alpha(F_0,F)$ in Subsection 1.4.1. Thus, we define L_δ from

$$P\left(\frac{1}{1-\alpha}\max_{t\in[0,1]}|B(t)|>\lambda\rho-\sqrt{L_\delta}d_K(F_0,R_\alpha(F))\right)=\delta$$

and, similarly, U_δ from

$$P\left(N\left(0,\frac{1}{4(1-\alpha)^2}\right)>\lambda\rho-\sqrt{U_\delta}d_K(F_0,R_\alpha(F))\right)=\delta.$$

Equivalently,

$$L_\delta=\Big(\frac{\lambda\rho-\Psi^{-1}(\delta)/(1-\alpha)}{d_K(F_0,R_\alpha(F))}\Big)^2,\ U_\delta=\Big(\frac{\lambda\rho-\Phi^{-1}(\delta)/(2(1-\alpha))}{d_K(F_0,R_\alpha(F))}\Big)^2. \tag{1.27}$$

and it follows easily that $N_\delta\in[L_\delta,U_\delta]$. We see also that the empirical estimators $L_{\delta,n}$ and $U_{\delta,n}$, built replacing F by F_n in (1.27), are consistent estimators of L_δ and U_δ, respectively.

We end this section discussing on the practical use of the tK-index of fit, α^*, introduced in (1.2). A consistent estimator α_n^* would provide an intuitive measure of proximity of the model to the data, assessing to what extent the data can be considered a contaminated sample from the model F_0. Recalling the setting of the testing problem (1.10) and the subsequent discussion, we would reject the null hypothesis if $d_K(F_0,R_\alpha(F_n))>\lambda\rho_n$. This suggests to consider α_n^* as the smallest of the solutions of the equation

$$\sqrt{\frac{1}{2n}\log\frac{2}{\epsilon_1}}=(1-\alpha)d_K\left(F_0,R_\alpha(F_n)\right)\quad\text{if}\quad\sqrt{\frac{1}{2n}\log\frac{2}{\epsilon_1}}<d_K\left(F_0,F_n\right), \tag{1.28}$$

and $\alpha_n^*=0$ whenever $\sqrt{\frac{1}{2n}\log\frac{2}{\epsilon_1}}\geq d_K\left(F_0,F_n\right)$. This goal is feasible by numerical methods, allowing the use of α_n^* in practice. Moreover, from (1.28) and Proposition 1.2, α_n^* is almost surely consistent. The carried simulations show that α_n^* converges rather slowly to the theoretical value. In fact, there are connections between this estimator and those considered in the FDR setting (see [Genovese and Wasserman, 2004]), that justify this slow convergence rate even in the DCN case. Since a lower bound for α^* is a main goal in FDR analysis, we will deserve some comparisons in Section 1.5.

1.5 Relations with the FDR setting

To our effects, the False Discovery Rate model essentially assumes a **dominated contamination model** (DCN) like (3) in the Introduction, $F=(1-\alpha)F_0+\alpha F'$, where F'

(so F) must be stochastically dominated by F_0. Recall that the stochastic order $F' \leq_{st} F_0$ is defined by the relation $F'(x) \geq F_0(x)$ for all $x \in \mathbb{R}$. The DCN assumption notably simplifies the FDR theory (which can be based on one-sided statistics), but the methodology developed in this chapter can be useful for applications in FDR in which, as often happens, the DCN can be hardly justified. To appreciate the differences between the general framework of CN's and DCN's, it seems worthwhile to take advantage of the analyses in Examples 1.1 and 1.2.

Example 1.13. Dominated contamination neighbourhoods. *In the scenarios considered in Example 1.1, only the second case of i) presents a dominated contamination* $F = (1-\varepsilon)F_0 + \varepsilon F'$, *with* $F' \leq_{st} F_0$. *In fact, between the d.f.'s* F' *of* $U(a,b)$ *laws, only those verifying* $a \leq \inf\{0,b\}$ *and* $b \leq 1$ *are stochastically dominated by* F_0. *Therefore, considering*

$$R_\alpha^-(F,F_0) := \{F\} \cup \{F' \in R_\alpha(F) : F' \leq_{st} F_0\}, \tag{1.29}$$

it holds $d_K(F_0, R_\alpha^-(F,F_0)) = \varepsilon - \alpha$ *if* F *is the d.f. of the* $U(-\varepsilon,1)$ *law and* $0 \leq \alpha \leq \varepsilon$; $R_\alpha^-(F,F_0) = \{F\}$ *for* $0 \leq \alpha < 1$ *under ii), while under iii):* $R_\alpha^-(F,F_0) = \{F\}$ *for* $0 \leq \alpha < \varepsilon$ *and* $R_\alpha^-(F,F_0) = \{F_0\}$ *for* $\varepsilon \leq \alpha < 1$. *This shows that, in presence of non-dominated contamination, trimming under the restricting domination scheme does not necessarily improve the approximation (measured under any metric).*

In the Gaussian model, stochastic dominance $N(\mu_1,\sigma_1^2) \leq_{st} N(\mu_2,\sigma_2^2)$ *is equivalent to* $\mu_1 \leq \mu_2$ *and* $\sigma_1 = \sigma_2$. *Thus only normal distributions with* $\sigma = 1$ *and* $\mu \leq 0$ *are dominated by a* $N(0,1)$ *law. For fixed* α, *solving the relation (1.6)*$= 0$ *would give the set of normal distributions that are dominated contamination versions of the* $N(0,1)$ *law. Therefore the only normal law in a DCN of a normal law is the same Gaussian. In particular, in the examples considered in Figure 1.2, only the point* $(1,0)$ *belongs to the DCN. Of course, non-normal distributions like the mixtures* $(1-\alpha)N(0,1)+\alpha N(\mu,1)$ *for any* $\mu < 0$, *would belong to such a DCN.* □

Regarding the hypothesis testing problem and Theorem 1.6, note the very different nature of the problems of interest in the FDR setup: the control of the false discovery rate through a confidence lower bound and the detection of the particular false hypotheses. Resorting to a simplified version, the problem would be described through the DCN as $F = (1-\alpha)F_0 + \alpha F'$, where F_0 is the f.d. of the $U(0,1)$ law and F' is a d.f. with support on $(0,1)$ and $F'(x) \geq x$ for every $x \in (0,1)$. The null would be $\alpha = 0$, and the alternative would be $\alpha > \alpha_0$. Acceptance of the null hypothesis with our testing procedure, for a given α, under the DCN setting would indicate that a lower proportion than α false hypotheses are compatible with our data.

In the FDR setting, estimation and confidence intervals for the contamination level are main objectives. In fact, there are connections between the estimator defined in (1.28) and those considered in the FDR setting (see [Genovese and Wasserman, 2004]), that justify the slow convergence rate even in the DCN case. Since a lower bound for α^* is a main goal in FDR analysis, some comparison is in order, but previously we will introduce a new estimate.

It is easy to see that, (4) is also equivalent to $P(B) \geq (1-\alpha)P_0(B)$ for any Borel set $B \subset \mathbb{R}$ and to $P(B) \leq (1-\alpha)P_0(B) + \alpha$ for any such set. Moreover, the Borel sets in $\mathbb{R}$ can be arbitrarily well approximated by finite unions of disjoint intervals. From these

considerations, we could use of the bound

$$\alpha \geq \alpha(P, P_0) := 1 - \inf \left\{ \frac{P(J)}{P_0(J)},\ J \text{ intervals in } \mathbb{R} \right\}, \tag{1.30}$$

noting that $\alpha(P, P_0)$ is a semicontinuous statistical functional in the sense of [Donoho, 1988], allowing the obtention of nontrivial lower confidence bounds for α. This suggests that the combination of CN with the distance of Kuiper, $d_{Kuiper}(P, Q) := \sup\{|P(J) - Q(J)|, J \text{ interval in } \mathbb{R}\}$, could be more natural that the d_K distance. That goal deserves future work, but now we devote some attention to another, novel (Bonferroni type) lower confidence bound for $\alpha(P, P_0)$, thus for α^*:

$$\hat{\alpha}_k = 1 - \min_{(i,j)} \frac{\beta^{-1}_{j-i,n+1-j+i}(1 - \gamma/M_n)}{P_0(\left[X_{(i)}, X_{(j)}\right])}. \tag{1.31}$$

Here $1 - \gamma$ is the confidence level, $\beta^{-1}_{k,l}$ denotes the quantile function of the $Beta(k, l)$ distribution, and the minimum is taken over all $M_n = n(n + 3)/2$ index pairs (i, j) such that $0 \leq i < j \leq n + 1, j - i \leq n$.

Although not implemented here, we should mention that the bound could be refined in two ways: Replace the Bonferroni quantiles

$$\beta^{-1}_{j-i,n+1-j+i}(1 - \gamma/M_n) = 1 - \beta^{-1}_{n+1-j+i,j-i}(\gamma/M_n)$$

with

$$\beta^{-1}_{j-i,n+1-j+i}(1 - \gamma_n) = 1 - \beta^{-1}_{n+1-j+i,j-i}(\gamma_n),$$

where γ_n is the exact γ-quantile of the distribution of

$$\begin{aligned} &1 - \max_{0 \leq i < j \leq n+1 : j-i \leq n} \beta^{-1}_{j-i,n+1-j+i}(U_{(j)} - U_{(i)}) \\ &= \min_{0 \leq i < j \leq n+1 : j-i \leq n} \beta^{-1}_{n+1-j+i,j-i}(1 - U_{(j)} + U_{(i)}) \end{aligned}$$

with the order statistics $0 = U_{(0)} < U_{(1)} < \cdots < U_{(n)} < U_{(n+1)} = 1$ of a random sample from the $U([0, 1])$ distribution. Furthermore, since the small intervals are more important than the large ones, one could restrict attention to all pairs (i, j) of indices $0 \leq i < j \leq n + 1$ such that $j - i \leq d_n$, with $d_n = \lfloor n/2 \rfloor$, say. This means, one would consider $M_n = ((2n + 3)d_n - d_n^2)/2$ pairs (i, j).

Example 1.14. Some comparisons between estimates of α^*. *In Figure 1.5 we compare the behaviour of our estimate α^*_n (based on $\epsilon_1 = 0.05$) of α^* with some confidence lower bounds, associated to bounding functions, as described in [Meinshausen and Rice, 2006]. We denote by $\hat{\alpha}_c$ to the lower bound with confidence level 0.95, i.e., $P(\hat{\alpha}_c \leq \alpha) \geq 0.95$, associated to the constant bounding function $\delta(t) = 1$; $\hat{\alpha}_l$ is the one associated to the linear bounding function $\delta(t) = t$; $\hat{\alpha}_s$ is obtained with the standard deviation-proportional bounding function $\delta(t) = \sqrt{t(1 - t)}$. The legend in the figure explains the way in which the corresponding samples have been obtained. Let X_0 be a random variable with a $N(0, 1)$ law and recall that Φ denotes its d.f.. In the graphics of the first row, we take $X_1 = \Phi(X_0)$ and $Y_1 = \Phi(X_0 + 4)$; in those of the second row, $X_2 = \Phi(X_0), Y_2 = \Phi(3X_0 + 4)$. In the third row, we consider X_3 with a $U(0, 1)$ law and Y_3 with a $Beta(5, 1)$ law.*

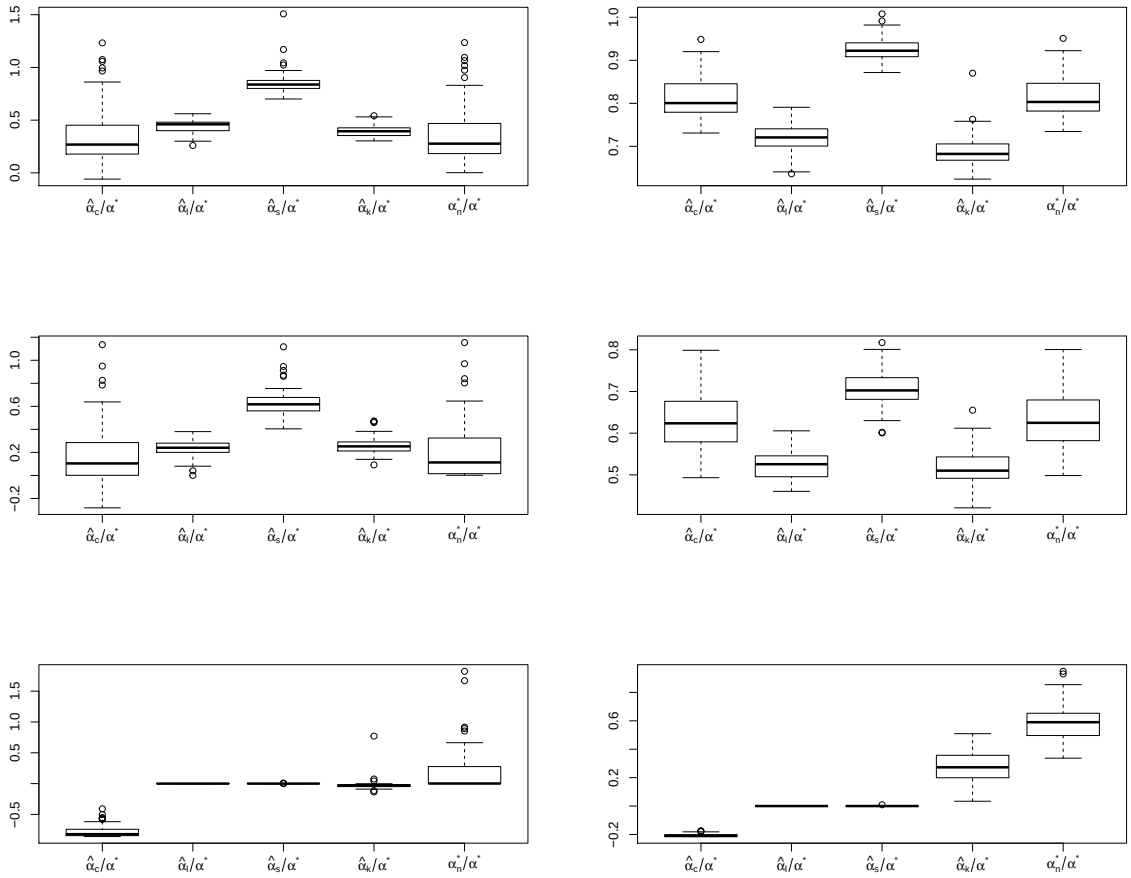


Figure 1.5: In the graphics on the left (resp. right) column we use $\alpha^* = 0.05$ (resp. $\alpha^* = 0.2$). Every graphic is based on 100 samples of size 1000 obtained joining independent samples, one of size $1000(1-\alpha^*)$ of a random variable X_i and other of size $1000\alpha^*$ of another Y_i for respective rows $i = 1, 2, 3$. The estimates $\alpha_n^*, \hat{\alpha}_c, \hat{\alpha}_l, \hat{\alpha}_s, \hat{\alpha}_k$ and laws of X_i and Y_i are described in Example 1.14

The first row is a very favourable case for the procedures shown in Section 4 in [Meinshausen and Rice, 2006]. The second is a perturbation of that, allowing greater dispersion on the contamination, thus breaking the domination. In the lower row, we present a case where the procedures described in [Meinshausen and Rice, 2006] do not give meaningful bounds, while our procedure gives sensible results. We note that [Meinshausen and Rice, 2006] seemingly do not use the DCM assumption but, as it is apparent from the pictures in Figure 1.5, in fact their proposals are not meaningful when that condition fails. We may conclude that our estimate is competitive when we are under the assumptions of [Meinshausen and Rice, 2006], but also works when these assumptions fail. □

1.6 Simulations and a real data example

1.6.1 A toy example

Let us explore the practical use of our tools to evaluate the quality of a given model on the basis of the sample. We fix the model $F_0 \sim N(0,1)$ and consider three different large samples ($n = 20000$) simulated from three different distributions. The first sample comes from $F_1 \sim Logistic(0, \sqrt{3}/\pi)$, with the same mean (=0) and variance (=1) as F_0. We notice that this model distribution has been reported in [Lindsay and Liu, 2009] as generating datasets very close to 'normality'. The other two samples come from contaminated normal distributions $F_2 \sim 0.867N(0,1) + 0.133N(0,4^2)$ and $F_3 \sim 0.9N(0,1) + 0.1N(3,1)$ (we will refer to these samples as contaminated by 'inliers' and 'outliers', respectively).

Our first step is to assess whether these samples can be assumed as coming from a contamination of level at most $\alpha = 0.05$ of the model F_0. We fix $\epsilon_2 = 0.05$ and $\epsilon_1 = 0.05/(0.999e^2)$. We note (recall the discussion about the testing problem (1.10)) that this choice of ϵ_1 is very close to the minimal admissible value for the validity of (1.12) and (1.13), that is, we are taking a very conservative approach, rejecting the null only if we have very strong evidence against it. From (1.12) and (1.13) we see that this amounts to fixing $\lambda = 0.9997$ and $\rho_{2\times10^4} = 0.012555$, the null being rejected if $d_K(F_0, R_{0.05}(F_{i,2\times10^4})) > \lambda\rho_{2\times10^4} = 0.012552$. The first column in Table 1.3 reports the observed values of $d_K(F_0, R_{0.05}(F_{i,2\times10^4}))$, $i = 1,2,3$. Despite the very conservative approach taken, the null is rejected for the three samples, that is, we should not consider them as (0.05) contaminated samples from our model F_0.

Table 1.3: For $F_0 \sim N(0,1)$, $F_1 \sim Logistic(0, \sqrt{3}/\pi)$, $F_2 \sim 0.867N(0,1) + 0.133N(0,4^2)$ and $F_3 \sim 0.9N(0,1) + 0.1N(3,1)$, the table shows the results obtained from samples of size $n = 20000$. We denote $d_{K,n} = d_K\left(F_0, R_{0.05}(F_{i,n})\right)$, $d_{K,95\%}$ are the 95% lower (top) and upper (bottom) confidence bounds for $d_K\left(F_0, R_{0.05}(F_i)\right)$.

	$d_{K,n}$	$d_{K,95\%}$	$N_{0.5,indep}$	$L_{0.5,n}$	$U_{0.5,n}$	$N_{0.5,subs}$	α_n^*
F_1	0.0140	0.0000	12370	4170	16079	15670	0.054
		0.0262					
F_2	0.0200	0.0000	7610	2045	7886	6840	0.069
		0.0322					
F_3	0.0477	0.0275	1135	359	1386	1020	0.089
		0.0599					

Next, we try to assess the quality of the rejected model as a good description of the underlying distributions of the samples. The simplest approach could be to use the estimated d_K distance. Looking back at the first column of Table 1.3 we see that F_1 is closer to the rejection boundary than F_2, and the later is closer than F_3. This estimate is complemented by the 95% lower (top cell) and upper (bottom cell) confidence bounds for $d_K(F_0, R_{0.05}(F_i))$, included in the second column of the table. We see, for instance, that the generator of the first sample is (with 95% confidence) at small d_K distance (0.0262)

from an α-contamination of the standard normal distribution.

Alternatively, we could consider credibility indices, looking for the sample sizes from the generators that will be suitably represented by the model plus the corresponding CN (we keep our choice of $\alpha = 0.05$). The estimators $L_{0.5,n}$ and $U_{0.5,n}$ are reported in the fourth and fifth column of Table 1.3, we expect the credibility index to be in the interval $[359, 1386]$ for F_3, in $[2045, 7886]$ for F_2 and in $[4170, 16079]$ for F_1. Once more F_1 is the closest to the model in this sense, followed by F_2 and then F_3. We also see that, from a conservative point of view, F_1 can generate samples of size up to 4170 while rejecting the null less than 50% of the time. Therefore, at least in this sample size range F_0 can be considered as a useful model for the data (allowing 5% contamination).

In this controlled setup we can use our knowledge of the underlying distributions of the samples to estimate the true credibility index, $N_{0.5}$. The index $N_{0.5,indep}$ denotes the sample size for which 5000 independent samples of that size from the true generator, give a rejection frequency of 50%. $N_{0.5,subs}$ is the subsampling approximation to the credibility index described in Section 1.4.2. We see in Table 1.3 that the interval $[L_{0.5,n}, U_{0.5,n}]$ in all three cases contains $N_{0.5,indep}$ and $N_{0.5,subs}$, as expected.

A last way of comparison is given by α_n^*. As before, F_1 is closest to the model ($\alpha_n^* = 0.054$), then comes F_2 ($\alpha_n^* = 0.069$), and last F_3 ($\alpha_n^* = 0.089$). This suggests that the random generators of the samples are not too far from the model, F_0. On the other hand, $F_0 \in R_{0.1}(F_2)$ and $F_0 \in R_{0.1}(F_3)$ and in both cases we have $\alpha^* = 0.1$. Note, in this respect, the slow convergence of α_n^* showed in the last column of Table 1.3.

To summarize, up to some 'small' contamination (0.05), the logistic generated sample is the closest one to normality. It is closer to normality than samples coming from 0.1-contaminations of the normal model. Also, scale contaminations with the same mean (F_2), generate samples that 'look' more normal than location contaminations, when allowing some (0.05) trimming.

1.6.2 Trying a real data example

Here we analyse the heights of 52402 individuals with ages between 2 and 84. The data has been obtained from NHANES (https://www.cdc.gov/nchs/nhanes/) and consists of height measurements (in centimeters) of 26625 females and 25777 males. The dataset analysed here is available at

https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/SHBF2G

We consider three age groups, which are related to human body development. The first group includes kids before puberty (ages between 2 and 10). The second group, puberty period, includes individuals aged between 11 and 18, with adults (over 18) making the third group. We start analysing the adult group (30284 individuals). The data consists of height measurements on 15679 females and 14605 males.

Notice that in our analysis we will use the population estimates of the mean and variance. This is very usual in the goodness of fit setting based on procedures designed for testing simple hypothesis and, in particular in the FDR setting. There, the $U(0, 1)$ law, considered as the hypothesis, arises from the integral, or p-value transformation, but it depends on the (unknown) true distribution. In our framework, that license is even more permisible because we are interested in getting a useful description of the data.

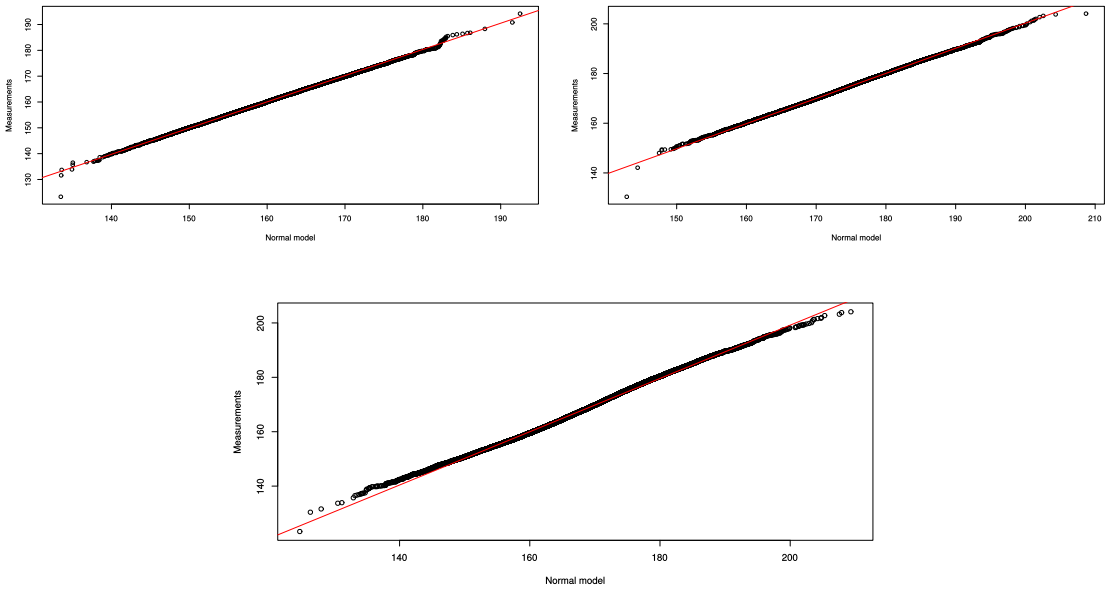


Figure 1.6: QQ-plots of the measured heights of 15679 females (left), 14605 males (right), and of the combined joint sample (below) against a Normal distribution with the same mean and variance as the corresponding data set.

We analyse first the sample by gender group. In Figure 1.6 we see qq-plots from normal distributions with the same mean and variance as the female's and male's heights data. The pictures suggest that the normal model could provide a reasonable description of the data. Also, a Kolmogorov-Smirnov test of normality yields a p-value of 0.5385 for the male group and of 0.2997 for the female group, thus we do not get enough evidence to reject that the data sets come from normal random generators.

Next we take a look at the combined data set. The previous analysis suggests the model $F_0 \sim 0.52N(161.0, 7.1^2) + 0.48N(174.6, 7.9^2)$. If, however, we perform a gender-blind analysis and take a look at the qq-plot in the third graphic in Figure 1.6 for the combined sample, we may be tempted to say that the normal distribution is not a bad model for the data (nevertheless, the K-S normality test yields a p-value smaller than 10^{-16}). After the discussion in the previous sections, we could yet stick to the normal model and consider $F_0^* \sim N(167.6, 10.1^2)$ hoping a useful description of the random generator of the data.

Figure 1.7 shows the empirical d.f. together with the models F_0 and F_0^*. While the gender-blind model, which is in disadvantage (since it is blinded to relevant information), is further away from the data than F_0 we may wonder how bad is F_0^* as a model. If trimming is allowed, we would need a 6% trimming to avoid rejection of the null hypothesis, i.e., $d_K(F_0^*, R_{0.06}(F)) = 0$ would not be rejected, thus $\alpha_n^* = 0.06$, and our data are compatible with a generation from F_0^* with a proportion of until 6% wrong data. Actually, F_0 is still a better model, since $d_K(F_0, R_{0.06}(F_n)) = 0.00231$ with 95% confidence interval for $d_K(F_0, R_{0.06}(F))$ of $[0, 0.01468]$, while $d_K(F_0^*, R_{0.06}(F_n)) = 0.01026$ with a confidence interval for $d_K(F_0^*, R_{0.06}(F))$ of $[0, 0.02264]$. We could even look for other normal distri-

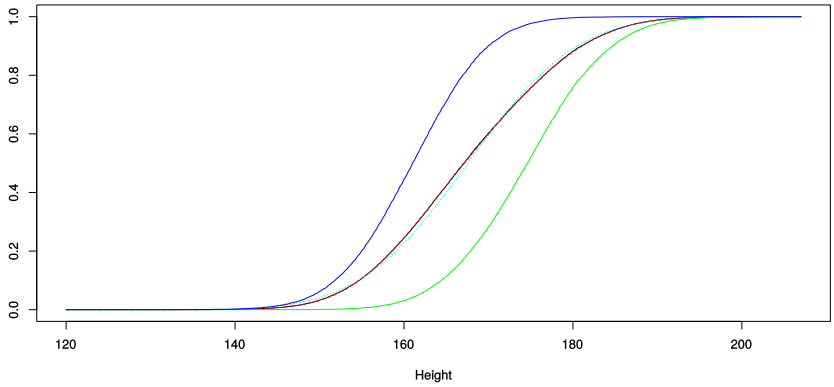


Figure 1.7: Comparison of data and models. Solid lines correspond to the empirical d.f.'s: black for the joint samples, and blue (resp. green) for females (resp. males). F_0 is represented in dashed red and F_0^* in dotted cyan.

butions inside the tolerance region, shown in green in Figure 1.8, and choose one as a sensible model. Alternatively, if we find this trimming level unacceptable, we may try to use smaller CN's and asses model adequacy using credibility analysis.

Table 1.4: $d_{K,\alpha,n} = d_K\left(F_0^*, R_\alpha(F_n)\right)$, where $F_0^* \sim N(167.6, 10.1^2)$ and F is the true generating mechanism. $N_{0.5,subs}$ is obtained taking 1000 sub-samples of the heights data. $N_{0.5,indep}$ is obtained taking 1000 independent samples from $F = F_0$.

α	$d_{K,\alpha,n}$	$d_{K,\alpha,95\%}$	$N_{0.5,\text{indep}}$	$L_{0.5,n}$	$U_{0.5,n}$	$L_{0.5,95\%}$	$U_{0.5,95\%}$	$N_{0.5,\text{subs}}$
0.015	0.0184	0.0000	8350	2239	8631	832	3206	7225
		0.0302				∞	∞	
0.035	0.0143	0.0000	17250	3888	14993	1143	4407	13280
		0.0263				∞	∞	
0.055	0.0110	0.0000	37300	6851	26417	1523	5872	25810
		0.0233				∞	∞	

The output of this type of analysis is reported in Table 1.4. We have fixed $\epsilon_1 = 0.01$ (recall the discussion leading to (1.13)) and considered three different trimming levels ($\alpha = 0.015, 0.035$ and 0.055, leading to optimal rejection boundaries $\lambda\rho_{30284} = 0.0098, 0.0100$ and 0.0103, respectively). Testing for these contamination levels results in rejection of the null hypothesis, since the values in the first column of Table 1.4 are above the respective rejection boundaries. But we see how the empirical trimmed Kolmogorov distance approaches the rejection boundary as the trimming level increases. Further informative values are provided by the intervals $[L_{0.5,n}, U_{0.5,n}]$. With the available data, since we expect N_δ to be in $[L_{0.5,n}, U_{0.5,n}]$, we see how reasonably the gender-blind model, F_0^*, represents the data. With a conservative point of view, after trimming only 0.055, samples of the true generator of size 6851 will not be rejected as coming from F_0^* (more than 50% of the time). If we take an optimistic point of view, we can say the same thing but for a sample size of 26417. As in our toy example, we see that $N_{0.5,subs} \in [L_{0.5,n}, U_{0.5,n}]$,

therefore our estimated interval for the credibility index contains the estimation proposed in [Lindsay and Liu, 2009]. If, on the other hand, we admit that the data comes from F_0 and calculate the estimate $N_{0.5,indep}$, we see that $N_{0.5,subs}$ is far from $N_{0.5,indep}$ and our upper bounds $U_{0.5,n}$ get closer to $N_{0.5,indep}$. Furthermore, we could plug-in our upper and lower confidence bounds (1.22) and (1.23) into (1.27) to get upper and lower confidence bounds for $L_{0.5}$ and $U_{0.5}$. These are reported in the columns labeled $L_{0.5,95\%}$ and $U_{0.5,95\%}$. We can assure with more than 95% confidence that $N_{0.5} \geq 1143$ for $\alpha = 0.035$ and, similarly, that $N_{0.5} \geq 1523$ for $\alpha = 0.055$.

Finally, we study the normality of the data for grouping ages. Using the same mean and variance as the data, we propose $F_1 \sim N(116.9, 18.2^2)$ for the age group under 11, $F_2 \sim N(163.1, 10.9^2)$ for the ages 11 and 18, and $F_3 (= F_0^*) \sim N(167.6, 10.1^2)$ for ages over 18. The tK-index of fit allows us to compare how normal is the data in each age group. We obtain the following indices: $\alpha^*_{1,n} = 0.3665$, $\alpha^*_{2,n} = 0.0057$ and, as before, $\alpha^*_{3,n} = 0.06$. This gives a clear 'normality' ranking. Somewhat surprisingly the data from the puberty group (ages 11 to 18) is almost normal. The adult group is close to normality and the children group is very far from normality. We emphasize that normality is rejected for each data set by a K-S test. To gain some intuition of what is really happening, we plot in Figure 1.8 the tolerance region for the normal family inside each respective CN for $\alpha^*_{2,n}$ and $\alpha^*_{3,n}$. The plot shows remarkably well how much closer to being normally distributed is the data of the teenagers compared to the adult group.

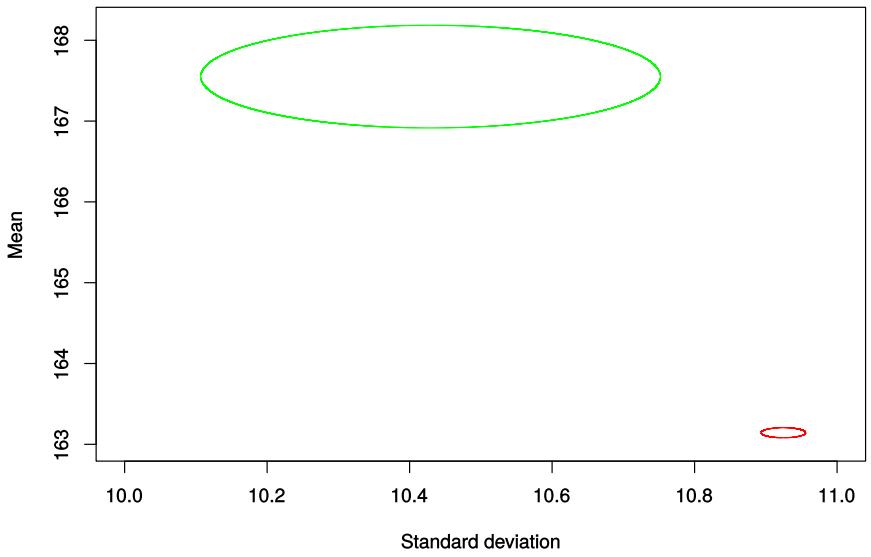


Figure 1.8: Tolerance regions for the normal family based on F_2 and F_3 in Section 1.6.2. In red the tolerance region that is inside $\mathcal{V}_{0.0057}(N(163.1, 10.9^2))$, in green the one that is inside $\mathcal{V}_{0.06}(N(167.6, 10.1^2))$.

1.7 Proofs

In this section we will provide the proofs of the main results of the chapter. In particular we will prove Lemma 1.3 which is a fundamental characterization as a variational problem of the trimmed Kolmogorov distance. For readability we restate here that result. By taking $\Gamma = F_0(F^{-1})$, F_0 and F being the distribution functions of P_0 and P, with great generality, the following identity holds:

$$d_K(P_0, R_\alpha(P)) = \min\{\|h - \Gamma\|, h \in \mathcal{C}_\alpha\}, \tag{1.32}$$

$$\text{where} \quad \mathcal{C}_\alpha := \{h : [0,1] \to [0,1] \text{ with } h(0) = 0, h(1) = 1, \text{ and } \|h\|_{\text{Lip}} \leq 1/1 - \alpha\}. \tag{1.33}$$

Here, as will be used throughout this section, for any real valued mapping $f : \aleph \to \mathbb{R}$ defined on a metric space $(\aleph, d)$, with $\|f\|$ and $\|f\|_L$ we will denote the L_∞ and the Lipschitz norms:

$$\|f\| = \sup_{x \in \aleph} |f(x)|, \qquad \|f\|_{\text{Lip}} = \sup_{x,y \in \aleph} \frac{|f(x) - f(y)|}{d(x,y)}.$$

The representation in (1.32) translates the problem of best trimmed approximation in Kolmogorov distance into finding a useful expression for a best L_∞-approximation to a monotone function by monotone, Lipschitz-continuous functions verifying the boundary conditions $h(0) = 0, h(1) = 1$.

In Section 1.7.1 we give the proof of Proposition 1.1 which provides the topological properties of trimmings in the L_∞ setting. We also provide a proof of the variational representation (Lemma 1.3).

In section 1.7.2 we provide the solution to the variational problem of obtaining the trimmed Kolmogorov distance, hence prooving Theorem 1.4. We do this by showing how the Pasch-Hausdorff envelopes (see [Rockafellar and Wets, 2009]) of a monotone function preserve monotonicity and provide the basis to build a best L_∞-approximation verifying the boundary constraints. We will also relate this process with the alternative way of obtaining the Ubhaya's monotone L_∞-best approximation (see [Ubhaya, 1974a, Ubhaya, 1974b]) to the Lipschitz regularization of the objective function.

In section 1.7.3 we provide the proof of our central limit theorem given in Theorem 1.9. The proof essentially follows from a result on directional differentiability of the L_∞-distance to the regularized version given in Theorem 1.21.

1.7.1 The set of trimmings in the L_∞-topological setting

For the sake of the reader we recall that the set of α-trimmings of F can be characterized as

$$R_\alpha(F) = \{F_h = h \circ F : h \in \mathcal{C}_\alpha\}.$$

We also recall that the trimmed Kolmogorov distance from F to F_0 is

$$d_K(F_0, R_\alpha(F)) := \inf_{\tilde{F} \in R_\alpha(F)} \|\tilde{F} - F_0\| = \inf_{h \in \mathcal{C}_\alpha} \|h \circ F - F_0\|.$$

Proof of Proposition 1.1. By the Ascoli-Arzelà Theorem, $\mathcal{C}_\alpha$ is a compact subset of the space of continuous functions on $[0,1]$ endowed with the uniform norm. Hence, from

any sequence of elements in $R_\alpha(F)$, say $\{h_n \circ F\}$, we can extract a uniformly convergent subsequence $h_{n_j} \to h_0 \in \mathcal{C}_\alpha$. But then, obviously, $h_{n_j} \circ F \to h_0 \circ F$ in d_K, which proves (a). Since, on the other hand,

$$\big|\|h_1 \circ F - F_0\| - \|h_2 \circ F - F_0\|\big| \le \|h_1 \circ F - h_2 \circ F\| \le \|h_1 - h_2\|,$$

we see that the map $h \mapsto \|h \circ F - F_0\|$ is continuous and, consequently, it attains its minimum in $R_\alpha(F)$, as claimed in (b). Finally, to check (c) we note that

$$\begin{aligned} \big|d_K(G_1, R_\alpha(F_1)) - d_K(G_1, R_\alpha(F_2))\big| &\le \sup_{h\in\mathcal{C}_\alpha} \big|\|G_1 - h \circ F_1\| - \|G_1 - h \circ F_2\|\big| \quad (1.34)\\ &\le \sup_{h\in C_\alpha} \|h \circ F_1 - h \circ F_2\| \le \tfrac{1}{1-\alpha}\|F_1 - F_2\| \end{aligned}$$

and

$$\big|d_K(G_1, R_\alpha(F_2)) - d_K(G_2, R_\alpha(F_2))\big| \le \sup_{h\in\mathcal{C}_\alpha} \big|\|G_1 - h \circ F_2\| - \|G_2 - h \circ F_2\|\big| \le \|G_1 - G_2\|. \quad (1.35)$$

Now, (1.34) and (1.35) yield (c).

To prove (d1), since $\tilde{F} = h \circ F$, with $h \in \mathcal{C}_\alpha$, it suffices to consider $\tilde{F}_n := h \circ F_n \in R_\alpha(F_n)$ and recall that h is Lipschitz. For (d2), we write $\tilde{F}_n = h_n \circ F_n$ and argue as in (a) to get a d_K-convergent subsequence $h_{n_k} \to h \in \mathcal{C}_\alpha$ from which we easily get $d_K(h_{n_k} \circ F_{n_k}, h \circ F) \to 0$. Finally (d3) is a direct consequence of (c). □

Proof of Lemma 1.3. For the first identity observe that

$$\begin{aligned} \|h \circ F - F_0\| &= \sup_{x\in\mathbb{R}} |h(F(x)) - F_0(x)| = \sup_{F(x)\in[0,1]} |h(F(x)) - F_0(F^{-1}(F(x)))| \\ &= \sup_{t\in[0,1]} |h(t) - F_0(F^{-1}(t))| = \|h - F_0(F^{-1})\|. \end{aligned}$$

On the other hand, if $x_{(i)}, i = 1, \ldots, n$, denote the ordered sample associated to $x_1, \ldots, x_n$ (the same set of values but ordered in nondecreasing sense) and

$$t_0 = 0, \quad t_i = \frac{i}{n}, \quad h_i = h(F_n(x_{(i)})) = h(t_i), \quad \text{and} \quad F_{0,i} = F_0(x_{(i)}), \quad 1 \le i \le n.$$

Taking into account that $h(F_n)$ and $F_0(F_n^{-1})$ are piecewise constant while F_0 and h are non decreasing and continuous, we obtain

$$\|h(F_n) - F_0\| = \max_{1\le i\le n} \max\Big(F_{0,i} - h_{i-1}, h_i - F_{0,i}\Big) = \|h - F_0(F_n^{-1})\|,$$

and the other identity follows from Proposition 1.1, part (b). □

1.7.2 Best L_∞-approximations by Lipschitz-continuous functions with box constraints

The proof of Theorem 1.4 will be developed in this section. We start with a refreshed notation. The role of $1/(1-\alpha)$ will be played now by a generic Lipschitz constant L; our Γ will be substituted by a bounded function $f : \aleph \to \mathbb{R}$, where $(\aleph, d)$ is (at least at the

beginning) a general metric space, while we maintain $[0,1]$ as the range of values. We will also use the notation $x \vee y$ (resp. $x \wedge y$) for the maximum (resp. minimum) of both numbers (or functions). Regarding the Lipschitz norm, recall the trivial inequalities

$$\|f \wedge g\|_{\mathrm{Lip}}, \|f \vee g\|_{\mathrm{Lip}} \leq \|f\|_{\mathrm{Lip}} \vee \|g\|_{\mathrm{Lip}}. \tag{1.36}$$

The first lemma collects some basic properties on the role of the Pasch-Hausdorff envelopes of a function to obtain a Lipschitz-continuous best L_∞-approximation with constrained Lipschitz constant. For the sake of completeness, we will also include a simple proof.

Lemma 1.15. *For a function $f : \aleph \to [0,1]$, given a constant $L \geq 0$, let us consider*

$$f_{L,1}(x) := \inf_{y \in \aleph}(f(y) + Ld(x,y)), \quad f_{L,2}(x) := \sup_{y \in \aleph}(f(y) - Ld(x,y)).$$

(i) This defines functions $f_{L,1}, f_{L,2} : \aleph \to \mathbb{R}$ such that $0 \leq f_{L,1} \leq f_{L,2} \leq 1$.

(ii) $f_{L,1}$ is the pointwise largest function $g : \aleph \to \mathbb{R}$ satisfying $g \leq f$ and $\|g\|_{\mathrm{Lip}} \leq L$. Likewise $f_{L,2}$ is the pointwise smallest function $g : \aleph \to \mathbb{R}$ satisfying $g \geq f$ and $\|g\|_{\mathrm{Lip}} \leq L$.

(iii) The average $f_L := (f_{L,1} + f_{L,2})/2$ satisfies $\|f_L\|_{\mathrm{Lip}} \leq L$ and

$$\|g - f\| \geq \|f_L - f\| = \|f_{L,2} - f_{L,1}\|$$

for any function $g : \aleph \to \mathbb{R}$ such that $\|g\|_{\mathrm{Lip}} \leq L$.

Proof. Part (i) follows directly from the definitions of $f_{L,1}$ and $f_{L,2}$, because, for every $x \in \aleph$:

$$\inf_{y \in \aleph} f(y) \leq f_{L,1}(x) \leq f(x) + Ld(x,x) = f(x) = f(x) - Ld(x,x) \leq f_{L,2}(x) \leq \sup_{y \in \aleph} f(y).$$

To address part (ii) observe that, for arbitrary $x_1, x_2, y \in \aleph$, the triangle inequality for the distance implies $|Ld(x_1,y) - Ld(x_2,y)| \leq Ld(x_1,x_2)$, leading to the inequalities

$$|f_{L,j}(x_2) - f_{L,j}(x_1)| \leq Ld(x_1,x_2) \quad \text{for } j = 1,2,$$

thus to $\|f_{L,j}\|_{\mathrm{Lip}} \leq L, j = 1,2$. Now, if $g : \aleph \to \mathbb{R}$ satisfies $g \leq f$ and $\|g\|_{\mathrm{Lip}} \leq L$, then for $x, y \in \aleph$: $g(x) \leq g(y) + Ld(x,y)$ with equality if $x = y$. Hence

$$g(x) = \inf_{y \in \aleph}(g(y) + Ld(x,y)) \leq \inf_{y \in \aleph}(f(y) + Ld(x,y)) = f_{L,1}(x).$$

Analogously, it follows from $g \geq f$ and $\|g\|_{\mathrm{Lip}} \leq L$ that $g \geq f_{L,2}$, proving (ii).

As to part (iii), let $\epsilon := \|g - f\|$. Then $\|g \pm \epsilon\|_{\mathrm{Lip}} = \|g\|_{\mathrm{Lip}}$ and $g - \epsilon \leq f \leq g + \epsilon$. Consequently, by part (ii),

$$g - \epsilon \leq f_{L,1} \leq f \leq f_{L,2} \leq g + \epsilon$$

This implies that

$$|f_L - f| = (f - f_L) \vee (f_L - f) \leq (f_{L,2} - f_L) \vee (f_L - f_{L,1}) = \frac{f_{L,2} - f_{L,1}}{2} \leq \epsilon,$$

whence

$$\|f_L - f\| \leq \frac{\|f_{L,2} - f_{L,1}\|}{2} \leq \|g - f\|.$$

Since $\|f_L\|_{\mathrm{Lip}} \leq \|f_{L,1}\|_{\mathrm{Lip}}/2 + \|f_{L,2}\|_{\mathrm{Lip}}/2 \leq L$, taking $g = f_L$ gives the announced equality $\|f_L - f\| = \|f_{L,2} - f_{L,1}\|/2$. □

When $\aleph$ is a real interval and f is non-decreasing, the functions $f_{L,1}$ and $f_{L,2}$ in Lemma 1.15 share also that property and can be alternatively expressed in terms of the the Ubhaya's monotone envelopes of the function $f(x) - Lx$. This is the content of the following lemma.

Lemma 1.16. *Let $\aleph$ be a real interval, equipped with the usual distance $d(x, y) = |x - y|$. If $f : \aleph \to [0, 1]$ is non-decreasing, then the functions $f_{L,1}, f_{L,2}$ in Lemma 1.15 are non-decreasing too, and for arbitrary $x \in \aleph$ and $j = 1, 2$,*

$$f_{L,j}(x) = \gamma_{L,j}(x) + Lx,$$

where $\gamma_{L,j}, j = 1, 2$ are the non-increasing functions

$$\gamma_{L,1}(x) := \inf_{y \in \aleph : y \leq x} (f(y) - Ly) \quad \textit{and} \quad \gamma_{L,2}(x) := \sup_{y \in \aleph : y \geq x} (f(y) - Ly).$$

In particular,

$$\|f_{L,2} - f_{L,1}\| = \|\gamma_{L,2} - \gamma_{L,1}\| = \sup_{y,x \in \aleph : y \leq x} (f(x) - f(y) - L(x - y)). \tag{1.37}$$

Proof. The representations of $f_{L,1}$ and $f_{L,2}$ in terms of $\gamma_{L,1}$ and $\gamma_{L,2}$ follow from the fact that for arbitrary $x, y \in \aleph$,

$$f(y) + Ld(x, y) \begin{cases} = f(y) + L(x - y) = f(y) - Ly + Lx & \text{if } y \leq x \\ \geq f(x) = f(x) - Lx + Lx & \text{if } y \geq x, \end{cases}$$

$$f(y) - Ld(x, y) \begin{cases} = f(y) - L(y - x) = f(y) - Ly + Lx & \text{if } y \geq x \\ \leq f(x) = f(x) - Lx + Lx & \text{if } y \leq x, \end{cases}$$

where the inequalities follow from f being non-decreasing. Note that both functions $\gamma_{L,1}$ and $\gamma_{L,2}$ are non-increasing, but adding the term Lx to them leads to non-decreasing functions: For $x_1, x_2 \in \aleph$ with $x_1 < x_2$, isotonicity of f implies that

$$\begin{aligned} f_{L,2}(x_1) &= \sup_{y \geq x_2} (f(y) - Ly + Lx_1) \vee \sup_{x_1 \leq y \leq x_2} (f(y) - Ly + Lx_1) \\ &\leq (f_{L,2}(x_2) - Lx_2 + Lx_1) \vee f(x_2) \\ &\leq f_{L,2}(x_2), \end{aligned}$$

and

$$\begin{aligned} f_{L,1}(x_2) &= \inf_{y\le x_1}(f(y)-Ly+Lx_2)\wedge\sup_{x_1\le y\le x_2}(f(y)-Ly+Lx_2)\\ &\ge (f_{L,1}(x_2)+Lx_2-Lx_1)\wedge f(x_1)\\ &\ge f_{L,1}(x_1), \end{aligned}$$

because $f_{L,1}\le f\le f_{L,2}$. $\square$

Finally, let us include in the problem the boundary restrictions. Notice that Theorem 1.17 is just a rephrasing of Theorem 1.4. A look at that Theorem shows that $h_\alpha=\tilde{h}_\alpha+\frac{\cdot}{1-\alpha}$ is an element of $\mathcal{C}_\alpha$ such that $\|h_\alpha-\Gamma\|=\min_{h\in\mathcal{C}_\alpha}\|h-\Gamma\|$, that is, h_α is an optimal trimming function in the sense described above. We recall that we do not claim uniqueness of this minimizer, but this particular choice allows to compute $d_K(F_0,R_\alpha(F_n))$ for sample d.f.'s.

Theorem 1.17. *Let $f:[0,1]\to[0,1]$ be non-decreasing. For $L\ge 1$ consider the function*

$$\begin{aligned} \tilde{f}_L(x) &:= (f_L(x)\vee(1-L+Lx))\wedge Lx\\ &= ((\gamma_L(x)\vee(1-L))\wedge 0)+Lx, \end{aligned}$$

where $\gamma_L:=(\gamma_{L,1}+\gamma_{L,2})/2$, and $f_L,\gamma_{L,1},\gamma_{L,2}$ are defined as in Lemmas 1.15 and 1.16. Then $\tilde{f}_L:[0,1]\to\mathbb{R}$ is non-decreasing and verifies $\tilde{f}_L(0)=0$ and $\tilde{f}_L(1)=1$ and $\|\tilde{f}_L\|_{\mathrm{Lip}}\le L$, and for arbitrary functions $g:[0,1]\to\mathbb{R}$ with $g(0)=0$ and $g(1)=1$ and $\|g\|_{\mathrm{Lip}}\le L$,

$$\begin{aligned} \|g-f\| &\ge \|\tilde{f}_L-f\|\\ &= \max\Big\{f_{L,2}(0),1-f_{L,1}(1),\sup_{0\le y\le x\le 1}(f(x)-f(y)-L(x-y))/2\Big\} \end{aligned} \tag{1.38}$$

Proof of Theorem 1.4 and 1.17. Let us begin noting that both expressions for $\tilde{f}_L$ are trivially equivalent from the relations between $\gamma_{L,j}$ and $f_{L,j}$.

That $\tilde{f}_L$ verifies the required properties easily follows from the preceding lemmas (recall also inequalities (1.36)). Let then $g:[0,1]\to\mathbb{R}$ with $\|g\|_{\mathrm{Lip}}\le L$. Also by the precedent lemmas,

$$\|g-f\|\ge\|f_L-f\|=\sup_{0\le y\le x\le 1}(f(x)-f(y)-L(x-y))/2.$$

Under the additional constraint that $g(0)=0$, for arbitrary $x\in[0,1]$,

$$f(x)-g(x)=f(x)-(g(x)-g(0))\ge f(x)-Lx,$$

whence

$$\|g-f\|\ge\sup_{0\le x\le 1}(f(x)-Lx)=f_{L,2}(0).$$

Analogously, the additional constraint $g(1)=1$ implies that

$$f(x)-g(x)=f(x)+(g(1)-g(x))-1\le f(x)+L(1-x)-1,$$

whence

$$-\|g-f\|\le\inf_{0\le x\le 1}(f(x)+L(1-x))-1=f_{L,1}(1)-1.$$

These considerations show that for any function $g : [0,1] \to \mathbb{R}$ verifying the conditions $g(0) = 0$, $g(1) = 1$ and $\|g\|_{\text{Lip}} \le L$,

$$\|g - f\| \ge \|f_L - f\| \vee f_{L,2}(0) \vee (1 - f_{L,1}(1)).$$

The function $\tilde{f}_L$ satisfies the previous constraints on g, too, so

$$\|\tilde{f}_L - f\| \ge \|f_L - f\| \vee f_{L,2}(0) \vee (1 - f_{L,1}(1)).$$

It remains to prove the reverse inequality. For $x \in [0,1]$, we have to distinguish three cases: If $1 - L + Lx \le f_L(x) \le Lx$, then $\tilde{f}_L(x) = f_L(x)$, so $|\tilde{f}_L(x) - f(x)| \le \|f_L - f\|$. If $f_L(x) > Lx$, then $\tilde{f}_L(x) = Lx$, and

$$f(x) - \tilde{f}_L(x) \begin{cases} = f(x) - Lx \le f_{L,2}(0), \\ > f(x) - f_L(x) \ge -\|f_L - f\|. \end{cases}$$

Similarly, if $f_L(x) < 1 - L + Lx$, then $\tilde{f}_L(x) = Lx$, and

$$f(x) - \tilde{f}_L(x) \begin{cases} = f(x) + L(1-x) - 1 \ge f_{L,1}(1) - 1, \\ < f(x) - f_L(x) \le \|f_L - f\|. \end{cases}$$

□

In the case, considered in Theorem 1.17, of a non-decreasing function f, since the functions $f_{L,j}$ are absolutely continuous and the relations $\gamma_{L,j} = f_{L,j} - Lx$ hold, all the functions $f_L, \gamma_L, \gamma_{L,j}$ are absolutely continuous so $\{\gamma_L \le 1-L\}, \{\gamma_L \ge 0\}, \{\gamma_L \in [1-L, 0]\}$ are compact sets and continuous functions attain their maximum values on these sets. This allows to get alternative expressions for (1.38) as given in the following theorem. We note that here and throughout we use the convention that the max over an empty set equals $-\infty$.

Theorem 1.18. *Let $f : [0,1] \to [0,1]$ be non-decreasing and continuous and assume the notation in Theorem 1.17. Then the following alternative expressions for (1.38) hold:*

$$\|f - \tilde{f}_L\| = \max\left(\max_{x \in \mathcal{T}_1} (f(x) - Lx), \max_{x \in \mathcal{T}_2} (1 - L + Lx - f(x)), \frac{1}{2} \max_{1-L \le \gamma_L(x) \le 0} (\gamma_{L,2}(x) - \gamma_{L,1}(x))\right) \tag{1.39}$$

$$= \max\left(\max_{x \in \mathcal{T}_1} (f(x) - Lx), \max_{x \in \mathcal{T}_2} (1 - L + Lx - f(x)), \frac{1}{2} \max_{(y,x) \in \mathcal{T}_3} (f(x) - f(y) - L(x-y))\right). \tag{1.40}$$

Here, we used the notation $\mathcal{T}_1 = \{x \in [0,1] : \gamma_L(x) \ge 0\}$, $\mathcal{T}_2 = \{x \in [0,1] : \gamma_L(x) \le 1-L\}$, $\mathcal{T}_3 = \{(y,x) : 0 \le y \le x \le 1, 1 - L \le \frac{1}{2}(f(y) + f(x) - L(y+x)) \le 0\}$.

Once we know Theorem 1.17, a proof of this result would take advantage of the fact that the right-hand side in (1.39) is upper bounded by the same expression with the unrestricted maxima, which, by (1.37) is just the right-hand side in (1.38) when f is continuous. However, with some additional effort we can obtain a more general result that does not requires the monotonicity assumption on the objective function and opens a way to address the directional differentiability of the functional $f \to \|f - \tilde{f}_L\|$. Both goals will be carried through the following section.

1.7.3 Best L_∞-approximations by monotone functions with box constraints

The following theorem gives appropriate characterizations of the best approximation of a bounded function (in uniform norm) by monotone functions with a box constraint. Without this constraint, best approximation by monotone functions in the L_∞-norm has been considered in [Ubhaya, 1974a, Ubhaya, 1974b], with results that cover the case $A = -\infty$, $B = \infty$ in Theorem 1.19 below. Notice that this theorem, based on Ubhaya's envelopes, would also provide an (arguably more involved) alternative proof for Theorem 1.17. Notice that the function G plays the role of the transformed function, $f(x) - Lx$ (the difference of two non-decreasing functions) in the previous section, while the scope here is general.

Theorem 1.19. *Assume $G : [0,1] \to \mathbb{R}$ is a bounded function and $-\infty \le A \le B \le \infty$. Define $U(x) = \sup_{x \le y \le 1} G(y)$, $L(x) = \inf_{0 \le y \le x} G(y)$, $\bar{G}(x) = (L(x) + U(x))/2$ and*

$$\bar{G}_{A,B}(x) = \max(\min(\bar{G}(x), B), A).$$

Then $U, L, \bar{G}$ and $\bar{G}_{A,B}$ are non-increasing, $L(x) \le G(x) \le U(x)$ and for every non-increasing $h : [0,1] \to [A,B]$ we have

$$\|G - \bar{G}_{A,B}\| \le \|G - h\|. \tag{1.41}$$

Furthermore, if G is continuous then $U, L, \bar{G}$ and $\bar{G}_{A,B}$ are also continuous and

$$\begin{aligned} \|G - \bar{G}_{A,B}\| &= \max\left(\max_{\bar{G}(x) \ge B} (G(x) - B), \max_{\bar{G}(x) \le A} (A - G(x)), \tfrac{1}{2} \max_{A \le \bar{G}(x) \le B} (U(x) - L(x)) \right) \\ &= \max\left(\max_{x \in \mathcal{T}_1} (G(x) - B), \max_{x \in \mathcal{T}_2} (A - G(x)), \tfrac{1}{2} \max_{(y,x) \in \mathcal{T}_3} (G(x) - G(y)) \right), \end{aligned} \tag{1.42}$$

where $\mathcal{T}_1 = \{x \in [0,1] : \bar{G}(x) \ge B\}$, $\mathcal{T}_2 = \{x \in [0,1] : \bar{G}(x) \le A\}$ and $\mathcal{T}_3 = \{(y,x) : 0 \le y \le x \le 1, A \le \frac{1}{2}(G(y) + G(x)) \le B\}$.

Proof. The bounds $L(x) \le G(x) \le U(x)$ are obvious, and also the fact that U and L are non-increasing (hence, also $\bar{G}$ and $\bar{G}_{A,B}$).

- Next, consider some non-increasing $h : [0,1] \to [A,B]$ and $x \in [0,1]$. Since $L(x) \le G(x) \le U(x)$, we have that $G(x) = \bar{G}(x)$ whenever $U(x) = L(x)$. Hence, if $U(x) = L(x) \in [A,B]$ we have $\bar{G}_{A,B}(x) = G(x)$ and, consequently,

$$0 = |\bar{G}_{A,B}(x) - G(x)| \le \|h - G\|.$$

- Obviously, $\bar{G}_{A,B}(x) = B$ if $U(x) = L(x) > B$ and we still have that

$$|\bar{G}_{A,B}(x) - G(x)| \le |h(x) - G(x)| \le \|h - G\|$$

and similarly for the case $U(x) = L(x) < A$.

- It remains to deal with the case $U(x) > L(x)$. For every $\varepsilon > 0$ there exist $x_a \in [0,x]$, $x_b \in [x,1]$ such that $G(x_a) < L(x) + \varepsilon$ and $G(x_b) > U(x) - \varepsilon$. If $\bar{G}(x) > B$ then $\bar{G}_{A,B}(x) = B$. Using again that $L(x) \le G(x) \le U(x)$ we see that $|\bar{G}_{A,B}(x) - G(x)| \le$

$U(x) - B < G(x_b) - B + \varepsilon \leq |G(x_b) - h(x_b)| + \varepsilon$ for small enough ε, showing that $|\bar{G}_{A,B}(x) - G(x)| \leq \|h - G\|$.

Similarly, if $\bar{G}(x) < A$ we conclude that $|\bar{G}_{A,B}(x) - G(x)| \leq \|h - G\|$.

Finally, assume that $U(x) > L(x)$ and $\bar{G}(x) \in [A, B]$. Since h is non-increasing we have that $h(x_a) \geq h(x_b)$ and, consequently,

$$\|h-G\| \geq \max(|h(x_a)-G(x_a)|, |h(x_b)-G(x_b)|) \geq \frac{G(x_b) - G(x_a)}{2} \geq |\bar{G}_{A,B}(x)-G(x)|-2\varepsilon$$

for ε small enough. This completes the proof of (1.41).

To check continuity of U note that for $0 \leq y < x \leq 1$ $U(y) = \max(U(x), \max_{y \leq z \leq x} G(z))$. Now, given $\varepsilon > 0$ we can fix $\delta > 0$ such that $|G(x) - G(y)| \leq \varepsilon$ whenever $|y - x| \leq \delta$. But then $|U(y) - U(x)| \leq \varepsilon$ if $|y - x| \leq \delta$, proving continuity of U. L can be handled similarly. As a consequence we see that $\bar{G}$ and $\bar{G}_{A,B}$ are also continuous.

Now, to prove the first equality in the statement we take $x \in [0, 1]$ and consider first the case $x \in \mathcal{T}_1$. Note that, necessarily, $U(x) \geq B$, $U(x)-B \geq B-L(x)$ and $\bar{G}_{A,B}(x) = B$.

- If $G(x) \geq B$ then $|G(t) - \bar{G}_{A,B}(x)| = G(x) - B$.
- Assume, on the contrary, that $G(x) < B$. Set $x_+ = \inf\{y \leq x : G(y) = U(x)\}$. By continuity, $G(x_+) = U(x) = U(x_+)$. Recall that if $x_+ = \emptyset$, it means that there is an $y_+ > x$ such that $G(y_+) = U(x)$ and that $\sup_{0 \leq y \leq x} \|G(y) - \bar{G}_{A,B}(y)\| \leq G(y_+) - B \leq |G(y_+) - \bar{G}_{A,B}(y_+)|$. Hence, $y_+ \in \mathcal{T}_i$ for some $i = 1, 2, 3$.

Now, if $\bar{G}(x_+) \geq B$ then $G(x_+) - B = U(x) - B \geq B - L(x) \geq B - G(x) = |G(x) - \bar{G}_{A,B}(x)|$. If, on the contrary, $\bar{G}(x_+) < B$, then there exists $x' \in [x, x_+]$ such that $\bar{G}(x') \in (A, B)$. But we must have $U(x') = U(x) = U(x_+)$ and $L(x') < L(x)$ and, consequently, we have that

$$|G(x) - \bar{G}_{A,B}(x)| = B - G(x) \leq B - L(x) \leq \frac{U(x) - L(x)}{2} < \frac{U(x') - L(x')}{2}.$$

Summarizing, we see that

$$\max_{\bar{G}(x) \geq B} |G(x) - \bar{G}_{A,B}(x)| \leq \max\left(\max_{\bar{G}(x) \geq B} (G(x) - B), \tfrac{1}{2} \max_{A \leq \bar{G}(x) \leq B} (U(x) - L(x))\right). \quad (1.43)$$

Similarly,

$$\max_{\bar{G}(x) \leq A} |G(x) - \bar{G}_{A,B}(x)| \leq \max\left(\max_{\bar{G}(x) \leq A} (A - G(x)), \tfrac{1}{2} \max_{A \leq \bar{G}(x) \leq B} (U(x) - L(x))\right) \quad (1.44)$$

and, obviously, if $\bar{G}(x) \in [A, B]$ then $\bar{G}_{A,B}(x) = \bar{G}(x)$ and $|G(x) - \bar{G}_{A,B}(x)| \leq \frac{1}{2}(U(x) - L(x))$, which implies that

$$\max_{A \leq \bar{G}(x) \leq B} |G(x) - \bar{G}_{A,B}(t)| \leq \tfrac{1}{2} \max_{A \leq \bar{G}(x) \leq B} (U(x) - L(x)). \quad (1.45)$$

Now combining (1.43), (1.44) and (1.45) we see that

$$\|G - \bar{G}_{A,B}\| \leq \max\left(\max_{\bar{G}(x) \geq B} (G(x) - B), \max_{\bar{G}(x) \leq A} (A - G(x)), \tfrac{1}{2} \max_{A \leq \bar{G}(x) \leq B} (U(x) - L(x))\right).$$

Assume now that x_0 is such that $\bar{G}(x_0) \geq B$. Then $\bar{G}_{A,B}(x_0) = B$ and $G(x_0) - B \leq |G(x_0) - \bar{G}_{A,B}(x_0)|$. This implies $\max_{\bar{G}(x)\geq B}(G(x) - B) \leq \|G - \bar{G}_{A,B}\|$.

Similarly, $\max_{\bar{G}(t)\leq A}(A - G(x)) \leq \|G - \bar{G}_{A,B}\|$.

Finally, suppose x_0 is such that $\bar{G}(x_0) \in [A, B]$ and

$$U(x_0) - L(x_0) = \max_{\bar{G}(x)\in[1,B]}(U(x) - L(x)) \geq \max\left(\max_{\bar{G}(x)\geq B}(G(x) - B), \max_{\bar{G}(x)\leq A}(A - G(x))\right).$$

- If $U(x_0) = L(x_0)$ then

$$\|G-\bar{G}_{A,B}\| = \max\left(\max_{\bar{G}(x)\geq B}(G(x) - B), \max_{\bar{G}(x)\leq A}(A - G(x)), \tfrac{1}{2}\max_{A\leq\bar{G}(x)\leq B}(U(x) - L(x))\right) = 0.$$

- If $U(x_0) > L(x_0)$ then we set $x_+ = \inf\{y \in [x_0, 1] : G(y) = U(x_0)\}$. Then $U(y) = U(x_0)$ for $y \in [x_0, x_+]$ and

$$G(x_+) = U(x_+) = U(x_0).$$

Set $x_+ = \sup\{y \in [0, x_0] : G(y) = L(x_0)\}$. We have $L(y) = L(x_0) = G(x_-)$ for $y \in [x_-, x_0]$. We claim that

$$L(y) = L(x_0) \quad \text{for } y \in [x_0, x_+]. \tag{1.46}$$

To check (1.46) note that, if $\bar{G}(x_0) > A$ and (1.46) fails then we could find $y \in [x_0, x_+]$ with $L(y) < L(x_0)$, $\bar{G}(y) \in (A, B]$ and $U(y) - L(y) > U(x_0) - L(x_0)$, while if $\bar{G}(x_0) = A$ and (1.46) fails then $G(y) < L(x_0)$ for some $y \in (x_0, x_+)$, $\bar{G}(y) < A$ and $A - L(y) > A - L(x_0) = \frac{1}{2}(U(x_0 - L(x_0)$, against the assumption on x_0.

Hence, from (1.46) we conclude that $\bar{G}(x_+) = \bar{G}(x_0) \in [A, B]$ and $|G(x_+)-\bar{G}_{A,B}(x_+)| = \frac{1}{2}(U(x_0) - L(x_0))$, showing that $\frac{1}{2}(U(x_0) - L(x_0)) \leq \|G - \bar{G}_{A,B}\|$. Combining the last estimates we see that the first equality in (1.42) holds.

For the second identity we note that arguing as above we see that $U(x_0) - L(x_0) = G(x) - G(y)$ for some $(y, x) \in \mathcal{T}_3$ if $\bar{G}(x_0) \in [A, B]$. Assume, on the other hand, that $(y_0, x_0) \in \mathcal{T}_3$ satisfies

$$\frac{1}{2}(G(x_0) - G(y_0)) \geq \max\left(\max_{\bar{G}(x)\geq B}(G(x) - B), \max_{\bar{G}(x)\leq A}(A - G(x))\right).$$

- We consider first the case $\frac{1}{2}(G(y_0) + G(x_0)) \in (A, B)$.

We claim that $U(x_0) = G(x_0)$ since, otherwise, there exists $x' > x_0$ such that $\frac{1}{2}(G(y_0)+ G(x')) \in (A, B)$ and $G(x') > G(x_0)$ and this would imply $G(x') - G(y_0) > G(x_0) - G(y_0)$, against the assumption.

Similarly, we see that $G(y_0) = L(x_0)$.

Furthermore, $L(x) = L(y_0)$ for $x \in [y_0, x_0]$. If $G(x_0) < U(x_0)$ then there exists $x' > x_0$ such that $\frac{1}{2}(G(y_0) + G(x')) \in (A, B)$ and $G(x') > G(x_0)$, but then $G(x') - G(y_0) > G(x_0) - G(y_0)$, contradicting maximality of (y_0, x_0). Similarly we see that $G(y_0) = L(y_0)$ and also that $L(x) = L(y_0)$ for $x \in [y_0, x_0]$. Hence, $G(x_0) - G(y_0) = U(x_0) - L(x_0)$ and $\bar{G}(x_0) \in (A, B)$.

• In the case $\frac{1}{2}(G(y_0)+G(x_0)) = B$ we have that necessarily $G(x_0) \geq B$ and, arguing as above, we see that $G(y_0) = L(y)$ for all $y \in [y_0, x_0]$. This implies that $\bar{G}(x_0) \geq B$ and $\frac{1}{2}(G(x_0) - G(y_0)) = G(x_0) - B$.
• Arguing similarly for the case $\frac{1}{2}(G(y_0)+G(x_0)) = A$ we conclude that the second equality in (1.42) holds. □

Remark 1.20. *The sets of optimizers within $\mathcal{T}_1, \mathcal{T}_2$ and $\mathcal{T}_3$ in Lemma 1.19 play an important role in the next results. For convenience, we denote $T_1 = \{x_0 \in \mathcal{T}_1 : G(x_0) - B = \|G - \bar{G}_{A,B}\|\}$, $T_2 = \{x_0 \in \mathcal{T}_2 : A - G(x_0) = \|G - \bar{G}_{A,B}\|\}$ and $T_3 = \{(y_0, x_0) \in \mathcal{T}_3 : \frac{1}{2}(G(x_0) - G(y_0)) = \|G - \bar{G}_{A,B}\|\}$. A look at the proof of Lemma 1.19 shows that if $x_0 \in T_1$ then G has a local maximum at x_0 and a local minimum if $x_0 \in T_2$. Also, if $(y_0, x_0) \in T_3$ then G has a local maximum at x_0 and a local minimum at y_0.*

Our next result addresses the directional differentiability of the functional $G \to \|G - \bar{G}_{A,B}\|$ that appeared in the last theorem. This kind of result typically allows to obtain efficiency and asymptotic distributional behaviour of functionals in the statistical setting (see e.g. [Cárcamo et al., 2019]).

Theorem 1.21. *Assume $G, J : [0,1] \to \mathbb{R}$ are continuous bounded functions and $r_n > 0$ is a sequence of real numbers such that $r_n \to \infty$. Define $G_n = G + \frac{J}{r_n}$ and consider $\bar{G}, \bar{G}_{A,B}$ as in Theorem 1.19 and $\bar{G}_{A,B,n}$ built in the same way as $G_{A,B}$ but from G_n. Assume further that T_1, T_2 and T_3 are as in Remark 1.20 and that there is no $x \in T_1$ with $\bar{G}(x) = B$, no $x \in T_2$ with $\bar{G}(x) = A$ and no $(y, x) \in T_3$ with $\frac{1}{2}(G(x) + G(y)) \in \{A, B\}$. Then*

$$r_n(\|G_n - \bar{G}_{A,B,n}\| - \|G - \bar{G}_{A,B}\|) \to \max\left(\max_{x \in T_1} J(x), \max_{t \in T_2}(-J(x)), \frac{1}{2}\max_{(y,x)\in T_3}(J(x) - J(y))\right).$$

Proof. We use the notation U, L from Theorem 1.19 and write $U_n, L_n, \bar{G}_n, T_{n,i}$ for the corresponding objects coming from G_n. Observe that $\|U_n - U\| \leq \|J\|/r_n \to 0$ and, similarly, $\|\bar{G}_n - \bar{G}\| \to 0$. Assume that $x \in T_1$. By assumption and the last convergence we have that $\bar{G}_n(x) > B$ for large enough n and, therefore, $\|G_n - \bar{G}_{A,B,n}\| \geq (G_n(t) - B)$. But this implies

$$r_n(\|G_n - \bar{G}_{A,B,n}\| - \|G - \bar{G}_{A,B}\|) \geq r_n((G_n(x) - B) - (G(x) - B)) = J(x).$$

Arguing similarly for T_2 and T_3 we conclude that

$$\begin{aligned} \liminf r_n(\|G_n - \bar{G}_{A,B,n}\| - \|G - \bar{G}_{A,B}\|) \\ \geq \max\left(\max_{x \in T_1} J(x), \max_{x \in T_2}(-J(x)), \frac{1}{2}\max_{(y,x)\in T_3}(J(x) - J(y))\right). \end{aligned} \tag{1.47}$$

For the upper bound assume $x_n \in T_{n,1}$ (that is, $x_n \in \mathcal{T}_{n,1}$ such that $G_n(x_n) - B = \|G_n - \bar{G}_{A,B,n}\|$). By compactness, taking subsequences if necessary, we can assume that $x_n \to x_0$ for some $x_0 \in [0,1]$ with $\bar{G}(x_0) \geq B$ and $G(x_0) - B = \|G - \bar{G}_{A,B}\|$. But this means that $x_0 \in T_1$. Hence, by assumption $G(x_0) > B$ and, consequently, $G(x_n) > B$ for large enough n. In this case $\|G - \bar{G}_{A,B}\| \geq (G(x_n) - B)$, which implies that

$$r_n(\|G_n - \bar{G}_{A,B,n}\| - \|G - \bar{G}_{A,B}\|) \leq r_n((G_n(x_n) - B) - (G(x_n) - B)) = J(x_n) \to J(x_0).$$

With the same argument applied to T_2 and T_3 we conclude that

$$\limsup r_n(\|G_n - \bar{G}_{A,B,n}\| - \|G - \bar{G}_{A,B}\|) \leq \max\left(\max_{x\in T_1} J(x), \max_{x\in T_2}(-J(x)), \frac{1}{2}\max_{(y,x)\in T_3}(J(x) - J(y))\right) \tag{1.48}$$

and complete the proof. □

Specializing the last results for $G(x) = f(x) - Lx$, where f is non-decreasing, $L \geq 1$ a constant, and $A = 1 - L, B = 0$, we can obtain a first result on the directional differentiability of the functional $f \to \|f - \tilde{f}_L\|$ considered in Section 1.7.2. Note that now, recovering the notation in that section, the relevant sets are $\mathcal{T}_1, \mathcal{T}_2$ and $\mathcal{T}_3$ as defined in Theorem 1.18, and $T_1 = \{x_0 \in \mathcal{T}_1 : f(x_0) - Lx_0 = \|f - \tilde{f}_L\|\}$, $T_2 = \{x_0 \in \mathcal{T}_2 : 1 - L + Lx_0 - f(x_0) = \|f - \tilde{f}_L\|\}$ and $T_3 = \{(y_0, x_0) \in \mathcal{T}_3 : \frac{1}{2}(f(x_0) - f(y_0) - L(x_0 - y_0)) = \|f - \tilde{f}_L\|\}$. Theorem 1.21 translates then to the following immediate corollary.

Corollary 1.22 (Directional differentiability.)**.** *Let $f, f_n : [0,1] \to \mathbb{R}$ be non-decreasing bounded functions, $r_n > 0$ a sequence of real numbers such that $r_n \to \infty$ and $r_n(f_n - f) \to J$ point-wise, where $J : [0,1] \to \mathbb{R}$ is a continuous bounded function. Assume further that f is continuous, that T_1, T_2 and T_3 are as above and that there is no $x \in T_1$ with $\gamma_L(x) = 0$, no $x \in T_2$ with $\gamma_L(x) = 1 - L$ and no $(y, x) \in T_3$ with $\frac{1}{2}(f(x) + f(y) - L(x + y)) \in \{1 - L, 0\}$. Let $\tilde{f}_{n,L}, \tilde{f}_L$ respectively denote the best L_∞-approximations to f_n and f by Lipschitz-continuous functions $h : [0,1] \to \mathbb{R}$ with $\|h\|_{\text{Lip}} \leq L$ and verifying $h(0) = 0, h(1) = 1$, as in Theorem 1.17. Then*

$$r_n(\|f_n - \tilde{f}_{L,n}\| - \|f - \tilde{f}_L\|) \to \max\left(\max_{x\in T_1} J(x), \max_{x\in T_2}(-J(x)), \frac{1}{2}\max_{(y,x)\in T_3}(J(x) - J(y))\right).$$

Proof of Theorem 1.9. As in Theorem 1.4 for Γ, we write $G(t) = H^{-1}(t) - \frac{t}{1-\alpha}$, $G_n(t) = H_n^{-1}(t) - \frac{t}{1-\alpha}$, keep the notation for $\tilde{h}_\alpha$ and write $\tilde{h}_{\alpha,n}$ for the corresponding object defined from G_n. With this notation we will show weak convergence of

$$A_n = \sqrt{n}\left(\|\tilde{h}_{\alpha,n} - G_n\| - \|\tilde{h}_\alpha - G\|\right)$$

to complete the proof. With this goal we consider the quantile process

$$Q_n(t) = \sqrt{n}(H_n^{-1}(t) - H^{-1}(t)), \quad 0 \leq t \leq 1.$$

Assumption (1.20) allows us to apply Theorem 18.1.1, p. 640 and Example 18.1.2, p.641, in [Shorack and Wellner, 1986] to conclude that we can choose a version of Q_n and a Brownian brigde, B, such that if $w = (H^{-1})'$ and $\tilde{B} = wB$ then

$$\|Q_n - \tilde{B}\| \to 0 \tag{1.49}$$

in probability. If (1.21), instead of (1.20), holds then we can still find versions of Q_n and $\tilde{B}$ such that $\max_{\varepsilon \leq t \leq 1-\varepsilon} |Q_n(t) - \tilde{B}(t)| \to 0$ in probability. It is easy to see that (1.21) implies that

$$\|\tilde{h}_\alpha - G\| = \max_{\varepsilon \leq t \leq 1-\varepsilon} |\tilde{h}_\alpha(t) - G(t)|$$

and also that, in a probability one set, eventually

$$\|\tilde{h}_{\alpha,n} - G_n\| = \max_{\varepsilon \le t \le 1-\varepsilon} |\tilde{h}_{\alpha,n}(t) - G_n(t)|.$$

From this point we assume that (1.20) (hence, also (1.49)) holds. Our last comments, however, show that our proof can be trivially adapted to cover the case when (1.21) holds. We omit further details.

Next, we note that w is a continuous function and, as a consequence, $\tilde{B}$ has, with probability one, continuous trajectories. We note that $G_n(t) = G(t) + \frac{Q_n(t)}{\sqrt{n}}$ and introduce $\bar{G}_n(t) = G(t) + \frac{\tilde{B}(t)}{\sqrt{n}}$ and the related functions $\bar{U}_n$, $\bar{L}_n$ and $\bar{h}_{\alpha,n}$ related to $\bar{G}_n$ as U_n, L_n and $\tilde{h}_{\alpha,n}$ are related to G_n. We consider

$$C_n = \sqrt{n}\left(\|\bar{h}_{\alpha,n} - \bar{G}_n\| - \|\tilde{h}_\alpha - G\|\right)$$

and observe that $|A_n - C_n| \le \sqrt{n}\|\tilde{h}_{\alpha,n} - \bar{h}_{\alpha,n}\| + \sqrt{n}\|G_n - \bar{G}_n\| = o_P(1)$ by (1.49) (we are using that $\sqrt{n}\|\bar{U}_n - U_n\| \le \|Q_n - \tilde{B}\|$, with a similar bound for the lower envelopes). Consequently, if suffices to prove convergence of C_n. Noticing that $\tilde{h}_\alpha = \bar{G}_{\frac{-\alpha}{1-\alpha},0}$ and $\tilde{h}_{\alpha,n} = \bar{G}_{\frac{-\alpha}{1-\alpha},0,n}$, from Theorem 1.21, we conclude that

$$C_n \underset{w}{\to} \max\left(\max_{t\in T_1} \tilde{B}(t), \max_{t\in T_2}(-\tilde{B}(t)), \max_{(s,t)\in T_3} \tfrac{1}{2}(\tilde{B}(t) - \tilde{B}(s))\right).$$

The conclusion follows upon noting that (see Remark 1.20) in the sets T_i, the function G has local maxima: if $t_0 \in T_1$ then G has a local maximum at t_0 and a local minimum if $t_0 \in T_2$, also, if $(s_0, t_0) \in T_3$ then G has a local maximum at t_0 and a local minimum at s_0. Therefore, $G'(t_0) = 0$ and $G'(s_0) = 0$ for every $t_0 \in T_1, T_2$ or $(s_0, t_0) \in T_3$ and this entails $w(t_0) = w(s_0) = \frac{1}{1-\alpha}$ for these points. □

2

Optimal-transport approach to flow cytometry

In this chapter we show how to apply the optimal-transport techniques mentioned in the Introduction to the field of Flow Cytometry. The theoretical motivation comes from the good properties of the Wasserstein barycenters as consensus representatives. The practical motivation is due to a collaboration with the clinical research team, lead by Dr. Alberto Orfao, of the Cancer Research Center in Salamanca. We are extremely grateful for their insights and help, as well as for all the data they have kindly allowed us to use in this work.

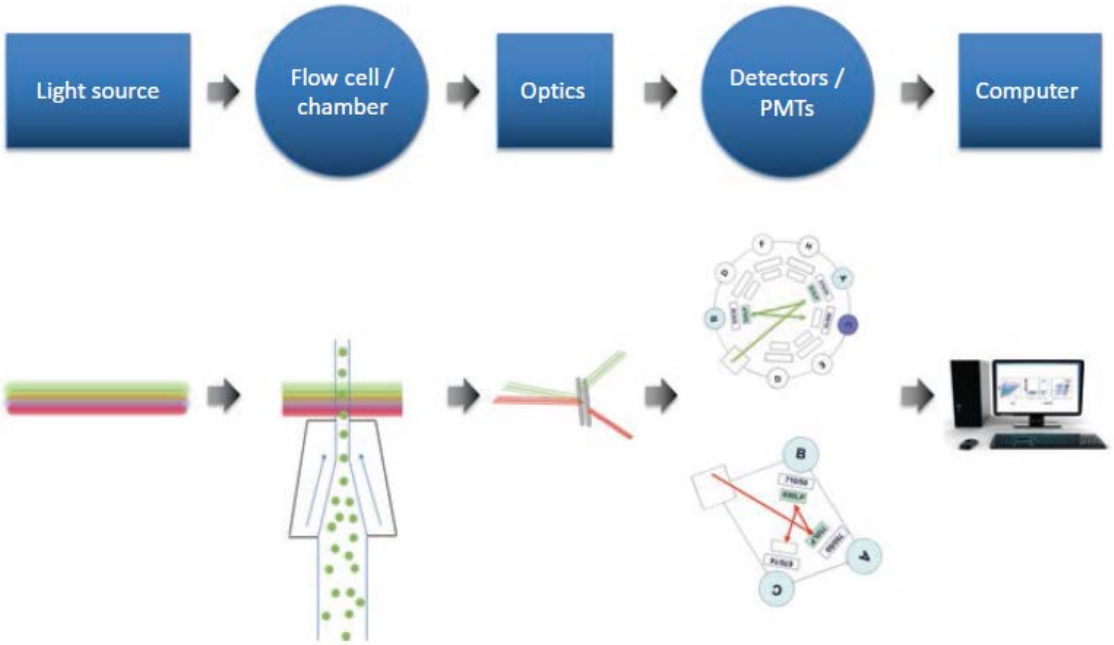
2.1 Introduction

We begin with a concise description of what Flow Cytometry (FC) is as stated in [Goetz et al., 2018]. "FC is the measure or quantification of cells suspended in a fluid phase. The cells are labeled with fluorochrome-coupled antibodies specific for a cellular marker of interest. The fluidics chamber ensures that cells pass single file through a laser beam, which excites the fluorochrome. The emitted light is picked up by a detector (photomultiplier [PMT]), and the signal is translated electronically to a computer for analysis". A schematic depiction of this process can be seen in Figure 2.1.

The same authors give the following commentary on the practical interest of FC. "Because of its single-cell nature, flow cytometry allows for the analysis of protein expression on a per cell basis, making it quantifiable with results expressed as a count or number of events. The technique allows researchers to evaluate and quantitate specific cell types within a heterogeneous population enabling the analysis of small numbers of cells or rare subsets in a mixed population. It also allows for multiprotein analysis within a single cell when using multiple antibodies conjugated to different fluorescent dyes. *This makes flow cytometry a prevalent technique that today is used in nearly every aspect of cell biology research and clinical patient cell analysis*" (our emphasis).

A main component for analysis in FC is gating, the assignment of individual cells (data records) into discrete cell types. Manual gating, i.e., an expert assigning cell types (labels) to individual cells, using a set of rules on one or two-dimensional projections, has been

Figure 2.1: Progression from antibody stained cells to flow cytometry data plots. Taken from [Goetz et al., 2018].

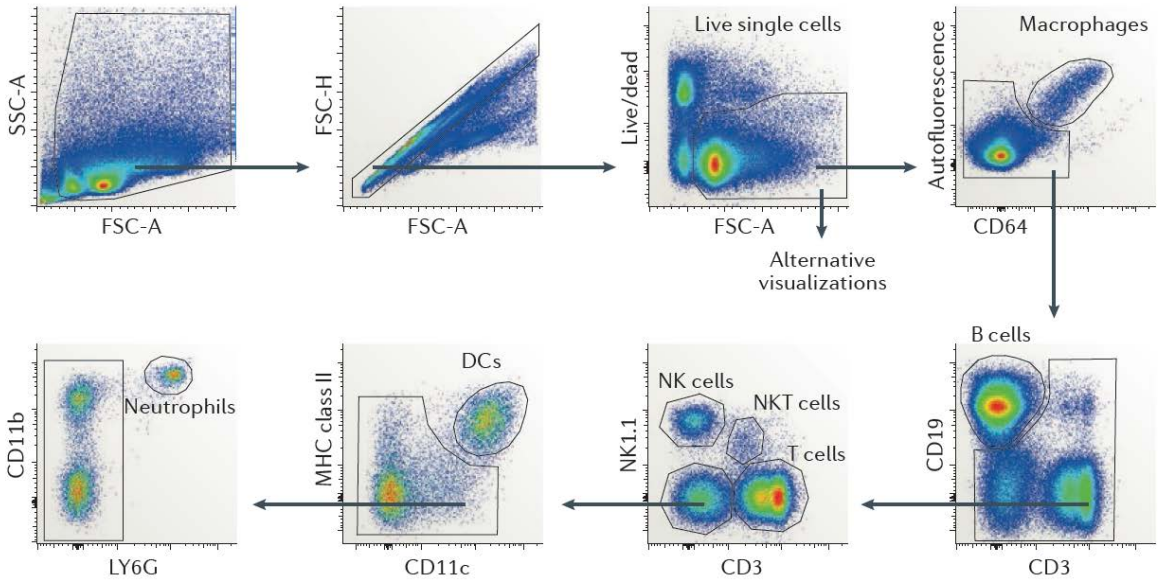


the prevalent option (see Figure 2.2 for an example). However, this manual approach has some shortcomings. First, it is subjective since it depends on the expertise of the user, on the sequence of markers (measured variables) used to do the projections and on the locations of the gates on those projections. Second, it can be very time consuming because it is 'roughly quadratic in the number of markers' (see [Li et al., 2017]). Third, the recent increase in the number of markers and number of cells per cytometry makes human error a relevant factor.

In order to avoid some of the difficulties related to manual gating there have been different approaches to automated gating. Some are unsupervised, therefore, there is no use of previously gated cytometries. Hence, gating is done through a clustering procedure. We present a small selection of such unsupervised automated gating procedures. FLOCK [Qian et al., 2010], which performs grid-based density estimation (with merging) and then k-means; FLAME [Pyne et al., 2009], which does skew t model based clustering and flowClust [Lo et al., 2008, Lo et al., 2009], which performs robust based clustering through t mixture models with Box-Cox transformation. Other related clustering procedures are: flowPeaks [Ge and Sealfon, 2012] does Gaussian mixture model based clustering (with modified covariances) and merging and flowMeans [Aghaeepour et al., 2011] performs k-means with initialization via mode detection through kernel density based estimation. More information about state of the art methods can be found in [Aghaeepour et al., 2013, Saeys et al., 2016].

Accuracy of cell type assignation can be improved using supervised machine learning where historical information is contained in previously gated cytometries (manually or otherwise). Recently, some methods have been produced addressing this problem. In [Li et al., 2017], DeepCyTOF was introduced, essentially combining de-noising, deep-learning

Figure 2.2: An example of manual gating. Taken from [Saeys et al., 2016].



algorithms and domain adaptation. In [Lux et al., 2018], flowLearn was introduced, combining density features of the data, manually selected gating thresholds and derivative-based density alignments. We stress that other more classical approaches for supervised learning are also available. For example, random forest algorithms, support vector machines or quadratic discriminant analysis can be used when learning from some previously gated cytometry. Supervised machine learning is a well documented topic and for more detailed explanations we refer to [Alpaydin, 2014].

There are two main set-ups for using supervised learning in the FC context. First, the classical one, where there is an available data base of historical information. This means that a collection of gated flow cytometries is available and we want to use this information in order to gate a new cytometry. Second, an alternative one, where we have a collection of ungated cytometries and it is required to gate manually a minimal amount of them and use these gated cytometries to classify the rest of the cytometries in the collection. In both set-ups there is a fundamental problem intrinsic to FC. That is, flow cytometry data has considerable technical and biological variability, as we mentioned in the Introduction.

In this chapter we provide novel methods for grouping (clustering) gated cytometries. By clustering a set of cytometries we are producing groups (clusters) of cytometries that have lower variability than the whole collection. This in turn allows to improve greatly the performance of any supervised learning procedure. We provide evidence of this below. Once we have a partition (clustering) of a collection of cytometries, we provide several methods for obtaining an artificial cytometry (prototype, template) that represents in some optimal way the cytometries in each respective group. These prototypes can be used, among other things, for matching populations between different cytometries as suggested in [Azad et al., 2012, Hsiao et al., 2016]. Even more, a procedure able to group similar cytometries could help to detect individuals with a common particular condition, i.e, a particular sickness (see Section 2.3.3).

optimalFlowTemplates is our procedure for clustering cytometries and obtaining tem-

plates. It is based on recent developments in the field of optimal transport such as a *similarity distance* between clusterings, recall (11), introduced in [Coen et al., 2010], and a *barycenter* (Frechet mean, see [Gouic and Loubes, 2017, Boissard et al., 2015]) and *k-barycenters* (see [Álvarez Esteban et al., 2016, del Barrio et al., 2019a, Álvarez Esteban et al., 2018]) of probability distributions, recall (14).

We introduce *optimalFlowClassification*, a supervised classification tool for the case when a database of gated cytometries is available. The procedure uses the prototypes obtained by optimalFlowTemplates on the database. These are used to initialise *tclust*, a robust extension of k-means that allows for non-spherical shapes, for gating a new cytometry (see [García-Escudero et al., 2008], not to be confused with TCLUST, [Dost et al., 2011]). By using a similarity distance between the best clustering obtained by *tclust* and the artificial cytometries provided by optimalFlowTemplates we can assign the new cytometry to the most similar template (and the respective group of cytometries). We provide several options of how to assign cell types to the new cytometry using the most relevant information, represented by the assigned template and the respective cluster of cytometries.

2.2 Methods

We start with the mathematical treatment of flow cytometry data. We can view a gated flow cytometry, say X^i, as a collection of n_i multidimensional points with their associated labels (cell types or group labels) forming a set $L^i = \{L^i_k\}_{k=1}^{k_i}$ of k_i different labels. Hence, a gated cytometry can be described as $X^i = \{(X^i_j, Y^i_j)\}_{j=1}^{n_i}$ where $X^i_j \in \mathbb{R}^d$ and $Y^i_j \in L^i$. Alternatively we could describe it as a partition (clustering) of all X^i_j into groups (clusters) formed by points sharing the same labels. That is, $\mathcal{C}^i = \{(\mathcal{C}^i_k, p^i_k)\}_{k=1}^{k_i}$ where $\mathcal{C}^i_k = \{X^i_j : 1 \leq j \leq n_i, Y^i_j = L^i_k\}$ is a cluster and p^i_k is a weight associated with label L^i_k. A third useful description is to view a gated cytometry as a clustering but coming from a mixture of location-scatter multivariate distributions. With some abuse of notation $\mathcal{C}^i = \{(m^i_k, S^i_k, p^i_k)\}_{k=1}^{k_i}$ where m^i_k, S^i_k are the multivariate mean and covariance of the points in cluster $\mathcal{C}^i_k$.

We provide an example of the different descriptions in Figure 2.3. We have five cell types, hence $L^1 = \{Basophils\,(black), CD4+CD8-\,(red), Eosinophils\,(green), Monocytes\,(blue), Neutrophils\,(Cyan)\}$. We have a three dimensional projection on to three different markers. We can interpret the image on the left as a plot of the coordinates of every cell with its label, but also as the plot of the group of cells labelled as Basophils (black group), and so on... On the other hand, the plot on the right is a representation of the ellipsoid containing 95% of the probability when we see each cluster as a multicvariate normal distribution with mean and covariance corresponding to the empirical mean and covariance. As we see from the plots, all of the above descriptions seem to represent well the data at hand and therefore all of them could be useful for different applications.

2.2.1 optimalFlowTemplates

Due to the the high variability in flow cytometry data we should expect that learning form different elements in the database should produce significantly different results on

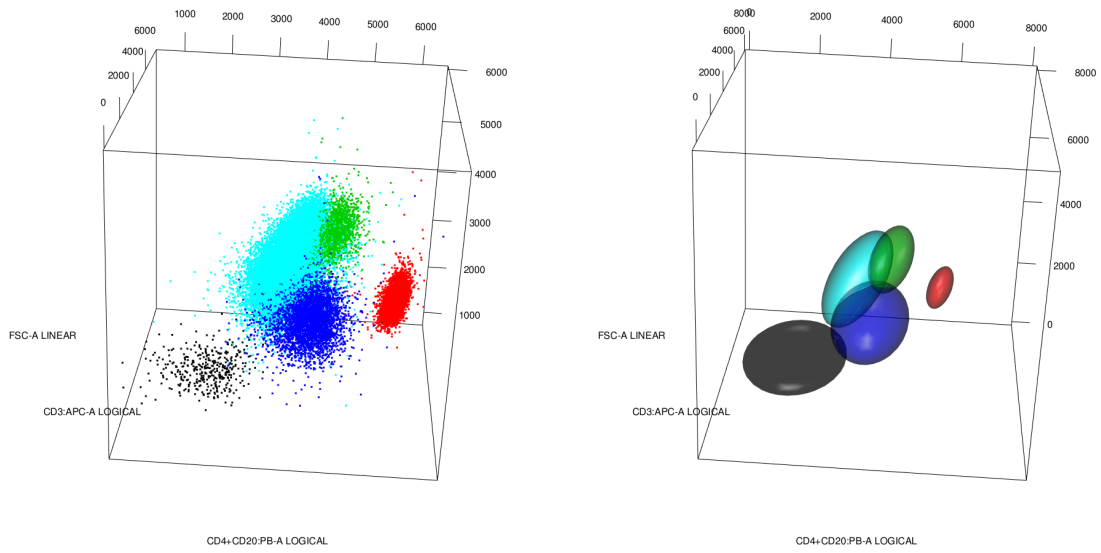


Figure 2.3: Five cell types viewed in a three dimensional projection: left as points with labels and right as 95% ellipsoids of multivariate normal distributions.

the classification of a new cytometry $X^T = \{X_1^T, \ldots, X_{n_T}^T\} \subset \mathbb{R}^d$. Our approach is to search for clusters of existing cytometries in the database. In this way we pursuit for a notable reduction of variability thus allowing a good representation of the cytometries in each of these groups through prototypic cytometries. Therefore, using a prototype of a group for learning should produce a similar result for classifying X^T to the one obtained using any other cytometry in the same group.

Clustering cytometries

Since gated cytometries can be viewed as partitions (clusterings) and we want to clusterize cytometries in order to reduce variability, we want to do clustering of clusterings, also known as metaclustering. The methodology we will develop in this chapter is to use some meaningful distance between partitions and then apply hierarchical clustering methods. As a distance between clusterings we propose to use the *similarity distance* introduced in [Coen et al., 2010] and that we have writen down in (11). We use hierarchical clustering since it does not rely on a particular distance and therefore it is well suited for handling the similarity distance. This is not the case in many other usual clustering procedures.

Recall that, d_{OT}, defined in (9), measures the cost of the optimal way of transforming one partition into the other. On the other hand, the naive transport distance, d_{NT}, defined in (10), measures the cost of naively transforming one partition into the other.

In order to completely define d_S, we need to specify a distance between clusters. Our choice is to use the well known Wasserstein distance (see Introduction) so

$$d(\mathcal{C}_k^i, \mathcal{C}_l^j) = \mathcal{W}_2(N(m_k^i, S_k^i), N(m_l^j, S_l^j)). \tag{2.1}$$

In essence, we are treating clusters as multivariate normal distributions, $N(m_k^i, S_k^i)$ and

Algorithm 2 optimalFlowTemplates

Input: $X^1, \ldots, X^N$, *equal.weights*

for $i \leq N$ **do**
  while $k \leq k_i$ and $|\mathcal{C}_k^i|$ enough for covariance estimation **do**
    $m_k^i \leftarrow \text{mean}\, \mathcal{C}_k^i$; $S_k^i \leftarrow \text{cov}\, \mathcal{C}_k^i$
    if $equal.weights = True$ **then**
      $p_k^i \leftarrow 1/k_i$
    else
      $p_k^i \leftarrow |\mathcal{C}_k^i| / \sum_{k=1}^{k_i} |\mathcal{C}_k^i|$
    end if
    $\mathcal{C}_k^i \leftarrow (m_k^i, S_k^i, p_k^i)$
  end while
end for
for $i \leq N$ **do**
  for $i < j \leq N$ **do**
    $S_{ij} \leftarrow d_S(\mathcal{C}^i, \mathcal{C}^j)$
  end for
end for
$\mathfrak{T} \leftarrow$ hierarchical clustering with S
for $i \leq |\mathfrak{T}|$ **do**
  $\mathcal{T}^i \leftarrow$ template obtention on cytometries in $\mathfrak{T}_i$
end for
$\mathcal{T} = \{\mathcal{T}^i, \ldots, \mathcal{T}^{|\mathfrak{T}|}\}$

Output: $\mathfrak{T}, \mathcal{T}$

$N(m_l^j, S_l^j)$, with means and covariances calculated from the clusters. Our choice of the Wasserstein distance is based on the desire to account for the spatial shapes of the clusters and to obtain templates for the groups of cytometries. We stress that all results in this chapter are also valid when understanding clusters as members of a location-scatter family.

Another interesting measure for cluster difference is, $\mathcal{W}_\gamma(\mathcal{C}_k^i, \mathcal{C}_l^j)$, the (entropy) regularized Wasserstein distance, where clusters are understood as empirical distributions. We recall that the entropy regularized Wasserstein distance is strictly convex and there are efficient solutions based on the Sinkhorn algorithm (see [Cuturi and Doucet, 2014]). For a fixed $\gamma > 0$ the regularized Wasserstein distance is defined as

$$\mathcal{W}_\gamma(\mu, \nu) = \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^* \|x_i - y_j\|^2 + \gamma \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}^* \log w_{ij}^*,$$

where (w_{ij}^*) are the solutions of the optimal transport linear program

$$\begin{array}{lll} \text{minimize} & \sum_{i=1}^n \sum_{j=1}^m w_{ij}\|x_i - y_j\|^2 + \gamma \sum_{i=1}^m \sum_{j=1}^m w_{ij} \log w_{ij} & \\ \text{subject to} & w_{ij} \geq 0, & 1 \leq i \leq n, 1 \leq j \leq m \\ & \sum_{j=1}^m w_{ij} = p_i, & 1 \leq i \leq n \\ & \sum_{i=1}^n w_{ij} = q_j, & 1 \leq j \leq m \\ & \sum_{i=1}^n \sum_{j=1}^m w_{ij} = 1. & \end{array}$$

However, any other dissimilarity measure can be used, for example the symmetric Kullback-Leibler was used in [Azad et al., 2012] where

$$d_{KL}(\mathcal{C}_k^i, \mathcal{C}_l^j) = \frac{1}{2}\left(KL(N(m_k^i, S_k^i)\|N(m_l^j, S_l^j)) + KL(N(m_l^j, S_l^j)\|N(m_k^i, S_k^i))\right) \tag{2.2}$$

or Friedman-Rafsky test statistic was used in [Hsiao et al., 2016], in the context of cluster comparison in flow cytometry.

When we see clusters as collections of points, and we have different clusterings of the same data, the Adjusted Rand Index, the Jaccard distance or other similar can be used, at the expense of loosing spatial information.

The clustering of cytometries is presented in lines 1-17 in Algorithm 2, resulting in a partition, $\mathfrak{T} = \{\mathfrak{T}_1, \ldots, \mathfrak{T}_{|\mathfrak{T}|}\}$, of the input cytometries. Lines 12-16 are concerned with the obtention of a distances matrix S that in line 17 is used to perform hierarchical clustering. Classical agglomerative algorithms can be used, but also density based algorithms as DBSCAN and HDBSCAN.

Template obtention through consensus clustering

Once we have a partition, $\mathfrak{T}$, of the collection of cytometries $\{\mathcal{C}^j\}_{j=1}^N$, we want to obtain a prototype cytometry, $\mathcal{T}^i$, for every group of cytometries, i, in the partition $\mathfrak{T}$ (lines 18-21 in Algorithm 2). To address this goal we resort to k-barycenters using Wasserstein distance, which provide a suitable tool for consensus on probability distributions (see Introduction). We propose three different methods on how to obtain a template cytometry from a group of cytometries, that is, on how to do consensus (ensemble) clustering on flow cytometries.

The intuition behind pooling (Algorithm 3), is to take advantage of the fact that we have groups of similar cytometries and that cell types are known. A prototype of a

Algorithm 3 Pooling. Only possible when $\{L^i\}_{i=1}^N \subset L = \{L_1, \ldots, L_K\}$. This is the case for a set of gated cytometries with identified cell populations.

Input: $\mathcal{C}^1, \ldots, \mathcal{C}^N$, $\mathfrak{T}$

1: **for** $j \leq K$ **do**
2: $C_{ij} \leftarrow$ set of all clusters associated with label L_j for the cytometries in group $\mathfrak{T}_i$.
3: **if** $|C_{ij}| > 0$ **then**
4: $\mathcal{T}_j^i \leftarrow$ take 1-barycenter of the clusters in C_{ij} viewed as multivariate normals.
5: **else**
6: $\mathcal{T}_j^i$ is empty
7: **end if**
8: **end for**
9: $\mathcal{T}^i \leftarrow \{\mathcal{T}_1^i, \ldots, \mathcal{T}_K^i\}$

Output: $\mathcal{T}^i$

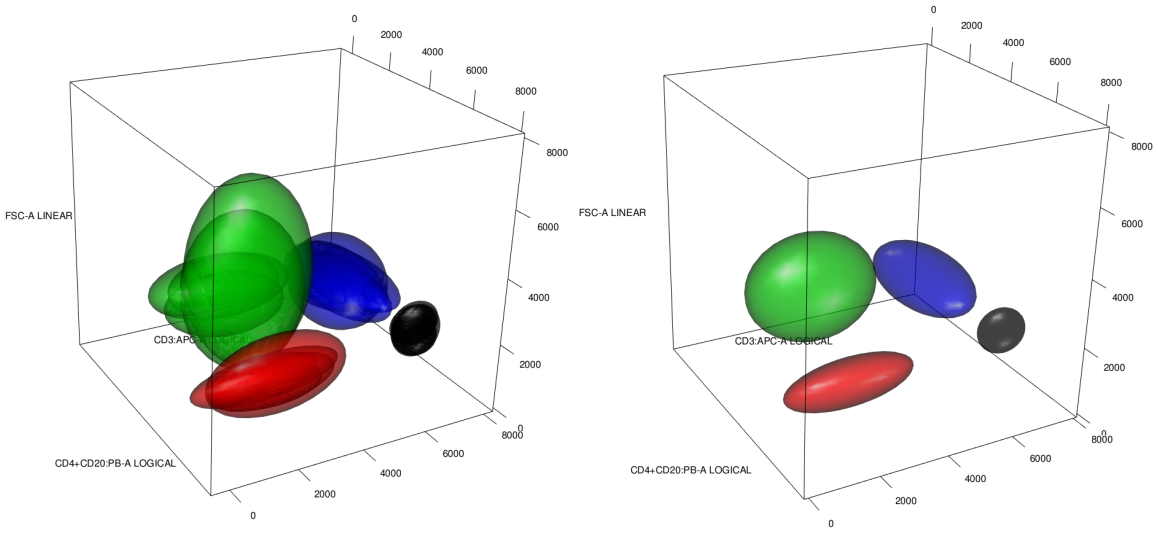


Figure 2.4: An application of Algorithm 3-Pooling.

Algorithm 4 Density based hierarchical clustering

Input: $\mathcal{C}^1, \ldots, \mathcal{C}^N$, $\mathfrak{T}$

$C^i \leftarrow$ set formed by every cluster of every cytometry in group $\mathfrak{T}_i$.
for $j, k \leq |C^i|$ **do**
    $W_{jk} \leftarrow \mathcal{W}_2(N(m_j^i, S_j^i), N(m_k^i, S_k^i))$
end for
$T \leftarrow$ partition using density based hierarchical clustering on W.
for $j \leq |T|$ **do**
    $\mathcal{T}_j^i \leftarrow$ barycenter of elements with label j in T.
end for
$\mathcal{T}^i \leftarrow \{\mathcal{T}_1^i, \ldots, \mathcal{T}_{|T|}^i\}$

Output: $\mathcal{T}^i$

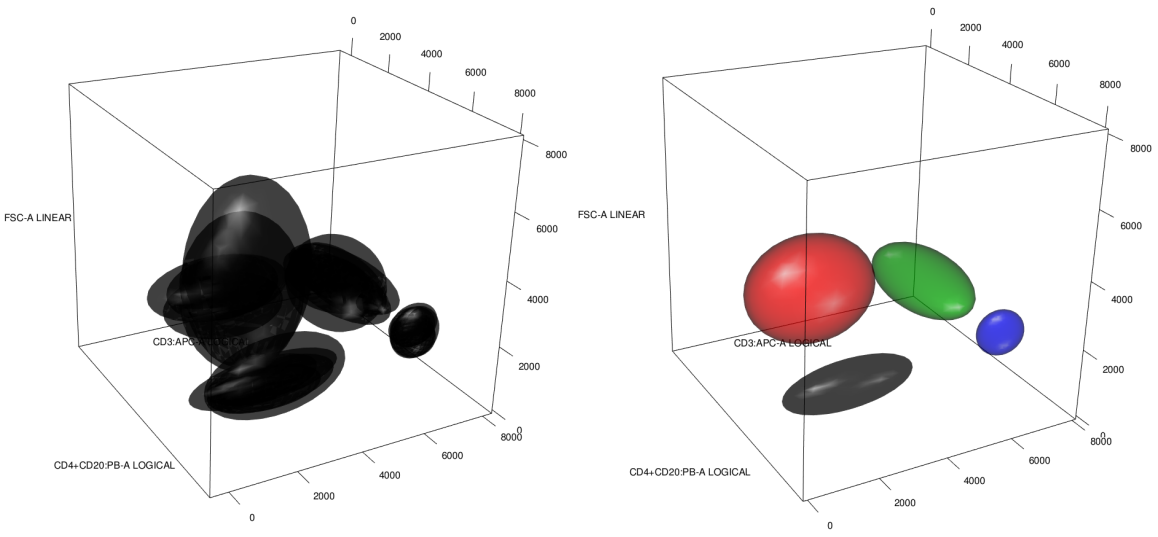


Figure 2.5: Application of Algorithm 4 - Density based.

Algorithm 5 k-barycenter

Input: $\mathcal{C}^1, \ldots, \mathcal{C}^N$, $\mathfrak{T}$, K

$C^i \leftarrow$ set formed by every cluster of every cytometry in group $\mathfrak{T}_i$.
$\mathcal{T}^i \leftarrow K$-barycenter of the elements in C^i.

Output: $\mathcal{T}^i$

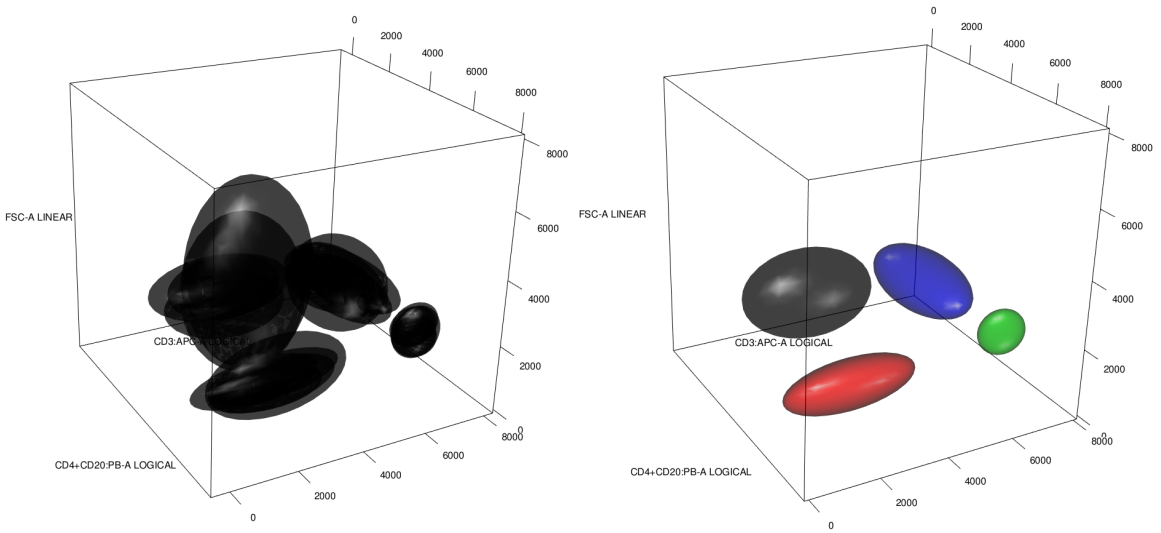


Figure 2.6: Application of Algorithm 5 - 4-barycenter.

cell type is the (1-)barycenter, a consensus representation, of the clusters (multivariate distributions) representing the same cell type in the cytometries that are members of the same group in $\mathfrak{T}$. A prototype cytometry is the collection of prototypes of each cell type. This can be seen in Figure 2.4. In the left hand side we have 5 different cytometries where each has 4 different cell types, hence $L = \{Monocytes\,(black), CD4 + CD8 - (red), Mature\,SIg\,Kappa\,(green), TCRgd - (blue)\}$. Taking advantage of the fact that cell types are known we take all the black ellipsoids of the left plot, representing the different normal distributions, and obtain the black ellipsoid, the barycenter of the group of normal distributions, as a consensus element for Monocytes. Doing this for every cell type gives us the prototype cytometry represented on the right of Figure 2.4.

However, our templates could be obtained even when we have gated cytometries but without cell type identification between them. This could be the case when unsupervised gating is used to obtain a database. Density based hierarchical clustering (Algorithm 4) and k-barycenter (Algorithm 5) are based on the idea that clusters that are close in Wasserstein distance should be understood as representing the same cell type, although we may not know which cell type. When using k-barycenters we have to specify the number of cell types, K, that we want for the artificial cytometry. However, when using density based hierarchical clustering as HDBSCAN (see [Campello et al., 2013]) or DBSCAN (see [Ester et al., 1996]) the selection of the number of cell types for the prototype cytometry is automatic. Recall that both k-barycenters, through trimming, and density based hierarchical clustering are robust clustering procedures.

In Figure 2.5 and 2.6 we have a representation of how Algorithm 4 and 5 work. Since we do not have cell type information for the 5 gated cytometries, we have the plot that can be seen on the left of Figure 2.5 and 2.6. However, the absence of this information can be mitigated using the spatial information, which clearly shows a group structure between the ellipsoids. We use density based hierarchical clustering and k-barycenters respectively

to try to capture this spatial information. As a result we obtain the template cytometries on the right of Figure 2.5 and 2.6. Clearly we see that the templates represent well the real cell types behind the cytometries (compare with Figure 2.4), although we still do not know the cell types corresponding to each ellipsoid. This could be achieved using expert information or matching populations.

2.2.2 optimalFlowClassification

Now, our goal is to do supervised classification, i.e., assign cell types to a new cytometry X^T, using the information given in a database of gated cytometries $\{\mathcal{C}^i\}_{i=1}^N$. The different sources of variability, mainly those of technical nature and those which are properly due to different cohorts present in the database, advise to search for different cytometric structures. Hence, we should assign X^T to the group of cytometries that is more similar to it and then use supervised techniques. Indeed, this is the purpose of optimalFlowClassification, as shown in Algorithm 6. As an input we apply optimalFlowTemplates to the database $\{\mathcal{C}^i\}_{i=1}^N$ in order to obtain the partition $\mathfrak{T}$ and the templates $\mathcal{T}$.

Algorithm 6 optimalFlowClassification

Input: $X^T = \{X_1^T, \ldots, X_{n_T}^T\}$, $\mathfrak{T}$,$\mathcal{T}$

1: **for** $i \leq |\mathfrak{T}|$ **do**
2: $\quad \mathcal{C}^{i,u} \leftarrow$ *tclust* on X_T initialized with $\mathcal{T}^i$
3: **end for**
4: $\mathcal{C}^u \leftarrow \arg\max$ of *tclust* objective function over all $C^{i,u}$
5: **for** $i \leq |\mathfrak{T}|$ **do**
6: $\quad S_i \leftarrow d_S(\mathcal{C}^u, \mathcal{T}^i)$
7: **end for**
8: $\mathcal{T}^* \leftarrow \mathcal{T}^{\arg\min S_i}$; $\mathfrak{T}_* \leftarrow \mathcal{T}_{\arg\min S_i}$
9: $\mathcal{C}^T \leftarrow$ labelling of X^T using transfer labelling or supervised classification based on $\mathcal{T}^*$ or $\mathfrak{T}_*$.

Output: $\mathcal{C}^T$

Lines 1-4 in Algorithm 6 are dedicated to finding an unsupervised partition of the new cytometry X^T using as initialization for *tclust* the prototypes of the database. Initializing with the database entries attempts to use optimally the available information. Hence, if X^T is similar to some of the cytometries in the database, appropriate initialization should be advantageous. However, some other suitable unsupervised initializations can be used, as the ones proposed in FLOCK, flowPeaks or flowMeans. We need to cluster X^T in order to compare it with the template cytometries.

Recall that *tclust*, introduced in [García-Escudero et al., 2008], is a robust model based clustering procedure that allows for non spherical clusters. Nontheless, it is possible to use any other unsupervised procedure that allows an initialization with a clustering defined by probability distributions. For example, this is the case for the popular *mclust* [Fraley and Raftery, 2002, Scrucca et al., 2016], a finite Gaussian mixture model based clustering solved by an EM-algorithm.

tclust searches for a partition $\{\mathcal{C}_0, \ldots, \mathcal{C}_k\}$ of $X = \{X_1, \ldots, X_n\}$, with $|\mathcal{C}_0| = \lceil n\alpha \rceil$, vectors m_j, positive definite matrices S_j and weights $p_j \in [0, 1]$ that approximately maximize

the pseudo-likelihood

$$\sum_{j=1}^{k} \sum_{i \in \mathcal{C}_j} \log \left(p_j \varphi(X_i; m_j, S_j) \right), \tag{2.3}$$

under restrictions over the scatter matrices S_j. By $\varphi(\cdot; m_j, S_j)$ we denote the density function of the multivariate normal $N(m_j, S_j)$. $\mathcal{C}_0$ is the cluster of trimmed observations, where the trimming level is α.

The details of the algorithm can be found in [Fritz et al., 2013]. For us it is relevant to recall only the initialization step, i.e, to provide an initial $\theta^0 = (p_1^0, \ldots, p_k^0, m_1^0, \ldots, m_k^0, S_1^0, \ldots, S_k^0)$. Hence, to initialize *tclust* we only need a set of weights with corresponding means and covariances.

We favour the use of *tclust* over k-means since it allows for non-spherical clusters and for trimming, making partitions more robust to outliers.

In lines 5-8 we look to assign X^T, using the clustering $\mathcal{C}^u$, produced in the previous step, to the template that is closest in similarity distance to $\mathcal{C}^u$. With this we hope to use only the most relevant information of the database, summarized in $\mathcal{T}^*$ and $\mathfrak{T}_*$.

The last step in algorithm 6, line 9, is concerned with assigning cell types to X^T. To do this we have several options. We can try to relabel $\mathcal{C}^u$ in an optimal way using $\mathcal{T}^*$ or $\mathfrak{T}_*$, i.e, do label transfer. Alternatively, we can use $\mathcal{T}^*$ to do Quadratic Discriminant Analysis (QDA) or we can find the most similar partition in similarity distance (11) from $\mathfrak{T}_*$ to $\mathcal{C}^u$ and use it to do QDA or random forest classification. In short, we can do label transfer or supervised classification.

For supervised classification we use standard tools, random forest and QDA, however, other methods can be used in a straightforward fashion. We remark that when using QDA and $\mathcal{T}^*$ we are using non-linear multidimensional gating regions obtained from $\mathcal{T}^*$ in order to classify X^T. This can be taught as an extension of the method presented in [Lux et al., 2018] where only linear one-dimensional regions are used. Another interesting fact is that the use of d_S allows us to select the most similar real cytometry to C^u, hence supervised tools should be more effective.

The problem of relabelling a clustering $\mathcal{C}^j$ with respect to another clustering $\mathcal{C}^i$ is usually stated as a weighted bipartite matching problem, where weights are related to the similarity between clusters in the two partitions. This problem can be solved by the hungarian method [Kuhn, 1995]. Generalized edge cover is another possible solution to relabelling (see [Azad et al., 2012]).

Additionally we introduce an approach to obtain a fuzzy relabelling based on solving the optimal transport linear program associated to (9). The solution, (w_{kl}^*), is the base for this fuzzy relabelling. We define the score of cluster l in $\mathcal{C}^j$ to come from cluster k in $\mathcal{C}^i$ as $s_k^l = w_{kl}^*/p_l^j$. In words, s_k^l is the proportion of probability coming from cluster k, with respect to the probability in cluster l, that arrives at cluster l. Clearly, $0 \leq s_k^l \leq 1$, and the closer to 1 the score is the more evidence we have that cluster k and l represent the same cluster. A fuzzy relabelling for cluster l in $\mathcal{C}^j$ is the collection of all the scores $s^l = \{s_1^l, \ldots, s_{|\mathcal{C}^i|}^l\}$. A variation of the previous score is $\tilde{s}_k^l = s_k^l * w_{kl}^*/p_k^i$, where we are weighting by the proportion of cluster k that goes to cluster l, with respect to the probability contained in cluster k. In this way we down-weight the effect of a small proportion of a big cluster with respect to a big proportion of a small cluster arriving to l. From these fuzzy relabellings a hard relabelling can be easily obtained.

Again, a suitable distance between clusters can be the Wasserstein distance as in (2.1). However, another possibility is to use

$$d(\mathcal{C}_k^i, \mathcal{C}_l^j) = \frac{1}{|\mathcal{C}_k^i||\mathcal{C}_l^j|} \sum_{x \in \mathcal{C}_k^i} \sum_{y \in \mathcal{C}_l^j} \|x - y\|^2. \tag{2.4}$$

(2.1) is computationally very efficient but does not allow to label very small clusters in $\mathcal{C}^j$.(2.4) does allow labelling small clusters in $\mathcal{C}^j$, at the price of using sub-sampling to compare bigger clusters (for example more than 10000 points).

2.3 Results

In this section we present several experiments and comparisons of our methods with other state of the art procedures on real databases.

2.3.1 Data

We will work with two databases of gated flow cytometries obtained following the Euroflow protocols, kindly provided by *Centro de Investigación del Cancer* (CIC) in Salamanca, Spain.

The first database is formed by 21 gated flow cytometries, $\{(X^i, Y^i)\}_{i=1}^{21}$ equivalently viewed as partitions $\mathcal{C} = \{\mathcal{C}^i\}_{i=1}^{21}$. All 21 cytometries have been obtained in a BD FACSCanto flow cytometer but in three different centres. The size of the cytometry datasets vary from 50,000 cells to 254,450 cells. The samples are from adult male and female individuals, with a varied range of ages, that have been diagnosed as healthy. More information about the data set can be found in Table 2.2.

Clearly, there is biological variability, since there are different individuals with different ages and other different characteristics. Moreover, we have technical variability since we have different centres, different dates of measurement and different incubation times. However, we remark that all individuals belong to the same class of healthy people.

The second database is formed by 20 gated flow cytometries but now belonging to 10 healthy and 10 sick individuals. We refer to this new collection of gated cytometries by $\mathcal{DB}_2 = \{\mathcal{C}^{1,s}, \mathcal{C}^{2,s}, \mathcal{C}^{3,s}, \mathcal{C}^{4,s}, \mathcal{C}^{5,s}, \mathcal{C}^{6,s}, \mathcal{C}^{7,s}, \mathcal{C}^{8,s}, \mathcal{C}^{9,h}, \mathcal{C}^{10,h}, \mathcal{C}^{11,h}, \mathcal{C}^{12,h}, \mathcal{C}^{13,h}, \mathcal{C}^{14,s}, \mathcal{C}^{15,s}, \mathcal{C}^{16,h}, \mathcal{C}^{17,h}, \mathcal{C}^{18,h}, \mathcal{C}^{19,h}, \mathcal{C}^{20,h}\}$, where the superindex s means sick and superindex h means healthy. The size of the cytometry datasets vary from 50,000 cells to 300,000 cells.

2.3.2 Measures of performance

We need appropriate measures of the performance of the different automated gating procedures that appear in this chapter. We recall that we use both unsupervised and supervised methods. In this set-up an appropriate tool is the *F-measure* statistic which has been used in [Aghaeepour et al., 2013, Aghaeepour et al., 2011, Ge and Sealfon, 2012, Li et al., 2017]. With our notation we have

$$F(\mathcal{C}^i, \mathcal{C}^j) = \sum_{k=1,\ldots,|\mathcal{C}^i|} \frac{|\mathcal{C}_k^i|}{M} \max_{l=1,\ldots,|\mathcal{C}^j|} F(\mathcal{C}_k^i, \mathcal{C}_l^j),$$

$$F(\mathcal{C}_k^i, \mathcal{C}_l^j) = 2\frac{R(\mathcal{C}_k^i, \mathcal{C}_l^j)P(\mathcal{C}_k^i, \mathcal{C}_l^j)}{R(\mathcal{C}_k^i, \mathcal{C}_l^j) + P(\mathcal{C}_k^i, \mathcal{C}_l^j)},$$

$$R(\mathcal{C}_k^i, \mathcal{C}_l^j) = \frac{|\mathcal{C}_k^i \cap \mathcal{C}_l^j|}{|\mathcal{C}_k^i|} \quad \text{and} \quad P(\mathcal{C}_k^i, \mathcal{C}_l^j) = \frac{|\mathcal{C}_k^i \cap \mathcal{C}_l^j|}{|\mathcal{C}_l^j|}$$

with $M = \sum_{k=1,\dots,|\mathcal{C}^i|} |\mathcal{C}_k^i| = \sum_{l=1,\dots,|\mathcal{C}^j|} |\mathcal{C}_l^j|$. We make the convention $R(\emptyset, \mathcal{C}_l^j) = P(\mathcal{C}_k^i, \emptyset) = 1$ and $R(\mathcal{C}_k^i, \emptyset) = P(\emptyset, \mathcal{C}_l^j) = 0$. Another appealing measure is the *median F-measure* used in [Lux et al., 2018] specifically for supervised learning. The formal definition is

$$\tilde{F}(\mathcal{C}^i, \mathcal{C}^j) = \text{median}\{\{F(\mathcal{C}_k^i, \mathcal{C}_{k*}^j) : k \text{ such that } L_k^i = L_{k^*}^j \in L^i \cap L^j\}, \{0\} \times |L^i \triangle L^k|\} \quad (2.5)$$

where $\mathcal{C}^i$ is the considered ground truth, in our case a manual gating, and $\mathcal{C}^j$ is another classification of the same data.

To measure how similar for learning are two gated cytometries, i.e., how well we do when learning from one to classify the other and how well we do when learning with the later to classify the former we introduce the following distance.

$$d_{learning}(X^i, X^j) = 1 - \frac{F(\mathcal{C}^j, \tilde{\mathcal{C}}^j) + F(\mathcal{C}^i, \tilde{\mathcal{C}}^i)}{2}$$

where $\tilde{\mathcal{C}}^j$ is the partition resulting from the classification of the data in X^j using a random forest learned in X^i. $\tilde{\mathcal{C}}^i$ is the partition resulting from the classification of the data in X^i using a random forest learned in X^j. This measure gives us a notion of how close in terms of being good predictors for one another are two cytometries. We have that $0 \leq d_{learning} \leq 1$, and two cytometries are interchangeable for learning if $d_{learning}$ is close to 0. A variation of this measure is

$$\tilde{d}_{learning}(X^i, X^j) = 1 - \frac{\tilde{F}(\mathcal{C}^j, \tilde{\mathcal{C}}^j) + \tilde{F}(\mathcal{C}^i, \tilde{\mathcal{C}}^i)}{2}.$$

2.3.3 Clustering and Template obtention

Suppose that we have a database, which is a subset of 15 cytometries, given by $\mathcal{DB} = \{\mathcal{C}^2, \mathcal{C}^3, \mathcal{C}^4, \mathcal{C}^5, \mathcal{C}^7, \mathcal{C}^8, \mathcal{C}^9, \mathcal{C}^{12}, \mathcal{C}^{13}, \mathcal{C}^{14}, \mathcal{C}^{15}, \mathcal{C}^{16}, \mathcal{C}^{17}, \mathcal{C}^{19}, \mathcal{C}^{21}\} \subset \mathcal{C}$. We want to compare different methods to cluster the database $\mathcal{DB}$. We use for a ground truth hierarchical clusterings obtained using $d_{learning}$ and $\tilde{d}_{learning}$. For a state of the art comparison, we use flowMatch, described in [Azad et al., 2012]. Recall that flowMatch is based in a Generalized Edge Cover procedure, a generalization of bipartite matching, where the cost between partitions is given bye

$$d(\mathcal{C}^i, \mathcal{C}^j) = \frac{1}{k_i k_j} \sum_{k=1}^{k_i} \sum_{l=1}^{k_j} d_{KL}(\mathcal{C}_k^i, \mathcal{C}_l^j),$$

where d_{KL} is as in (2.2), or

$$d(\mathcal{C}^i, \mathcal{C}^j) = \frac{1}{k_i k_j} \sum_{k=1}^{k_i} \sum_{l=1}^{k_j} d_{Mahalanobis}(N(m_k^i, S_k^i), N(m_l^j, S_l^j)),$$

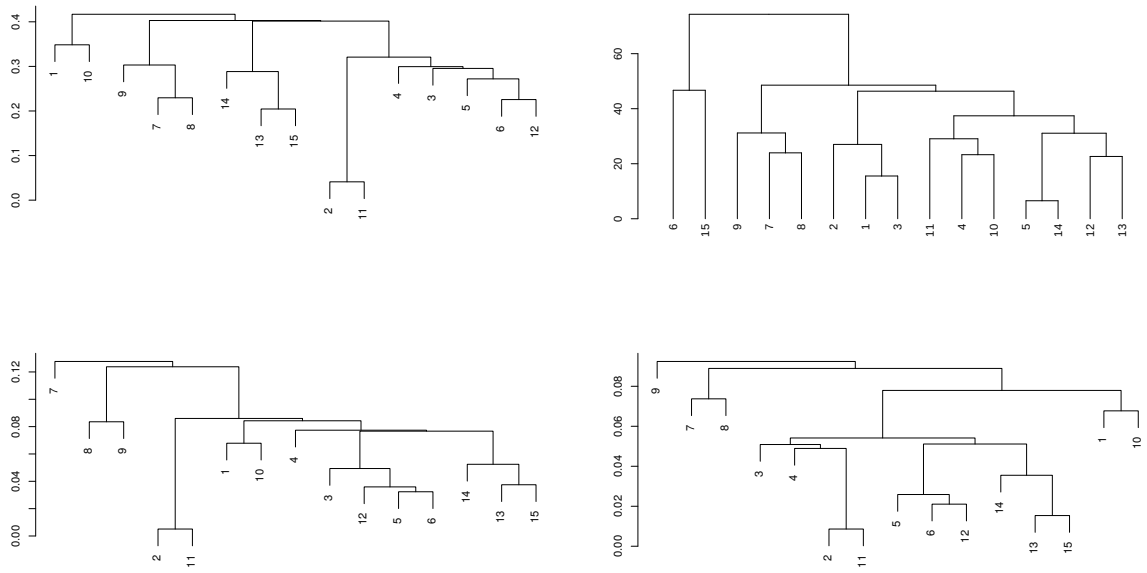


Figure 2.7: Hierarchical trees for the database $\mathcal{DB}$. Top-left: result of optimalFlowTemplates. Top-right: result of flowMatch with Mahalanobis distance. Bottom-left: single linkage with $\tilde{d}_{learning}$. Bottom-right: single linkage with $d_{learning}$.

where $d_{Mahalanobis}$ is the well known Mahalanobis distance between multivariate normals.

In Figure 2.7 we use single linkage hierarchical clustering with the distance matrices obtained with $d_{learning}$ (bottom-right) and $\tilde{d}_{learning}$ (bottom-left). We also use single linkage with optimalFlowTemplates (top-left). In the case of flowMatch we use Mahalanobis distance (top-right). In Figure 2.8 we change the clustering method to complete linkage and in the case of flowMatch we use the symmetric Kullback-Leibler divergence.

At a first glance it is clear that the results form optimalFlowTemplates are much more similar to the ground truth than those of flowMatch. This should be interpreted as the fact that optimalFlowTemlates captures more accurately the similarity between cytometries than flowMatch. Two additional facts should be stated: first, the similarity distance is independent of parameters, something that is not the case for the generalized edge cover distance used in flowMatch. Second, optimalFlowTemplates produces templates only at one stage, once the number of clusters is determined, while flowMatch produces templates at every stage of the hierarchical clustering procedure.

Our next goal is to see if our clustering of cytometries is sensitive enough to detect differences between normal and sick individuals. For this we are going to use $\mathcal{DB}_2$. On the top row of Figure 2.9 we see the results of using the manually gated database with optimalFlowTemplates. Top-right we clearly see three different groups of cytometries, two of them corresponding to sick individuals and the other one containing all the healthy individuals. Therefore, it seems that our method has enough sensitivity to differentiate healthy from sick individuals. On the middle row of Figure 2.9 we present the results when using flowMatch. Clearly this method is not as sensitive as ours, since some of

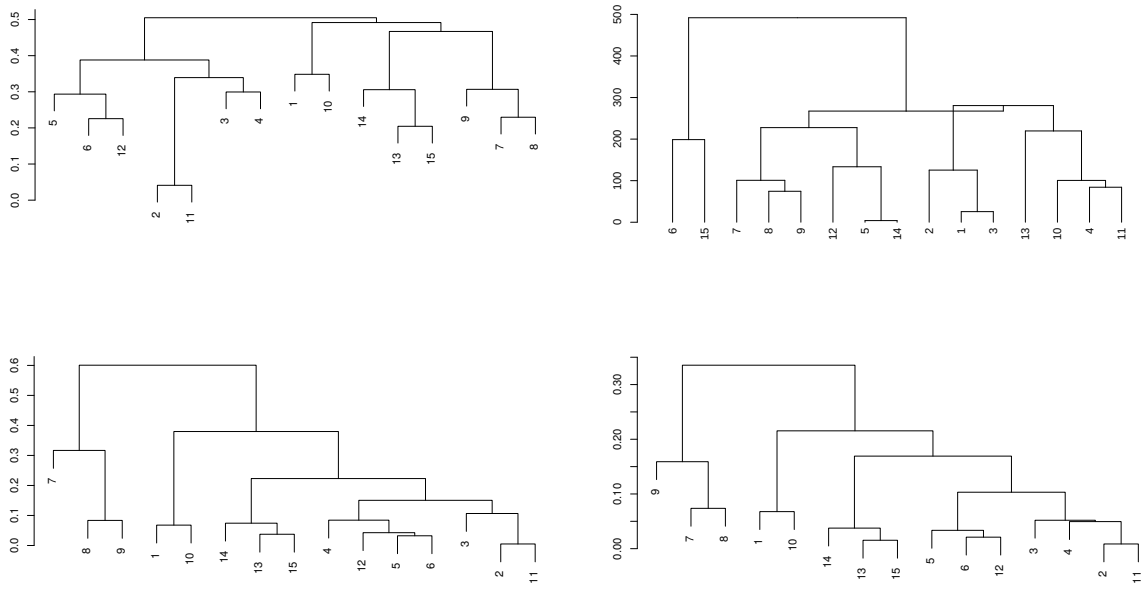


Figure 2.8: Hierarchical trees for the database $\mathcal{DB}$. Top-left: result of optimalFlowTemplates. Top-right: result of flowMatch with symmetric Kulback-Leibler divergence. Bottom-left: complete linkage with $\tilde{d}_{learning}$. Bottom-right: complete linkage with $d_{learning}$.

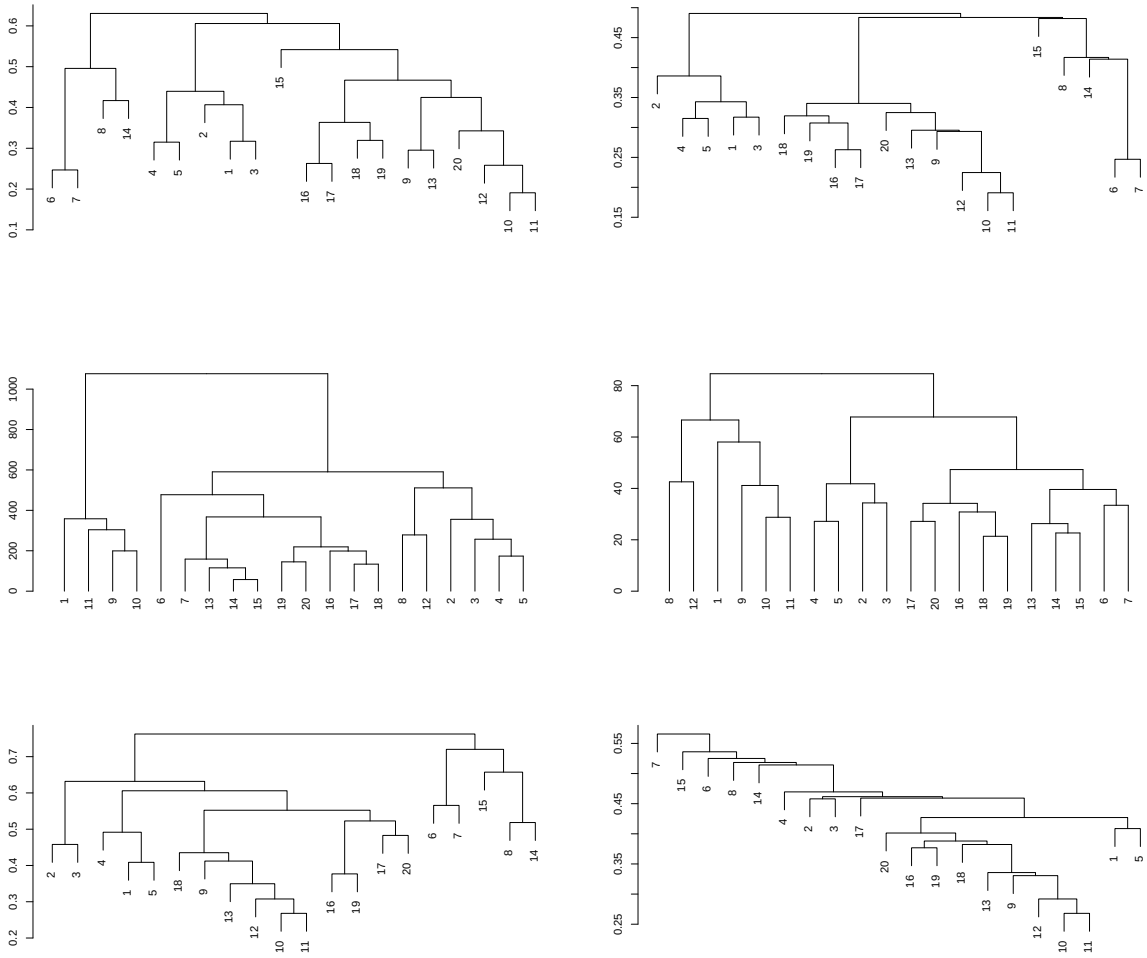


Figure 2.9: Hierarchical trees for the database $\mathcal{DB}_2$. Top-left: result of optimalFlowTemplates with complete linkage. Top-right: result of optimalFlowTemplates with single linkage. Middle-left: flowMatch with symmetric Kullback-Libler. Middle-right: flowMatch with Mahalanobis. Bottom-left: result of optimalFlowTemplates with complete linkage and unsupervised gating with flowMeans. Bottom-right: result of optimalFlowTemplates with single linkage and unsupervised gating with flowMeans.

	DeepCyTOF	DeepCyTOF 2	flowMeans	*tclust*	optFlowC
$\mathcal{C}^1$	0.9641	***0.9857***	0.9501	0.9504	0.9740
$\mathcal{C}^2$	0.9420	**0.9585**	0.8988		
$\mathcal{C}^5$	0.8728	0.8720	**0.8977**		
$\mathcal{C}^6$			0.9195	0.9335	**0.9522**
$\mathcal{C}^7$	0.8763	0.8062	**0.9508**		
$\mathcal{C}^{10}$			0.8610	0.8141	**0.9595**
$\mathcal{C}^{11}$			0.8653	0.9170	**0.9256**
$\mathcal{C}^{14}$	***0.9825***	0.9295	0.9004		
$\mathcal{C}^{17}$	0.6982	***0.9816***	0.8978		
$\mathcal{C}^{18}$	0.6840	0.9797	0.9003	0.8716	**0.9853**
$\mathcal{C}^{20}$	0.6884	0.9760	0.9223	0.8661	**0.9834**
$\mathcal{C}^{21}$	0.6942	**0.9699**	0.9269		

Table 2.1: Table of F-measure statistics. DeepCyTOF: results of deepCyTOF on $\mathcal{S}$. DeepCyTOF 2: results of deepCyTOF on $\mathcal{S}_1$ and $\mathcal{S}_2$. flowMeans: results of flowMeans. *tclust*: results of optimalFlowTemplates initialized *tclust* on $\mathcal{TS}$. optimalFlowC: results ot optimalFlowClassification on $\mathcal{TS}$.

the groups clearly contain healthy and sick individuals. Indeed, optimalFlowTemplates also outperforms flowMatch in this case. In the bottom row of Figure 2.9, we offer a slightly modified example. In this case the gating has been done in an unsupervised fashion using flowMeans. It is remarkable, that even in this case, for example bottom-left, optimalFlowTemplates is able to separate quite well sick from healthy individuals.

2.3.4 Gating and Classification

We will apply optimalFlowTemplates+optimalFlowClassification to the database $\mathcal{DB}$ introduced in the previous section. We will use as a test set $\mathcal{TS} = \{\mathcal{C}^1, \mathcal{C}^6, \mathcal{C}^{10}, \mathcal{C}^{11}, \mathcal{C}^{18}, \mathcal{C}^{20}\} \subset \mathcal{C}$. For the cytometries in $\mathcal{TS}$, we also perform an unsupervised gating given by flowMeans and a semi unsupervised procedure given by *tclust* initialized with the templates obtained by optimalFlowTemplates.

Results can be seen in columns 3-5 of Table 2.1. In Table 2.3 and 2.4 we have a full description of the results of optimalFlowClassification. We see that *tclust* initialized with optimalFlowTemplates is competitive with flowMeans, but more importantly, optimalFlowTemplates+optimalFlowClassification is superior in every of the test cytometries, giving 5 form 6 F-measures higher than 0.95 and the other higher than 0.92. Clearly our supervised procedure is working well and, as expected, is giving better performance than state of the art unsupervised alternatives.

However, we also want to compare with a state of the art supervised procedure. In this case we will use deepCyTOF, with some bug corrections and some adaptations to our setting of the github version, implemented in Python with *tensorflow* 0.12 and *keras* 1.2.2. In order to use deepCyTOF we need cytometries with the same number and type of cell types so we use a data set $\mathcal{S} = \{\tilde{\mathcal{C}}^1, \tilde{\mathcal{C}}^2, \tilde{\mathcal{C}}^5, \tilde{\mathcal{C}}^7, \tilde{\mathcal{C}}^{14}, \tilde{\mathcal{C}}^{17}, \tilde{\mathcal{C}}^{18}, \tilde{\mathcal{C}}^{20}, \tilde{\mathcal{C}}^{21}\}$, where we have eliminated one group from each cytometry in order to make them fulfil the mentioned

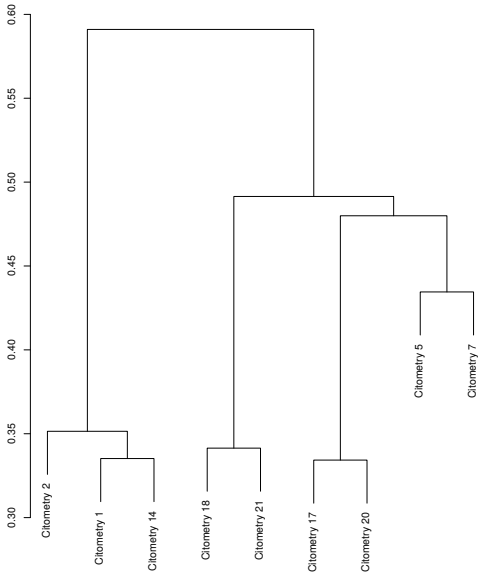


Figure 2.10: Result of optimalFlowTemplates on the databse $\mathcal{S}$ after gating each cytometry with flowMeans.

requirements. We recall that deepCyTOF only uses the supervised information of one of the cytometries in $\mathcal{S}$ to classify all other members. We see the results of deepCyTOF, with domain adaptation and without de-noising, since all entries are classified, in column 1 of Table 2.1. DeepCyTOF's performance is rather poor, achieving worst F-measure than flowMeans in 6 of the 9 cases and also for all applicable cases (cytometries 1,18,20) than optimalFlowTemplates+optimalFlowClassification.

However, these poor results are due to the high variability of the cytometries that can not be accommodated by the domain adaptation procedure of deepCyTOF. Hence if we were able to reduce this variability, deepCyTOF should give better results. Indeed, if we use flowMeans to gate the cytometries in $\mathcal{S}$, and then we use optimalFlowTemplates, we obtain the hierarchical tree presented in Figure 2.10. It suggest to split $\mathcal{S}$ into $\mathcal{S}_1 = \{\tilde{\mathcal{C}}^1, \tilde{\mathcal{C}}^2, \tilde{\mathcal{C}}^{14}\}$ and $\mathcal{S}_2 = \{\tilde{\mathcal{C}}^5, \tilde{\mathcal{C}}^7, \tilde{\mathcal{C}}^{17}, \tilde{\mathcal{C}}^{18}, \tilde{\mathcal{C}}^{20}, \tilde{\mathcal{C}}^{21}\}$. We recall that until now we have not used supervised information. Applying deepCyTOF to $\mathcal{S}_1$ and $\mathcal{S}_2$ we obtain the results in column 2 of Table 2.1. Now deepCyTOF performs better than flowMeans in 7 of the 9 cases, however it is better than optimalFlowTemplates+optimalFlowClassification only for Cytometry 1, which is the one that deepCyTOF uses for learning in $\mathcal{S}_1$.

Participant	*Final diagnosis*	*Type of tested sample*	*Type of coagulant (preservation)*	*Sex*	*Age (years)*	*Incubation period*	*Type of Flow Cytometer*
Centre 1	HD	PB	EDTA	M	53	30 min	BD FACSCanto
Centre 1	HD	PB	EDTA	M	50	30 min	BD FACSCanto
Centre 1	HD	PB	EDTA	M	61	30 min	BD FACSCanto
Centre 2	HD	PB	Heparin	M	29	30 min	BD FACSCanto
Centre 2	HD	PB	Heparin	M	38	30 min	BD FACSCanto
Centre 2	HD	PB	Heparin	F	27	30 min	BD FACSCanto
Centre 2	HD	PB	Heparin	F	NA	30 min	BD FACSCanto
Centre 2	HD	PB	Heparin	M	NA	30 min	BD FACSCanto
Centre 2	HD	PB	Heparin	F	NA	30 min	BD FACSCanto
Centre 2	HD	PB	Heparin	F	NA	30 min	BD FACSCanto
Centre 3	HD	PB	NA	M	34	15 min	BD FACSCanto
Centre 3	HD	PB	NA	F	33	15 min	BD FACSCanto
Centre 3	HD	PB	NA	M	32	15 min	BD FACSCanto
Centre 3	HD	PB	NA	M	33	15 min	BD FACSCanto
Centre 3	HD	PB	NA	F	35	15 min	BD FACSCanto
Centre 3	HD	PB	EDTA	NA	Adult	15 min	BD FACSCanto
Centre 3	HD	PB	EDTA	NA	Adult	15 min	BD FACSCanto
Centre 3	HD	PB	EDTA	NA	Adult	15 min	BD FACSCanto
Centre 3	HD	PB	EDTA	NA	Adult	15 min	BD FACSCanto
Centre 3	HD	PB	EDTA	NA	Adult	15 min	BD FACSCanto

Table 2.2: Detailed information about the participants and the measurments for 20 of the 21 cytometries.

Cytometry 1						
Cytometries clustering	Templates formation	Method 1	Method 2	Method 3	Method 4	Method 5
complete, k = 5	pooling	0.9064	0.9339	0.8992	0.9196	0.8521
HDBSCAN	pooling	0.9064	0.9323	0.8992	0.9196	0.8521
complete, k = 5	HDBSCAN	0.0000	0.9398	0.8825	0.3542	0.7429
HDBSCAN	HDBSCAN	0.0000	0.9111	0.7235	0.5263	0.9186
complete, k = 5	37-barycenter, alpha = 0.05	0.0000	0.9498	**0.9512**	0.2298	0.8707
HDBSCAN	37-barycenter, alpha = 0.05	0.0000	0.9485	0.7755	0.6846	0.5988
Ctometry 6						
Cytometries clustering	Templates formation	Method 1	Method 2	Method 3	Method 4	Method 5
complete, k = 5	pooling	0.7692	0.8524	0.7613	0.8373	0.8373
HDBSCAN	pooling	0.8787	0.8571	0.7762	0.8427	0.8427
complete, k = 5	HDBSCAN	0.7212	0.9265	0.8398	0.0653	0.8163
HDBSCAN	HDBSCAN	0.8270	**0.9276**	0.8399	0.7924	0.8034
complete, k = 5	37-barycenter, alpha = 0.05	0.0000	0.9184	0.8399	0.8110	0.7572
HDBSCAN	37-barycenter, alpha = 0.05	0.1468	0.9112	0.7849	0.6314	0.6314
Citometry 10						
Cytometries clustering	Templates formation	Method 1	Method 2	Method 3	Method 4	Method 5
complete, k = 5	pooling	**0.9525**	0.9431	0.9440	0.9306	0.9306
HDBSCAN	pooling	**0.9525**	0.9421	0.9440	0.9306	0.9306
complete, k = 5	HDBSCAN	0.9009	0.9451	0.9445	0.0739	0.8181
HDBSCAN	HDBSCAN	0.9009	0.9423	0.9445	0.0739	0.8181
complete, k = 5	37-barycenter, alpha = 0.05	0.0000	0.9345	0.9407	0.9358	0.1163
HDBSCAN	37-barycenter, alpha = 0.05	0.0000	0.6694	0.6325	0.7901	0.1610

Table 2.3: Median F-measure for $\mathcal{C}^1, \mathcal{C}^6$ and $\mathcal{C}^{10}$ for different combinations of our procedures. *Cytometries clustering* indicates the method used for clustering in optimalFlowTemplates. *Templates formation* indicates the method used for obtaining templates for the clusters of cytometries. Method 1 refers to using the best template and QDA. Method 2 refers to using random forest with the cytometry closest in similarity distance in the assigned group. Method 3 is like Method 2 but using QDA. Method 4 is label matching between the best template and the best *tclust* result. Method 5 is label matching through a vote between label matchings of every cytometry in the best group.

Citometry 11						
Cytometries clustering	Templates formation	Method 1	Method 2	Method 3	Method 4	Method 5
complete, k = 5	pooling	0.9069	**0.9506**	0.9336	0.9378	0.9363
HDBSCAN	pooling	0.9069	0.9496	0.9336	0.9378	0.9363
complete, k = 5	HDBSCAN	0.4304	0.7500	0.6864	0.1264	0.5928
HDBSCAN	HDBSCAN	0.4304	0.7556	0.6864	0.1264	0.5928
complete, k = 5	37-barycenter, alpha = 0.05	0.6319	0.9383	0.9116	0.6206	0.9095
HDBSCAN	37-barycenter, alpha = 0.05	0.0000	0.9494	0.9116	0.7151	0.8975
Citometry 18						
Cytometries clustering	Templates formation	Method 1	Method 2	Method 3	Method 4	Method 5
complete, k = 5	pooling	0.9088	0.9611	0.9038	0.6433	0.3232
HDBSCAN	pooling	0.8465	0.8585	0.7045	0.7901	0.4740
complete, k = 5	HDBSCAN	0.9043	**0.9618**	0.9171	0.6415	0.7912
HDBSCAN	HDBSCAN	0.9043	0.9617	0.9171	0.2133	0.6214
complete, k = 5	37-barycenter, alpha = 0.05	0.0000	0.9617	0.9171	0.7879	0.7895
HDBSCAN	37-barycenter, alpha = 0.05	0.0000	0.8501	0.9171	0.8468	0.7911
Citometry 20						
Cytometries clustering	Templates formation	Method 1	Method 2	Method 3	Method 4	Method 5
complete, k = 5	pooling	0.9140	0.9557	0.9046	0.7517	0.4447
HDBSCAN	pooling	0.9140	0.9537	0.9046	0.7517	0.4447
complete, k = 5	HDBSCAN	0.9140	0.9587	0.9231	0.6020	0.4346
HDBSCAN	HDBSCAN	0.9140	**0.9589**	0.9231	0.6308	0.7542
complete, k = 5	37-barycenter, alpha = 0.05	0.2538	0.9587	0.9231	0.6865	0.6198
HDBSCAN	37-barycenter, alpha = 0.05	0.0000	0.9621	0.9231	0.6354	0.4060

Table 2.4: Same as Table 2.3 but for cytometries $\mathcal{C}^{11}, \mathcal{C}^{18}$ and $\mathcal{C}^{20}$.

2.4 A short tutorial on *optimalFlow*

We start by installing the required packages from gitHub using *devtools*.

```
require(devtools)
install_github("HristoInouzhe/optimalFlowData", build_vignettes = TRUE)
install_github("HristoInouzhe/optimalFlow", build_vignettes = TRUE)
```

Next, we load the required packages.

```
library(optimalFlowData)
library(optimalFlow)
require(rgl)
```

Our tutorial will be based on the database $\mathcal{DB}$, but selecting only 4 cell types for ease of visualization. Hence, we have the following database.

```
database = list(
  Cytometry2[which(Cytometry2$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry3[which(Cytometry3$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry4[which(Cytometry4$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry5[which(Cytometry5$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry7[which(Cytometry7$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry8[which(Cytometry8$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry9[which(Cytometry9$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry12[which(Cytometry12$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry13[which(Cytometry13$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry14[which(Cytometry14$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry15[which(Cytometry15$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry16[which(Cytometry16$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry17[which(Cytometry17$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry19[which(Cytometry19$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),],
  Cytometry21[which(Cytometry21$`Population ID (name)` %in%
                   c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-")),]
  )
```

We start with a default application of *optimalFlowTemplates* looking for 5 groups of cytometries.

```
templates.optimalFlow =
  optimalFlowTemplates(
    database = database, templates.number = 5, cl.paral = 1
    )
templates.optimalFlow$clustering
[1] 1 2 3 3 3 3 4 4 4 1 2 3 5 5 5
```

We see the clustering of the cytometries in the database in the last line. A three dimensional representation of the cytometries corresponding to the group labelled as 3 can be obtained with the following code.

```
rgl::plot3d(rgl::ellipse3d(templates.optimalFlow$database.elliptical[[3]][[1]]$
            cov[c(4,3,9),][,c(4,3,9)],
            centre = templates.optimalFlow$database.elliptical[[3]][[1]]$
              mean[c(4,3,9)]),
            xlim = c(0,8000), ylim =c(0,8000), zlim = c(0,8000), alpha = 0.5,
            col = 1, xlab = names(Cytometry1)[4], ylab = names(Cytometry1)[3],
            zlab = names(Cytometry1)[9])
for (j in 2:4){
  rgl::plot3d(rgl::ellipse3d(templates.optimalFlow$
              database.elliptical[[3]][[j]]$cov[c(4,3,9),][,c(4,3,9)],
              centre = templates.optimalFlow$database.elliptical[[3]][[j]]$
              mean[c(4,3,9)]), alpha = 0.5, add = T, col = j)
}
for (i in c(4:6,12)){
  for (j in 1:4){
    rgl::plot3d(rgl::ellipse3d(templates.optimalFlow$
                database.elliptical[[i]][[j]]$cov[c(4,3,9),][,c(4,3,9)],
                centre = templates.optimalFlow$database.elliptical[[i]][[j]]$
                  mean[c(4,3,9)]), alpha = 0.5, add = T, col = j)
  }
}
```

This gives us the plot represented on the left of Figure 2.4. This is due to the fact that the default mode for *optimalFlowTemplates* is to use pooling. The artificial cytometry corresponding to the group of cytometries is plotted with the following code.

```
rgl::plot3d(rgl::ellipse3d(templates.optimalFlow$templates[[3]][[1]]$
            cov[c(4,3,9),][,c(4,3,9)],
            centre = templates.optimalFlow$templates[[3]][[1]]$
              mean[c(4,3,9)]), xlim = c(0,8000), ylim =c(0,8000),
            zlim = c(0,8000), lpha = 0.5, col = 1,
            xlab = names(Cytometry1)[4], ylab = names(Cytometry1)[3],
            zlab = names(Cytometry1)[9])
for (j in 2:4){
  rgl::plot3d(rgl::ellipse3d(templates.optimalFlow$templates[[3]][[j]]$
              cov[c(4,3,9),][,c(4,3,9)],
```

```
              centre = templates.optimalFlow$templates[[3]][[j]]$
                mean[c(4,3,9)]), alpha = 0.5, add = T, col = j)
```

Hence, we obtain the image on the right of Figure 2.4.

With the next lines we show how to use the other two methods for consensus clustering.

```
templates.optimalFlow.barycenter =
  optimalFlowTemplates(
    database = database, templates.number = 5, consensus.method =
    "k-barycenter", barycenters.number = 4, bar.repetitions = 10,
    alpha.bar = 0.05, cl.paral = 1
    )
templates.optimalFlow.hdbscan =
  optimalFlowTemplates(
    database = database, templates.number = 5, consensus.method =
    "hierarchical", cl.paral = 1
    )
templates.optimalFlow.barycenter$clustering
[1] 1 2 3 3 3 3 4 4 4 1 2 3 5 5 5
templates.optimalFlow.hdbscan$clustering
[1] 1 2 3 3 3 3 4 4 4 1 2 3 5 5 5
```

We see that the clusterings of the cytometries are the same since we are using the same default method, complete linkage hierarchical clustering looking for the number of clusters indicated by templates.number. However, the situation for obtaining a consensus representative for the cytometries in group 3 is now very different. We can plot this situation as follows.

```
rgl::plot3d(rgl::ellipse3d(templates.optimalFlow$database.elliptical[[3]][[1]]$
            cov[c(4,3,9),][,c(4,3,9)],
            centre = templates.optimalFlow$database.elliptical[[3]][[1]]$
            mean[c(4,3,9)]),
            xlim = c(0,8000), ylim =c(0,8000), zlim = c(0,8000), alpha = 0.5,
            col = 1, xlab = names(Cytometry1)[4], ylab = names(Cytometry1)[3],
            zlab = names(Cytometry1)[9])
for (j in 2:4){
  rgl::plot3d(rgl::ellipse3d(templates.optimalFlow$
              database.elliptical[[3]][[j]]$cov[c(4,3,9),][,c(4,3,9)],
              centre = templates.optimalFlow$database.elliptical[[3]][[j]]$
                mean[c(4,3,9)]), alpha = 0.5, add = T, col = 1)
}
for (i in c(4:6,12)){
  for (j in 1:4){
    rgl::plot3d(rgl::ellipse3d(templates.optimalFlow$
                database.elliptical[[i]][[j]]$cov[c(4,3,9),][,c(4,3,9)],
                centre = templates.optimalFlow$database.elliptical[[i]][[j]]$
                  mean[c(4,3,9)]), alpha = 0.5, add = T, col = 1)
  }
}
```

This is what is represented on the left of both Figure 2.6 and 2.5.

We can plot the consensus element given by a 4-barycenter, obtaining the plot on the left of Figure 2.6, using the lines

```
rgl::plot3d(rgl::ellipse3d(templates.optimalFlow.barycenter$templates[[3]][[1]]$
            cov[c(4,3,9),][,c(4,3,9)],
            centre = templates.optimalFlow.barycenter$templates[[3]][[1]]$
            mean[c(4,3,9)]), xlim = c(0,8000), ylim =c(0,8000), zlim = c(0,8000),
            alpha = 0.5, col = 1, xlab = names(Cytometry1)[4], ylab =
            names(Cytometry1)[3], zlab = names(Cytometry1)[9])
for (j in 2:4){
  rgl::plot3d(rgl::ellipse3d(templates.optimalFlow.barycenter$
              templates[[3]][[j]]$cov[c(4,3,9),][,c(4,3,9)],
              centre = templates.optimalFlow.barycenter$templates[[3]][[j]]$
                mean[c(4,3,9)]), alpha = 0.5, add = T, col = j)
}
```

The plot on the right of Figure 2.5 is obtained by

```
rgl::plot3d(rgl::ellipse3d(templates.optimalFlow.hdbscan$templates[[3]][[1]]$
            cov[c(4,3,9),][,c(4,3,9)],
            centre = templates.optimalFlow.hdbscan$templates[[3]][[1]]$
            mean[c(4,3,9)]), xlim = c(0,8000), ylim =c(0,8000), zlim = c(0,8000),
            alpha = 0.5, col = 1, xlab = names(Cytometry1)[4], ylab =
            names(Cytometry1)[3], zlab = names(Cytometry1)[9])
for (j in 2:4){
  rgl::plot3d(rgl::ellipse3d(templates.optimalFlow.hdbscan$
              templates[[3]][[j]]$cov[c(4,3,9),][,c(4,3,9)],
              centre = templates.optimalFlow.hdbscan$templates[[3]][[j]]$
              mean[c(4,3,9)]), alpha = 0.5, add = T, col = j)
}
```

Wit the previous examples we have given some ideas on how to use *optimalFlowTemplates*. Now we focus on how to use our other tool, *optimalFlowClassification*.

Our objective is to classify a new cytometry, in this case it is Cytometry 1 with the corresponding four cell types, but without using the label information. We achieve this by using the lines

```
classification.optimalFlow =
  optimalFlowClassification(
    Cytometry1[which(match(Cytometry1$`Population ID (name)`,
    c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-"), nomatch = 0) > 0),
    1:10],
    database, templates.optimalFlow, consensus.method = "pooling", cl.paral = 1
  )
```

The default method of *optimalFlowClassification* is to classify using QDA and the template closest to the cytometry. In this case it is the artificial cytometry corresponding to the group labelled as 1. A median F-measure, given by (2.5), can be calculated as follows

```
scoreF1.optimalFlow =
  optimalFlow::f1Score(
  classification.optimalFlow$cluster,
  Cytometry1[which(match(Cytometry1$`Population ID (name)`,
  c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-"), nomatch = 0)>0),],
  noise.types
  )
print(scoreF1.optimalFlow)
           CD4+CD8- Monocytes Mature SIg Kappa    TCRgd-
F1-score  0.9981778 0.9998445                1 0.8913043
Precision 1.0000000 1.0000000                1 0.8039216
Recall    0.9963622 0.9996890                1 1.0000000
```

We see that results are satisfying since values are close to 1 in the first row (named F1-score) that gives de median F-score.

We get even better results by using the templates obtained via the 4-barycenter. In order to do this, we use the lines

```
classification.optimalFlow.barycenter =
  optimalFlowClassification(
    Cytometry1[which(match(Cytometry1$`Population ID (name)`,
    c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-"), nomatch = 0) > 0),
    1:10],database,
    templates.optimalFlow.barycenter, consensus.method = "k-barycenter",
    cl.paral = 1
  )
scoreF1.optimalFlow.barycenter =
   f1ScoreVoting(
     classification.optimalFlow.barycenter$cluster.vote,
     classification.optimalFlow.barycenter$cluster,
     Cytometry1[which(match(Cytometry1$`Population ID (name)`,
     c("Monocytes", "CD4+CD8-", "Mature SIg Kappa", "TCRgd-"), nomatch = 0)>0),],
     1.01, noise.types
   )
print(scoreF1.optimalFlow.barycenter$F1_score)
            TCRgd- Mature SIg Kappa Monocytes  CD4+CD8-
F1-score  0.9318182                1 0.9997667 0.9990246
Precision 0.8723404                1 1.0000000 1.0000000
Recall    1.0000000                1 0.9995334 0.9980512
```

Again, results are quite close to 1 in the first row, so the result of the supervised classification can be considered satisfying. We remind the reader that additional examples can be found in the Vignette corresponding to the package *optimalFlow*.

3

Attraction-Repulsion clustering for fairness

3.1 Introduction

In the process of machine learning there are some instances where bias and unfairness can appear. First, collecting data can be problematic, since data can reflect past unfair or biased decisions or reflect some social bias present in the real world. Therefore we could be dealing with a biased sample. Second, algorithms are developed by humans and may reflect the creators viewpoint in a subtle fashion. This may be seen in the underlying assumptions made during the development fase or in the selection of the objective functions to be optimized. A more detailed explanation of sources of bias in machine learning can be found in [Tolan, 2018].

In recent years, fair learning, a new field trying to address fairness in machine learning, has emerged. The aim is to ensure that some variables which should not be taken into account due to moral or legal issues are not playing a role in the decisions produced by the machine learning algorithms. This has naturally led to some definitions of fairness and how to try to enforce it.

When dealing with fair classification several fairness definitions have appeared in the literature. In general, in the field of fair learning data present a special form. There are some unprotected attributes X, which are the main information we want to use in the decision making. There are some protected attributes, S, whose influence in the decisions made by the algorithms we want to mitigate or eliminate. In particular, when classifying, for simplicity of exposition we consider binary classification, we have a decision rule (a classifier) $f : X \to \{0, 1\}$ and we usually have a record of the true decisions $Y \in \{0, 1\}$. Again for simplicity, let us consider $S \in \{0, 1\}$, that is the protected variable is also binary.

In this set up a possible definition of fairness known as *demographic* or *statistical parity* or *disparate impact doctrine* is to ensure that $P(f(X) = 1|S = 0) = P(f(X) = 1|S = 1)$ or equivalently $P(f(X) = 1|S) = P(f(X) = 1)$. This fairness definition has been widely used, see for example [Feldman et al., 2015], and essential imposes that the classification rule is independent of the protected variables.

In the same set up, another fairness definition known as *equalized odds* looks to impose

$P(f(X) = 1|S = 0, Y = y) = P(f(X) = 1|S = 1, Y = y)$ for $y \in \{0, 1\}$. This definition was introduced in [Hardt et al., 2016] and as stated there "Unlike demographic parity, equalized odds allows $[f(X)]$ to depend on $[S]$ but only through the target variable Y. As such, the definition encourages the use of features that allow to directly predict Y , but prohibits abusing $[S]$ as a proxy for Y."

There have been different proposals on how to achieve or approximate fairness when dealing with classification. For instance some methods try to impose demographic parity using transformations of the original variable X (see [Feldman et al., 2015, del Barrio et al., 2018]). Other methods use a direct modification of the algorithms in order to obtain fairness (see [Zafar et al., 2017, Kehrenberg et al., 2018]).

However, our purpose is to work with a related but different problem, i.e., unsupervised fair learning. In this case, we have a dataset $\mathcal{DS} = \{(X_1, S_1), \ldots, (X_n, S_n)\}$ and we would like to obtain a *fair clustering* $\mathcal{C}_f = \{\mathcal{C}_i\}_{i=1}^k$ where $\mathcal{C}_i \subset \mathcal{DS}$. For that purpose we define the notion of fair clustering, following the notion of balanced clusters introduced in [Chierichetti et al., 2017]. A *fair clustering* or a *fair partition* fulfils that

$$\frac{|\{(x, s) \in \mathcal{C}_i : s = j\}|}{|\mathcal{C}_i|} = \frac{|\{(x, s) \in \mathcal{DS} : s = j\}|}{|\mathcal{DS}|} \quad \text{for every} \quad j \in S \quad \text{and} \quad i = 1, \ldots, k. \tag{3.1}$$

In essence, it is the notion of demographic parity adapted to the clustering setting. In this case the proportion of individuals in any cluster for any protected class is the same as the respective proportion in the total data, hence we could say that proportions are independent of the particular cluster. This means that any decision taken with respect to a particular cluster will affect individuals in the same proportion as if it was taken for the entire population. Therefore disparate impact for some subpopulation would be avoided.

This definition brings with it some natural ways of enforcing it. The most straightforward approach was presented in [Chierichetti et al., 2017] and has led to many useful extensions as we have mentioned in the Introduction. Their approach is to impose the constraints defined in (3.1) by cleverly reformulating the constrained clustering problem as a minimum cost flow problem. However, our experiments shown in Section 3.6.2 suggest that imposing constraints on the proportions of the members in a cluster may be a rather strong requirement.

Our proposal takes a different approach avoiding constraints on the proportions in each cluster and instead looks for an appropriate transformation of the data, some how in the spirit of methods introduced for the supervised problem. Our transformation could be consider as a gerrymandering that looks to increase heterogeneity in the protected attributes in each cluster. The main objects of our methods are what we call *attraction-repulsion dissimilarities*, which are perturbations of the distances (or dissimilarities), based on the protected attributes, of the original points in the space of unprotected variables. Our approach allows for the use of any clustering procedure in relatively straightforward fashion.

The chaper falls into the following parts. Section 3.2 presents the *attraction-repulsion* dissimilarities. Clustering methods are developed in Section 3.3 while Section 3.4 is devoted to their extension to the kernel case. We study the importance of the choice of parameters in Section 3.5. Finally, applications for our technique are given in Section 3.6. We provide a thorough discussion on a synthetic dataset in 3.6.1, while in **??** we apply our methods to the Ricci dataset which describes the case of Ricci v. DeStefano of the Supreme Court of the United States, [Supreme Court of the United States, 2009].

In Section 3.6.2, we provide comparison between our methods and the ones proposed in [Chierichetti et al., 2017], that suggests that proportion constraints are too strong requirements if we want to retain structural information of the data.

3.2 Charged clustering via multidimensional scaling

Clustering relies on the choice of dissimilarities that control the part of information conveyed by the data that will be used to gather points into the same cluster, expressing how such points share some common characteristics. To obtain a fair clustering we aim at obtaining clusters which are not governed by the protected variables but are rather mixed with respect to these variables. For this, we introduce interpretable dissimilarities in the space $(X, S) \in \mathbb{R}^{d+p}$ aiming at separating points with the same value of the protected classes. Using an analogy with electromagnetism, the labels S play the role of an electric charge and similar charges tend to have a repulsive effect while dissimilar charges tend to attract themselves.

Our guidances for choosing these dissimilarities are that we would like the dissimilarities to

i) induce fairness into subsequent clustering techniques (eliminate or, at least, decrease dependence of the clusters on the protected attribute),

ii) keep the essential geometry of the data (with respect to non-protected attributes) and

iii) be easy to use and interpret.

Hence we propose the following dissimilarities.

Definition 3.1 (Attraction-Repulsion Dissimilarities)**.**

$$\delta_1\left((X_1, S_1), (X_2, S_2)\right) = 1'U1 + S_1'VS_2 + \|X_1 - X_2\|^2 \tag{3.2}$$

with U, V symmetric matrices in $\mathbb{R}^{p\times p}$;

$$\delta_2\left((X_1, S_1), (X_2, S_2)\right) = \left(1 + ue^{-v\|S_1 - S_2\|^2}\right)\|X_1 - X_2\|^2 \tag{3.3}$$

with $u, v \geq 0$;

$$\delta_3\left((X_1, S_1), (X_2, S_2)\right) = \|X_1 - X_2\|^2 - u\|S_1 - S_2\|^2 \tag{3.4}$$

with $u \geq 0$.
Let $0 \leq u \leq 1$ and $v, w \geq 0$,

$$\delta_4\left((X_1, S_1), (X_2, S_2)\right) = \left(1 + \operatorname{sign}(S_1'VS_2)u\left(1 - e^{-v(S_1'VS_2)^2}\right)e^{-w\|X_1 - X_2\|}\right)\|X_1 - X_2\|. \tag{3.5}$$

Remark 3.1. *$\delta_1((X, S), (X, S)) \neq 0$ and therefore it is not strictly a dissimilarity. Yet, for all practical purposes discussed in this chapter this does not affect the proposed procedures.*

To the best of our knowledge this is the first time that such dissimilarities have been proposed and used in the context of clustering (in [Ferraro and Giordani, 2013] repulsion was introduced modifying the objective function, only taking into account distances between points, to maintain centers of clusters separated). Dissimilarities (3.2) to (3.5) are natural in the context of fair clustering because they penalize the Euclidean distance taking into account the (protected) class of the points involved. Hence, some gains in fairness could be obtained.

The dissimilarities we consider are easily interpretable, providing the practitioner with the ability to understand and control the degree of perturbation introduced. Dissimilarity (3.2) is an additive perturbation of the squared Euclidean distance where the intensity of the penalization is controlled by matrices U and V, with V controlling the interactions between elements of the same and of different classes S. Dissimilarity (3.4) presents another additive perturbation but the penalization is proportional to the difference between the classes S_1 and S_2, and the intensity is controlled by the parameter u.

Dissimilarity (3.3) is a multiplicative perturbation of the squared Euclidean distance. With u we control the amount of maximum perturbation achievable, while with v we modulate how fast we diverge from this maximum perturbation when S_1 is different to S_2.

Dissimilarity (3.5) is also a multiplicative perturbation of the Euclidean distance. However, it has a very different behaviour with respect to (3.2)-(3.4). It is local, i.e., it affects less points that are further apart. Through w we control locality. With bigger w the perturbation is meaningful only for points that are closer together. With matrix V we control interactions between classes as in (3.2), while with u we control the amount of maximum perturbation as in (3.3). Again, v is a parameter controlling how fast we diverge from the maximum perturbation.

We present in the following a simple example for the case of a single binary protected attribute, coded as -1 or 1. This is an archetypical situation in which there is a population with an (often disadvantaged) minority, that we code as $S = -1$, and the new clustering has to be independent (or not too dependent) on S.

Example 3.1. *Let us take $S_1, S_2 \in \{-1, 1\}$. For dissimilarity (3.2) we fix $U = V = c \geq 0$, therefore*

$$\delta_1\left((X_1, S_1), (X_2, S_2)\right) = c(1 + S_1 S_2) + \|X_1 - X_2\|^2.$$

If $S_1 \neq S_2$, we have the usual squared distance $\|X_1 - X_2\|^2$, while when $S_1 = S_2$ we have $2c + \|X_1 - X_2\|^2$, effectively we have introduced a repulsion between elements with the same class. For dissimilarity (3.3) let us fix $u = 0.1$ and $v = 100$,

$$\delta_2\left((X_1, S_1), (X_2, S_2)\right) = \left(1 + 0.1e^{-100\|S_1 - S_2\|^2}\right)\|X_1 - X_2\|^2.$$

When $S_1 \neq S_2$ we have approximately $\|X_1 - X_2\|^2$, while when $S_1 = S_2$ we have $1.1\|X_1 - X_2\|^2$, again introducing a repulsion between elements of the same class. For dissimilarity (3.4), when $S_1 = S_2$ we have $\|X_1 - X_2\|^2$ and when $S_1 \neq S_2$ we get $\|X_1 - X_2\|^2 - 2u$, therefore we have introduced an attraction between different members of the sensitive class. When using dissimilarity (3.5), fixing $V = c > 0$, $u = 0.1$, $v = 100$, $w = 1$, we get

$$\delta_4\left((X_1, S_1), (X_2, S_2)\right) = \left(1 + 0.1\text{sign}(cS_1'S_2)\left(1 - e^{-100(cS_1'S_2)^2}\right)e^{-\|X_1 - X_2\|}\right)\|X_1 - X_2\|.$$

If $S_1 = S_2$ we get approximately $\left(1 + 0.1e^{-\|X_1 - X_2\|}\right) \|X_1 - X_2\|$, therefore we have a repulsion. If $S_1 \neq S_2$ we have approximately $\left(1 - 0.1e^{-\|X_1 - X_2\|}\right) \|X_1 - X_2\|$, which can be seen as an attraction.

Our proposals are flexible thanks to the freedom in choosing the class labels. If we codify S with $\{-1, 1\}$, as in the previous example, we can only produce attraction between different classes and repulsion between the same classes (or exactly the opposite if $V < 0$) in (3.2) and (3.5). On the other hand, if we codify S as $\{(1,0),(0,1)\}$, we have a wider range of possible interactions induced by V. For example taking $V = ((1,-1)'|(-1,0)')$ we produce attraction between different classes, no interaction between elements labelled as $(0,1)$ and repulsion between elements labelled as $(1,0)$. If we had three classes we could use $\{(1,0,0),(0,1,0),(0,0,1)\}$ as labels and induce a personalized interaction between the different elements via a 3×3 matrix V. For example $V = ((0,-1,-1)'|(-1,0,-1)'|(-1,-1,0)')$ provides attraction between different classes and no interaction between elements of the same class. Extensions to more than three classes are straightforward. More details on parameter and class codification selection will be given in Section 3.5.

These dissimilarities can then be used directly in some agglomerative hierarchical clustering methods, as described in Section 3.3. Alternatively, we could use these dissimilarities to produce some embedding of the data into a suitable Euclidean space and use some optimization clustering technique (in the sense described in Chapter 5 in [Everitt et al., 2011]) on the embedded data. Actually, the dissimilarities δ_l can be combined with common optimization clustering techniques, such as k-means, via some embedding of the data. We note that our dissimilarities aim at increasing the separation of points with equal values in the protected attributes while respecting otherwise the geometry of the data. Using multidimensional scaling (MDS) we can embed the original points in the space $\mathbb{R}^{d'}$ with $d' \leq d$ and use any clustering technique on the embedded data. Quoting [Cox and Cox, 2000], multidimensional scaling 'is the search for a low dimensional space, usually Euclidean, in which points in the space represent the objects, one point representing one object, and such that the distances between the points in the space, match, as well as possible, the original dissimilarities'. Thus, applied to dissimilarities δ_l, MDS will lead to a representation of the original data that conveys the original geometry of the data in the unprotected attributes and, at the same time, favours clusters with diverse values in the protected attributes.

Here is an outline of how to use the dissimilarities δ_l coupled with MDS for a sample $(X_1, S_1), \ldots, (X_n, S_n)$.

Attraction-Repulsion MDS For any $l \in \{1, 2, 3, 4\}$

- *Compute the dissimilarity matrix $[\Delta_{i,j}] = [\delta_l((X_i, S_i), (X_j, S_j))]$ with a particular choice of the free parameters.*
- *If $\min \Delta_{i,j} \leq 0$, transform the original dissimilarity to have positive entries: $\Delta_{i,j} = \Delta_{i,j} + |\min \Delta| + \epsilon$, where ϵ is small.*
- *For $\delta_1, \delta_2, \delta_3$: $\Delta_{i,j} = \sqrt{\Delta_{i,j}}$.*
- *Use MDS to transform $(X_1, S_1), \ldots, (X_n, S_n)$ into $X'_1, \ldots, X'_n \in \mathbb{R}^{d'}$, where $D_{i,j} = \|X'_i - X'_j\|$ is similar to $\Delta_{i,j}$.*

- *Apply a clustering procedure on the transformed data* $X'_1, \ldots, X'_n$.

This procedure will be studied in Section 3.6 for some synthetic and real datasets.

3.3 Charged hierarchical clustering

Agglomerative hierarchical clustering methods (bottom-top clustering) encompass many of the most widely used methods in unsupervised learning. Rather than a fixed number of clusters, these methods produce a hierarchy of clusterings starting from the bottom level, at which each sample point constitutes a group, to the top of the hierarchy, where all the sample points are grouped into a sigle unit. We refer to [Murtagh and Contreras, 2011] for an overview. The main idea is simple. At each level, the two groups with the lowest dissimilarity are merged to form a single group. The starting point is typically a matrix of dissimilarities between pairs of data points. Hence, the core of a particular agglomerative hierarchical clustering lies at the way in which dissimilarities between groups are measured. Classical choices include single linkage, complete linkage, average linkage or McQuitt's method. Additionally, some other methods are readily available for using dissimilarities, as, for example, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) introduced in [Ester et al., 1996].

When a full data matrix (rather than a dissimilarity matrix) is available it is possible to use a kind of agglomerative hierarchical clustering in which every cluster has an associated prototype (a center or centroid) and dissimilarity between clusters is measured through dissimilarity between the prototypes. A popular choice (see [Everitt et al., 2011]) is *Ward's minimum variance clustering*: dissimilarities between clusters are measured through a weighted squared Euclidean distance between mean vectors within each cluster. More precisely, if clusters i and j have n_i and n_j elements and mean vectors g_i and g_j then Ward's dissimilarity between clusters i and j is

$$\delta_W(i,j) = \tfrac{n_i n_j}{n_i+n_j}\|g_i - g_j\|^2,$$

where $\|\cdot\|$ denotes the usual Euclidean norm. Other methods based on prototypes are the centroid method or Gower's median method (see [Murtagh and Contreras, 2011]). However, these last two methods may present some undesirable features (the related dendrograms may present *reversals* that make the interpretation harder, see, e.g., [Everitt et al., 2011]) and Ward's method is the most frequently used within this prototype-based class of agglomerative hierarchical clustering methods.

Hence, in our approach to fair clustering we will focus on Ward's method. Given two clusters i and j consisting of points $\{(X_i, S_i)\}_{i=1}^{n_i}$ and $\{(Y_j, T_j)\}_{j=1}^{n_j}$, respectively, we define the charged dissimilarity between them as

$$\delta_{W,l}(i,j) = \tfrac{n_i n_j}{n_i+n_j}\delta_l((\tfrac{1}{n_i}\textstyle\sum_{i=1}^{n_i} X_i, \tfrac{1}{n_i}\sum_{i=1}^{n_i} S_i), (\tfrac{1}{n_j}\sum_{j=1}^{n_j} Y_j, \tfrac{1}{n_j}\sum_{j=1}^{n_j} T_j)) \qquad (3.6)$$

where δ_l, $l = 1, \ldots, 4$ is any of the point dissimilarities defined by (3.2) to (3.5).

The practical implementation of agglomerative hierarchical methods depends on the availability of efficient methods for the computation of dissimilarities between merged clusters. This is the case of the family of Lance-Williams methods (see [Lance and Williams,

1967], [Murtagh and Contreras, 2011] or [Everitt et al., 2011]) for which a recursive formula allows to update the dissimilarities when clusters i and j are merged into cluster $i\cup j$ in terms of the dissimilarities of the initial clusters. This allows to implement the related methods using computer time of order $O(n^2\log n)$. We show next that a recursive formula similar to the Lance-Williams class holds for the dissimilarities $\delta_{l,W}$ and, consequently, the related agglomerative hierarchical method can be efficiently implemented. The fact that we are dealing differently with genuine and protected attributes results in the need for some additional notation (and storage). Given clusters i and j consisting of points $\{(X_i,S_i)\}_{i=1}^{n_i}$ and $\{(Y_j,T_j)\}_{j=1}^{n_j}$, respectively, we denote

$$d_x(i,j)=\big\|\tfrac{1}{n_i}\textstyle\sum_{i=1}^{n_i}X_i-\tfrac{1}{n_j}\sum_{j=1}^{n_j}Y_j\big\|. \tag{3.7}$$

Note that $d_x(i,j)$ is simply the Euclidean distance between the means of the X-attributes in clusters i and j. Similarly, we set

$$d_s(i,j)=\big\|\tfrac{1}{n_i}\textstyle\sum_{i=1}^{n_i}S_i-\tfrac{1}{n_j}\sum_{j=1}^{n_j}T_j\big\|. \tag{3.8}$$

Proposition 3.1. *For $\delta_{W,l}$ as in (3.6), $d_x(i,j)$ as in (3.7) and $d_s(i,j)$ as in (3.8) and assuming that clusters i,j and k have sizes n_i,n_j and n_k, respectively, we have the following recursive formulas:*

i) $\delta_{W,1}(i\cup j,k)=\frac{n_i+n_k}{n_i+n_j+n_k}\delta_{W,1}(i,k)+\frac{n_j+n_k}{n_i+n_j+n_k}\delta_{W,1}(j,k)-\frac{n_k}{n_i+n_j+n_k}d^2_{W,x}(i,j)$;

ii)

$$\delta_{W,2}(i\cup j,k)=\Big(1+ue^{-v(\frac{n_i}{n_i+n_j}d_s^2(i,k)+\frac{n_j}{n_i+n_j}d_s^2(j,k)-\frac{n_in_j}{(n_i+n_j)^2}d_s^2(i,j))}\Big)\times\Big(\tfrac{n_i+n_k}{n_i+n_j+n_k}d^2_{W,x}(i,k)+\tfrac{n_j+n_k}{n_i+n_j+n_k}d^2_{W,x}(j,k)-\tfrac{n_k}{n_i+n_j+n_k}d^2_{W,x}(i,j)\Big);$$

iii) $\delta_{W,3}(i\cup j,k)=\frac{n_i+n_k}{n_i+n_j+n_k}\delta_{W,3}(i,k)+\frac{n_j+n_k}{n_i+n_j+n_k}\delta_{W,3}(j,k)-\frac{n_k}{n_i+n_j+n_k}\delta_{W,3}(i,j)$,

where $d^2_{W,x}(i,j)=\frac{n_in_j}{n_i+n_j}d_x^2(i,j)$.

Proof. For *i)* we just denote by R_s,S_t and T_r the protected attributes in clusters i,j and k, respectively and note that

$$\begin{aligned}
\delta_{W,1}(i\cup j,k)&=\tfrac{(n_i+n_j)n_k}{n_i+n_j+n_k}\Big(1'U1+\tfrac{1}{n_i+n_j}\big(\textstyle\sum_{s=1}^{n_i}R_s+\sum_{t=1}^{n_j}S_t\big)'V\frac{1}{n_k}\sum_{r=1}^{n_k}T_r+d_x^2(i\cup j,k)\Big)\\
&=\tfrac{(n_i+n_j)n_k}{n_i+n_j+n_k}\tfrac{n_i}{n_i+n_j}\Big(1'U1+\tfrac{1}{n_i}\big(\textstyle\sum_{s=1}^{n_i}R_s\big)'V\frac{1}{n_k}\sum_{r=1}^{n_k}T_r\Big)\\
&+\tfrac{(n_i+n_j)n_k}{n_i+n_j+n_k}\tfrac{n_j}{n_i+n_j}\Big(1'U1+\tfrac{1}{n_j}\big(\textstyle\sum_{t=1}^{n_j}S_t\big)'V\frac{1}{n_k}\sum_{r=1}^{n_k}T_r\Big)+d^2_{W,x}(i\cup j,k)\\
&=\tfrac{n_i+n_k}{n_i+n_j+n_k}\tfrac{n_in_k}{n_i+n_k}\Big(1'U1+\tfrac{1}{n_i}\big(\textstyle\sum_{s=1}^{n_i}R_s\big)'V\frac{1}{n_k}\sum_{r=1}^{n_k}T_r+d_x^2(i,k)\Big)\\
&+\tfrac{n_j+n_k}{n_i+n_j+n_k}\tfrac{n_jn_k}{n_j+n_k}\Big(1'U1+\tfrac{1}{n_j}\big(\textstyle\sum_{j=1}^{n_j}S_j\big)'V\frac{1}{n_k}\sum_{r=1}^{n_k}T_r+d_x^2(j,k)\Big)\\
&-\tfrac{n_k}{n_i+n_j+n_k}d^2_{W,x}(i,j)\\
&=\tfrac{n_i+n_k}{n_i+n_j+n_k}\delta_{W,1}(i,k)+\tfrac{n_j+n_k}{n_i+n_j+n_k}\delta_{W,1}(j,k)-\tfrac{n_k}{n_i+n_j+n_k}d^2_{W,x}(i,j).
\end{aligned}$$

Observe that we have used the well-known recursion for Ward's dissimilarities, namely,

$$d^2_{W,x}(i \cup j, k) = \tfrac{n_i+n_k}{n_i+n_j+n_k} d^2_{W,x}(i,k) + \tfrac{n_j+n_k}{n_i+n_j+n_k} d^2_{W,x}(j,k) - \tfrac{n_k}{n_i+n_j+n_k} d^2_{W,x}(i,j) \tag{3.9}$$

(see, e.g., [Everitt et al., 2011]). The update formulas *ii)* and *iii)* are obtained similarly. We omit details.

□

From Proposition 3.1 we see that a practical implementation of agglomerative hierarchical clustering based on $\delta_{W,l}$, $l = 1, 2$ would require the computation of $d^2_{W,x}(i,j)$, which can be done using the Lance-Williams formula (3.9). In the case of $\delta_{W,2}$ we also need $d^2_s(i,j)$, which again can be obtained through a Lance-Williams recursion. This implies that agglomerative hierarchical clustering based on $\delta_{W,l}$, $l = 1, 2$ or 3 can be implemented using computer time of order $O(n^2 \log n)$ (at most twice the required time for the implementation of an 'unfair' Lance-Williams method).

We end this section with an outline of the implementation details for our proposal for fair agglomerative hierarchical clustering based on dissimilarities $\delta_{W,l}$.

Iterative Attraction-Repulsion Clustering For $l \in \{1, 2, 3\}$

- *Compute the dissimilarity matrix* $[\Delta_{i,j}] = [\delta_l((X_i, S_i), (X_j, S_j))]$ *with a particular choice of the free parameters.*
- *If* $\min \Delta_{i,j} \leq 0$, *transform the original dissimilarity to have positive entries:* $\Delta_{i,j} = \Delta_{i,j} + |\min\Delta| + \epsilon$, *where* ϵ *is arbitrarily small.*
- *Use the Lance-Williams type recursion to determine the clusters* i *and* j *to be merged; iterate until there is a single cluster*

3.4 Fair clustering with kernels

Clustering techniques based on the minimization of a criterion function typically result in clusters with a particular geometrical shape. For instance, given a collection of points $x_1, \ldots x_n \in \mathbb{R}^d$, the classical k-means algorithm looks for a grouping of the data into $K \leq n$ clusters $C = \{c_1, \ldots, c_K\}$ with corresponding means $\{\mu_1, \ldots, \mu_K\}$ such that the objective function

$$\sum_{k=1}^{K} \sum_{x \in c_i} \|x - \mu_i\|^2$$

is minimized. The clusters are then defined by assigning each point to the closest center (one of the minimizing c_i's). This results in convex clusters with linear boundaries. It is often the case that this kind of shape constraint does not adapt well to the geometry of the data. A non-linear transformation of the data could map some clustered structure to make it more adapted to convex linear boundaries (or some other pattern). In some cases this transformation can be implicitly handled via kernel methods. We explore in this section how the charged clustering similarities that we have introduced can be adapted to the kernel clustering setup, focusing on the particular choice of kernel k-means.

Kernel k-means is a non-linear extension of k means that allows to find arbitrary shaped clusters introducing a suitable kernel similarity function $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, where the role of the squared Euclidean distance between two points x, y in the classical k-means is taken by

$$d_\kappa^2(x, y) = \kappa(x, x) + \kappa(y, y) - 2\kappa(x, y). \tag{3.10}$$

Details of this algorithm can be found in [Schölkopf et al., 1998].

In a first approach, we could try to introduce a kernel function for vectors (X_1, S_1), $(X_2, S_2) \in \mathbb{R}^{d+p}$ such that d_κ^2 takes into account the squared Euclidean distance between X_1 and X_2 but also tries to separate points of the same class and/or tries to bring closer points of different classes, i.e., makes use of S_1, S_2. Some simple calculations show that this is not an easy task, if possible at all in general. If we try, for instance, a joint kernel of type $\kappa((X_1, S_1), (X_2, S_2)) = \tau(S_1, S_2) + k(X_1, X_2)$, $S_1, S_2 \in \{-1, 1\}$ with τ, k Mercer (positive semi-definite) kernels (this covers the case $k(X_1, X_2) = X_1 \cdot X_2$, the usual scalar product in $\mathbb{R}^d$), our goal can be written as

$$d_\kappa^2((X_1, S_1), (X_2, S_1)) > d_\kappa^2((X_1, S_1), (X_2, S_2)), \tag{3.11}$$

for any X_1, X_2, with $S_1 \neq S_2$. However, the positivity contraints on τ, imply that

$$2\tau(S_1, S_2) > \tau(S_1, S_1) + \tau(S_2, S_2),\ \tau^2(S_1, S_2) \leq \tau(S_1, S_1)\tau(S_2, S_2).$$

But the solutions of this inequalities violate that τ is positive-semi-definite. Therefore, there is no kernel on the sensitive variables that we can add to the usual scalar product. Another possibility is to consider a multiplicative kernel, $\kappa((X_1, S_1), (X_2, S_2)) = \tau(S_1, S_2)k(X_1, X_2)$, $S_1, S_2 \in \{-1, 1\}$ with τ, k Mercer kernels. From (3.11) we get

$$2\left(\tau(S_1, S_1) - \tau(S_1, S_2)\right) k(X_1, X_2) < \left(\tau(S_1, S_1) - \tau(S_2, S_2)\right) k(X_2, X_2)$$

which depends on $k(X_1, X_2)$ and makes it challenging to eliminate the dependence of the particular combinations X_1, X_2.

The previous observations show that it is difficult to think of a simple and interpretable kernel κ that can be a simple combination of a kernel in the space of unprotected attributes and a kernel in the space of sensitive attributes. This seems to be caused by our desire to separate vectors that are similar in the sensitive space, which goes against our aim to use norms induced by scalar products. In other words a naive extension of the kernel trick to our approach to fair clustering seems to be inappropriate.

Nonetheless, the difficulty comes from a naive desire to carry out the (implicit) transformation of the attributes and the penalization of homogeneity in the protected attributes in the clusters in a single step. We still may obtain gains in fairness, while improving the separation of the clusters in the unprotected attributes if we embed the X data into a more suitable space by virtue of some sensible kernel κ and consider the corresponding kernel version of δ_l, with δ_l as in (3.2) to (3.5). Instead of using the Euclidean norm $\|X_1 - X_2\|$ we should use $d_\kappa(X_1, X_2)$. In the case of δ_1, for instance, this would amount to consider the dissimilarity

$$\delta_{\kappa,1}\left((X_1, S_1), (X_2, S_2)\right) = 1'U1 + S_1'VS_2 + d_\kappa(X_1, X_2)^2, \tag{3.12}$$

with similar changes for the other dissimilarities. Then we can use an embedding (MDS the simplest choice) as in Section 3.2 and apply a clustering procedure to the embedded

data. This would keep the improvement in cluster separation induced (hopefully) by the kernel trick and apply, at the same time, a fairness correction. An example of this adaptation of the kernel trick to our setting is given in Section 3.6.1.

3.5 Parameter selection

The attraction-repulsion dissimilarities (3.2) - (3.5) introduced in Section 3.2 depend on two main sets of parameters. First, on several free parameters used to balance the influence of the variables X and the protected variable S. Second, on the way the protected variables are labelled for the different classes, with the possibility to include different interactions between the groups. Here we propose some guidelines on the choice of these parameters. A complete example of how to select the best parameters, the best dissimilarity and the best clustering method is provided in Section 3.6.3.

The dissimilarities we consider can be divided into two groups: (3.3) and (3.4) do not depend on the codification of the class variable, while (3.2) and (3.5) do depend on such a choice. In our method, the level of perturbation, which influences the level of fairness is imposed through the choice of the parameters in the dissimilarities. Contrary to other methods such as [Chierichetti et al., 2017] where only completely fair solutions can be found (see for instance in Figure 3.1), choosing the parameters enables to balance fairness and the original structure of the data which may convey information that should not be erased by fairness constraints.

Consider first dissimilarities (3.3) and (3.4). They rely on two parameters u and v. In the multiplicative dissimilarity (3.3), v is a parameter that measures how sudden is the change in the distance when switching from elements with different protected class to elements with same protected class. For v large enough, $e^{-v\|S_1-S_2\|^2}$ is small when $S_1 \neq S_2$, which implies that the fair dissimilarity only modifies the distance between points inside the same protected class, increasing heterogeneity of the clusters.

Once v has been fixed, the main parameter u controls the intensity of the perturbation in a similar way for both dissimilarities (3.3) and (3.4). To illustrate the effect of this parameter we focus on (3.3) and perform a fair clustering, with MDS or hierarchically, for different values of the intensity parameter u and measure the fairness of the clusters obtained. Such example is depicted in the left middle row of Figure 3.1. We can see that, as expected, increasing the values for u puts more weight on the part of the dissimilarity that enforces heterogeneity of the clusters. $u = 0$ leads to the usual clustering. Indeed, varying u from 0 to 4.5 in steps of 0.5 increases the fairness achieved for both clusters, with a saturation effect from 4.5 to 5 where we do not appreciate an improvement in fairness. Hence, maximum fairness is achieved for $u = 4.5$ and gives the lowest perturbation that achieves the highest level of fairness. If one aims at preserving more of the structure of the original information at the expense of a lower level of fairness, some smaller value of u can be selected. For example, in the right middle row of Figure 3.1 we provide the result of choosing $u = 1$. Hence u balances both effects of closeness to the usual dissimilarities and the amount of heterogeneity reached in the clustering.

Next, dissimilarities (3.2) and (3.5), as described in Section 3.2, depend on the values chosen for the protected variable S, and a matrix V, which plays the role of the matrix of interactions for different classes. When dealing with a two-class discrimination

problem where the protected class has only two values, labelling the classes as $\{-1, 1\}$ or $\{(1, 0), (0, 1)\}$ can lead to the same results for appropriate choices of V. However, for more than two protected classes we will use only the following vectorial codification: for q different values of the protected class, we will codify the values as the q unitary vectors $\{a_1, \ldots, a_q\}$ where $a_{i,j} = 1$ if $i = j$ and $a_{i,j} = 0$ if $i \neq j$ for $1 \leq i, j \leq q$.

To build the interaction matrix we proceed as follows. First, consider a matrix $\tilde{V}_{i,j}$ with $1 \leq i, j \leq q$. We fix $\tilde{V}_{i,j} = 0$ if we want no interaction between classes i and j, in particular, if $i = j$ this means that there is no interaction between elements with the same class. We take $\tilde{V}_{i,j} = 1$ if we want repulsion (relative increase in the distance) between classes i and j. We fix $\tilde{V}_{i,j} = -1$ if we want attraction (relative decrease in distance) between classes i and j. Hence, if the practitioner believes that there is some discrimination, in the sense of disproportional impact, against a class represented by a_{i^*}, it is recommendable to set values of $\tilde{V}_{i^*,j} = -1$ for $j \neq i^*$. As an example, in Section 3.6.1, we have chosen the interaction matrix $V = ((1, -1)'|(-1, 0)')$, to model repulsion between elements of the same class $(1, 0)$, attraction between elements of the classes $(1, 0)$ and $(0, 1)$, and no interaction between the elements of the same class $(0, 1)$.
Then intensity of the interaction is modelled using a constant $v_0 > 0$, and we set $V = v_0\tilde{V}$. In the previous example $v_0 = 1$. The parameter v for dissimilarity (3.5) has the same meaning as the corresponding parameter for (3.3) and can be selected in the same way.

Finally, matrix U for dissimilarity (3.2) represents and extra additive shift. In many cases it can be set to $U = 0$ (the zero matrix).

We provide an example to explain how to select the intensity v_0 for dissimilarity (3.2) in the top left image of Figure 3.1. Notice that using $V > 0$ and $S \in \{-1, 1\}$ is the same as using $V = v_0\tilde{V}$ with $\tilde{V} = ((1, -1)'|(-1, 1)')$ and $S \in \{(1, 0), (0, 1)\}$. We plot the variation of the fairness in each cluster when we vary the intensity of the interaction between 0 and 4.4 with steps of size 0.44. There is a steady improvement in fairness in both clusters until the intensity reaches $v_0 = 3.52$, but from this level, as previously, there is no more improvement in fairness. Therefore, if a practitioner wants to achieve the highest level of fairness, $v_0 = 3.52$ should be the proper intensity, since it corresponds to the smallest perturbation to the geometry that achieves the best observed fairness. However, a smaller correction in fairness may be of interest, we have a representation of that top right in Figure 3.1 for $v_0 = 1.32$.

For dissimilarity (3.5), after choosing the interaction matrix $\tilde{V}$, we can try to find a maximum in fairness, fixing a grid formed by different combinations for the vector of parameters (v_0, u, w). In the second and third column of Table 3.2 we see the fairness of the respective clusters when we look at the grid $(1, u, 0.05)$ with $u = 0, 0.098, \ldots, 0.98$. What we notice is an improvement in fairness for all values of u, therefore a practitioner would be advised to select $u = 0.98$ where we obtain the best fairness values.

3.6 Applications

In this section we provide examples of attraction repulsion clustering. In the first two subsections we mainly want to describe how attraction-repulsion clustering works and how it compares to some other fair clustering procedures. The last subsection is a full example on a non trivial real data set where full tuning of the parameters and selection

of the best clustering algorithms is provided.

3.6.1 Synthetic data

General example

We generate 50 points from four distributions,

$$\mu_1 \sim N((-1, 0.5), \text{diag}(0.25, 0.25)), \mu_2 \sim N((-1, -0.5), \text{diag}(0.25, 0.25));$$

$$\mu_3 \sim N((1, 0.5), \text{diag}(0.25, 0.25)), \mu_4 \sim N((1, -0.5), \text{diag}(0.25, 0.25)),$$

and label the samples from μ_1 and μ_2 as $S = 1$ (squares) and the samples from μ_3 and μ_4 as $S = -1$ (circles). A representation of the data in the original space is given in the third column of Figure 3.1. We can think of the data as heavily biased in the x direction, therefore any sensible clustering procedure is going to have clusters that are highly homogeneous in the class S when the original coordinates are used. This is exemplified in Table 3.1, as we look for different number of clusters: with k-means we are detecting almost pure groups (1st row); the same happens with a complete linkage hierarchical clustering with the Euclidean distance (5th row) and with Ward's method with the Euclidean distance (9th row).

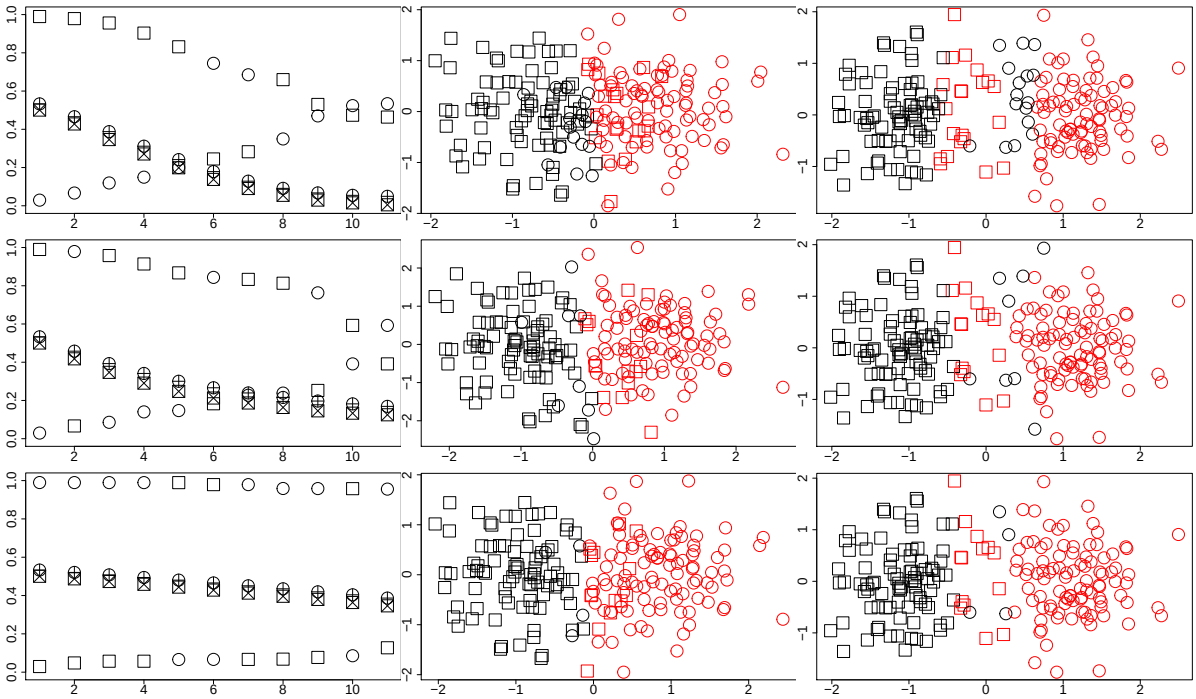
Therefore, it may be useful to apply our procedures to the data to gain diversity in S. In the first column of Figure 3.1 we study the relation between the gain in fairness from the increase in intensity of the corrections we apply and the disruption of the geometry of the original classes after MDS. In the first row we use dissimilarity (3.2), where we fix $U = 0$, and we vary $V = 0, 0.44, 0.88, \ldots, 4.4$. In the second row we work with dissimilarity (3.3), where we fix $v = 20$ and set $u = 0, 0.5, 1, \ldots, 5$. In the last row we work with dissimilarity (3.5) fixing $V = 1$, $v = 20$, $w = 1$ and we vary $u = 0, 0.099, 0.198, \ldots, 0.99$. We do not show results for dissimilarity (3.4), since in this example it gives results very similar to dissimilarity (3.2). With some abuse of notation, throughout Section 3.6 we will use S as the name of the protected variable. Squares and circles represent the proportion of class $S = 1$ in the two clusters found by k-means after the MDS transformation. Crossed squares and circles represent the average silhouette index of class $S = 1$ and class $S = -1$. We recall that the silhouette index of an observation X_i is given by

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

where $a(i)$ is the average distance to X_i of the observation in the same group as X_i, and $b(i)$ is the average distance to X_i of the observations in the closest group different than the one of X_i (see [Rousseeuw, 1987]). The average silhouette index of a group is the average of the silhouette indexes of the members of the group and the average silhouette index is the average of the silhouette indexes of all points.

What we see top-left and middle-left in Figure 3.1 is that greater intensity relates to greater heterogeneity but also relates to lower silhouette index. This can be interpreted as the fact that greater intensity in dissimilarities (3.2) and (3.3) has a greater impact in the geometry of the original problem. In essence, the greater the intensity, the more indistinguishable $S = 1$ and $S = -1$ become after MDS, therefore, any partition with k-means will result in very diverse clusters in S. By construction this is not what happens

Figure 3.1: Top row: dissimilarity (3.2). Middle row: dissimilarity (3.3). Bottom row: dissimilarity (3.5). Left column: proportions of $S = 1$ in the clusters (squares and circles) and average silhouette indexes for $S = 1$ and $S = -1$ in the transformed space (crossed squares and circles), for varying input parameters. Middle column: two clusters in the transformed space for a particular choice of parameters. Right column: same two clusters in the original space.



with dissimilarity (3.5). The strong locality penalty ($w = 1$) allows to conserve the geometry, shown by the little reduction in silhouette index (row 3 column 1), but results in smaller corrections in the proportions.

From the previous discussion, a practitioner interested in imposing fairness, with no strong requirements on the compactness of clusters in the original geometry should use high intensity corrections. However, a practitioner interested in gaining some fairness while still being able to keep most of the original geometry should go for low intensity or local corrections.

In the rest of Figure 3.1 we show the actual clusters in the MDS embedding obtained with k-means (column 2) and the same clusters in the original space (column 3), for some moderate intensities. For dissimilarity (3.2) we take $V = 1.32$, for (3.3) $u = 1$ and for (3.5) we use $u = 0.99$. A short remark is that a rotation of a MDS is a MDS, and that is the cause of the rotations that we see in column 2. Indeed, after MDS the geometry of the groups is not heavily modified, but at the same time some corrections to the proportions are achieved when clustering. This corrections appear very natural once we return to the original space.

For the same values as the previous paragraph we present Table 3.1, where we look for 2,3 and 4 clusters with MDS and k-means, but also using the approximation-free complete

Table 3.1: Proportion of class S = 1 in every group in different clustering procedures.

			Proportion of squares in the group								
			K = 2		K = 3			K = 4			
k-means	Unperturbed		0.03	0.99	0.02	0.88	0.98	0.98	1.00	0.06	0.02
	MDS	δ_1	0.15	0.90	0.11	0.49	0.95	0.75	0.94	0.16	0.14
		δ_2	0.09	0.96	0.43	0.04	0.97	0.08	0.97	0.85	0.06
		δ_4	0.13	0.96	0.96	0.05	0.42	0.96	0.11	0.13	0.95
Complete Linkage	Unperturbed		0.99	0.07	0.99	0.08	0.05	0.99	1.00	0.08	0.05
	δ_1		1.00	0.32	1.00	0.40	0.25	1.00	0.56	0.25	0.00
	δ_2		0.72	0.22	0.72	0.30	0.00	1.00	0.30	0.24	0.00
	δ_4		0.78	0.11	1.00	0.43	0.11	1.00	0.43	0.47	0.00
Ward's Method	Unperturbed		0.99	0.07	0.08	0.05	0.99	0.97	0.08	0.05	1.00
	δ_1		0.11	0.78	0.54	0.98	0.11	0.54	0.18	0.00	0.98
	δ_2		0.99	0.18	0.37	0.99	0.03	0.07	0.00	0.37	0.99

linkage hierarchical clustering and our Ward's-like method. Since we are applying a small perturbation, we see some, but not a drastic, improvement in the heterogeneity of the groups. We also see that the more clusters we want the smaller the improvement. We stress that we are able to produce some improvements in fairness while modifying slightly the geometry of the data. This is a desirable situation when a lot of relevant information is codified in the geometry of the data. We also notice that we are able to induce almost full fairness with a stronger perturbation as shown in the last row of Table 3.3.

Kernel trick example

Let us explore the adaptation of the kernel trick explained in Section 3.4. We consider the data in the top-left image of Figure 3.2. These data have a particular geometrical shape and are split into two groups. There is an inside ring of squares, a middle ring of circles, and then an outer ring of squares. There are 981 observations and the proportions of the classes are approximately 3 to 1 (circles are 0.246 of the total data).

It is natural to apply to the original data some clustering procedure as k-means or a robust extension as tclust (deals with groups with different proportions and shapes and with outliers [García-Escudero et al., 2008]). Looking for two clusters, we would be far from capturing the geometry of the groups, but the clusters would have proportions of the classes that are similar to the total proportion. Indeed, this is what we see in Figure 3.2 middle-left when we apply k-means to the original data.

On the other hand, the kernel trick is convenient in this situation. We propose to use the kernel function $\kappa(x, y) = x_1^2 y_1^2 + x_2^2 y_2^2$, which corresponds to a transformation $\phi((x_1, x_2)) = (x_1^2, x_2^2)$. This kernel indeed produces linear separation between the groups. The data in the transformed space is depicted in the top-right of Figure 3.2. Our adaptation to the kernel trick uses d_κ as defined in (3.10) and dissimilarity (3.5) in the form

$$\delta_{\kappa,4}((X_1, S_1), (X_2, S_2)) = \big(1 + \text{sign}(S_1' V S_2) u (1 - e^{-v(S_1' V S_2)^2}) e^{-w d_\kappa(X_1, X_2)}\big) d_\kappa(X_1, X_2), \tag{3.13}$$

for X_1, X_2 in the original two dimensional space, as described in Section 3.4.

Table 3.2: Effect of varying the intensity u of the local dissimilarity (3.13), for fixed $V = ((1,-1)'|(-1,0)')$, $v = 20$ and $w = 0.05$. First two columns contain the proportion of points with $S = (0,1)$ in the clusters found with tclust in the transformed space. Last two columns show the silhouette of the original classes in the MDS.

u	Prop. in cluster 1	Prop. in cluster 2	Silhouette for (0, 1)	Silhouette for (1,0)
0.000	0.629	0.950	-0.247	0.502
0.098	0.629	0.950	-0.245	0.502
0.196	0.629	0.950	-0.243	0.499
0.294	0.630	0.948	-0.241	0.495
0.392	0.631	0.945	-0.239	0.491
0.490	0.631	0.943	-0.237	0.486
0.588	0.631	0.943	-0.235	0.481
0.686	0.631	0.943	-0.234	0.476
0.784	0.630	0.946	-0.232	0.471
0.882	0.672	0.863	-0.231	0.467
0.980	0.681	0.849	-0.229	0.465

Taking into account the discussion at the end of Section 3.2 and Section 3.5 we use dissimilarity (3.13) with $S_1, S_2 \in \{(1,0),(0,1)\}$. In our setting circles are labelled as $(1,0)$ and squares as $(0,1)$. Now if we fix $u = 0$, use (3.13) to calculate the dissimilarity matrix Δ and use MDS, essentially, we will be in the space depicted top-right on Figure 3.2. Looking for two clusters with tclust, allowing groups with different sizes, we get the result depicted middle-right in Figure 3.2. We have captured the geometry of the clusters but the proportions of the class S are not the best, as seen in row 1 columns 2 and 3 of Table 3.2 (ideally they should be close to 0.754). In order to gain diversity in what is referred as cluster 2 in Table 3.2 (in red in the plots), we vary the intensity u of our local dissimilarity, with the other parameters set as indicated in Table 3.2. We see that as we increase the intensity of the interactions we gain in heterogeneity in cluster 2, and both proportions come closer to the total proportion 0.754 (columns 2-3). Again, this is achieved without destroying the geometry of the original classes after the MDS, as seen in the small variation of the average silhouette index in columns 4-5.

We plot the best performance, given by $u = 0.98$, after MDS in bottom-left and in the original space in bottom-right of Figure 3.2. It is clear that we have been able to capture the geometry of the groups and to produce relatively fair clusters.

3.6.2 Comparison with fair clustering through fairlets

In this section we present a comparison of the results of our methods with results obtained by implementing, in Python and R, the fair clustering procedure introduced in [Chierichetti et al., 2017] based on fairlets decomposition. Since our examples are concerned with two values for the protected class it is justified to use [Chierichetti et al., 2017] for comparison since it is well suited for this situation. Our implementation of the case when the size of both protected classes is the same, which reduces to an assignment problem, is implemented using the function *max_bipartite_matching* of the package *igraph* in R. In the case of different sizes, we have to solve a min cost flow problem as stated

Figure 3.2: Top row: data in the original space (left) and after transformation ϕ (right). Middle row: k-means in the original space (left) and tclust applied in the transformed space and plotted in the original one (right). Bottom row: tclust after fairnes corrections applied in the transformed space (left) and represented in the original space (right).

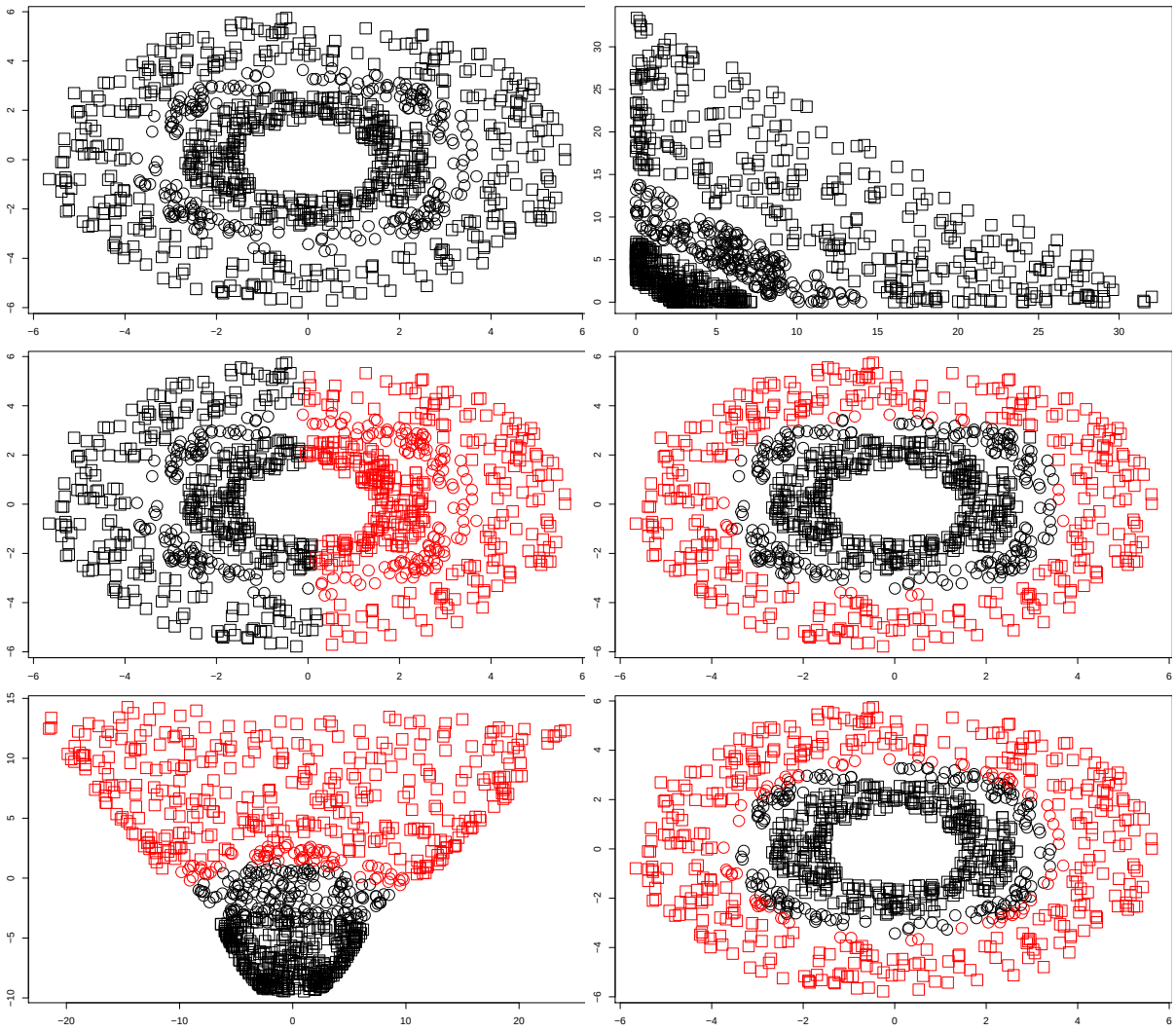


Table 3.3: Rows 1-7 are implementations of fair clustering with fairlets while the last row is k-median clustering after using δ_1, with $(U = 0, V = 4.4)$, and MDS embedding.

	balance	Average Silhouette	K-median Objective
Assignement Problem	1	-0.0043	462.5843
$(2, \infty)$	1	-0.0064	468.8754
$(2, 2.416589)$	1	-0.0010	471.2754
$(3, \infty)$	1	-0.0021	465.7702
$(3, 2.416589)$	1	-0.0035	468.3651
$(4, \infty)$	1	-0.0048	465.8894
$(4, 2.416589)$	1	-0.0009	464.3114
Attraction-repulsion	0.8929	0.2007	440.1799

in [Chierichetti et al., 2017], which can be done in Python with the function *min_cost_flow* of the package *networkx* (also it can be solved with a *min_cost_flow* solver in *ortools*).

We recall that the balance of a set X of points that are labelled black or red is defined as

$$\text{balance}(X) = \min\left(\frac{\#Black}{\#Red}, \frac{\#Red}{\#Black}\right)$$

and the balance of $\mathcal{C}$, a clustering of the data in X, is given by

$$\text{balance}(\mathcal{C}) = \min_{C \in \mathcal{C}} \text{balance}(C).$$

When implemented as a min cost flow problem, Chierichietti's et al. methodology has two free parameters, t' and τ. First, $1/t' \leq \text{balance}(\mathcal{C}) \leq \text{balance}(X)$ and hence t' controls the lower bound of the fairness of the partition. Second, τ is a free parameter that controls something similar to the locality that we have mentioned previously, and has a defined lower limit given by the maximal minimal distance between points of the two original classes.

We start with the data used for the example studied in Figure 3.1. We will address k-median clustering for which [Chierichetti et al., 2017] has a fair implementation. Since the data has two groups of the same size we can solve an assignment problem or use a min cost flow problem. For the min cost flow problem we have the set of parameters $\{(t', \tau)\}$ with $t' = 2, 3, 4$ and $\tau = \infty, 2.416589$. Values for τ represent no locality and maximum locality. As comparison we will use k-median clustering after perturbing the data with δ_1 with parameters $(U = 0, V = 4.4)$ and doing and MDS embedding. Results are shown in Table 3.3. Since the data is into one to one correspondence between the two classes, both the alignment solution and the different min cost flow solutions give balance = 1, hence total fairness is achieved. Our method gives balance close to 1, but does not achieve total fairness. However, the average silhouette index of our method is higher, which means that clusters are more identifiable and compact and the k-median objective function is also lower, and hence better. A plot of some of the different clusterings can be seen in Figure 3.3.

Our next comparison is for the data used in Figure 3.2. We stress that for fairlets we are working with the data after the k-trick transformation, i.e., the data shown top-right

Figure 3.3: Left: clusters obtained by fair k-median as an assignment problem([Chierichetti et al., 2017]). Middle: clusters obtained by fair k-median as min cost flow problem with $t' = 4, \tau = 2.416589$([Chierichetti et al., 2017]). Right: clusters obtained by attraction-repulsion clustering with dissimilarity (3.2) and MDS, using k-median.

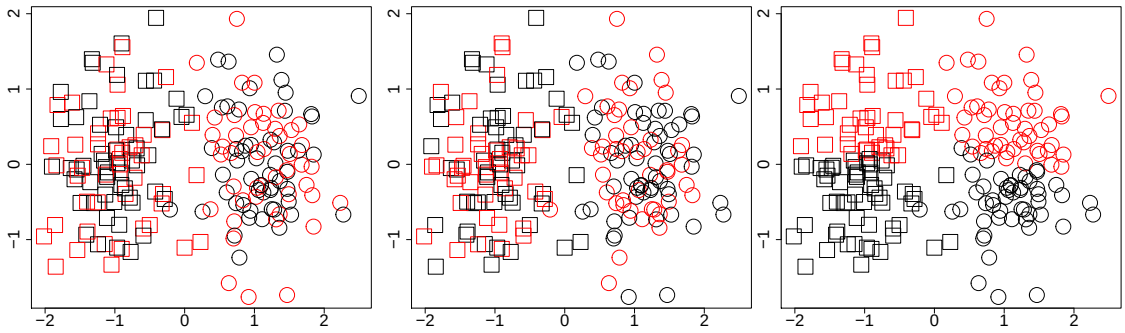
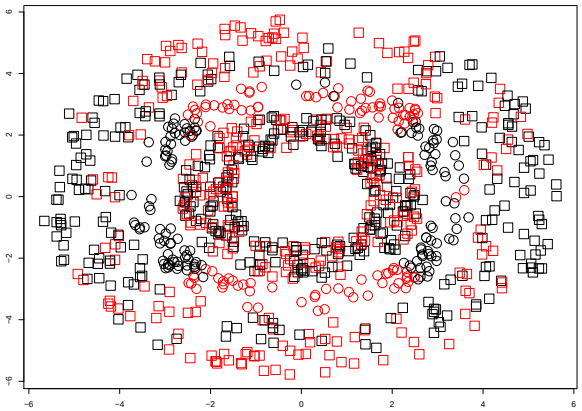


Figure 3.4: Left: clustering obtained using fair k-median as min cost flow problem with $t' = 4, \tau = 19.62385$([Chierichetti et al., 2017]).



in Figure 3.2, and therefore results are comparable with attraction-repulsion clustering in the k-trick setting. Results are shown in Table 3.4. We see that consistently the fair k-madian implementation gives balance values very close to $241/740 \approx 0.3257$, hence giving great approximation to fairness. Our method, gives a lower balance value, hence groups are more unfair, but as we see from the silhouette and k-median objective function values, the groups are more identifiable and more compact. Even more, comparing middle-right of Figure 3.2 and left of Figure 3.4, we see that our procedure is even able to capture the underlying geometry of the data.

Our last comparison is on a real data set known as the Ricci dataset, consisting of scores in an oral and a written exam of 118 firefighters, where the sensitive attribute is the race of the individual. This dataset was a part of the case Ricci v. DeStefano presented before the Supreme Court of the United States.

For applying our attraction-repulsion clustering we codify white individuals as $S = 1$

Table 3.4: Rows 1-6 are implementations of fair clustering with fairlets as a min cost flow problem with different parameters. Last row is a tclust clustering after using δ_4, with $(u = 0.98, v = 20, w = 0.05, V = ((1, -1)'|(-1, 0)')$, and MDS embedding.

	balance	Average Silhouette	K-median Objective
$(4, \infty)$	0.3163	0.0265	14613.74
$(4, 19.62385)$	0.3253	0.0685	14975.96
$(5, \infty)$	0.3220	0.0359	14819.47
$(5, 19.62385)$	0.3016	0.0636	14854.99
$(6, \infty)$	0.3049	0.0257	14683.25
$(6, 19.62385)$	0.2991	0.0511	14792.05
Attraction-repulsion	0.1890	0.3957	6751.01

Table 3.5: Rows 1-8 are implementations of fair clustering with fairlets as a min cost flow problem with different parameters. Rows 9-10 are the best results we have obtained with attraction-repulsion clustering. Last row represents the values for a k-means clustering in the original data.

	k = 2			k = 4		
	balance	Aver. Silhouette	K-median Objec.	balance	Aver. Silhouette	K-median Objec.
$(2, \infty)$	0.7353	0.0118	4060.61	0.5882	-0.0448	4038.87
$(2, 18.61958)$	0.7073	0.2081	3683.90	0.6667	0.0559	3658.68
$(3, \infty)$	0.6818	0.0652	3881.94	0.6552	-0.0420	4170.90
$(3, 18.61958)$	0.6818	0.1673	3546.61	0.6250	0.0633	3437.27
$(4, \infty)$	0.6500	0.0705	3949.70	0.6522	-0.0560	4160.95
$(4, 18.61958)$	0.7000	0.1626	3590.71	0.6429	0.0740	3547.68
$(5, \infty)$	0.6744	0.0594	3922.27	0.6842	-0.0527	3836.52
$(5, 18.61958)$	0.6154	0.2254	3475.19	0.5625	0.0593	3470.53
A-R Ward	0.6129	0.2948	3866.83	0.3571	0.2847	3356.86
A-R Kmeans	0.5238	0.3961	3420.14	0.4783	0.2863	3123.96
Kmeans	0.3409	0.4215	3265.82	0.1111	0.3912	2988.61

and black and hispanic individuals as $S = -1$. The appropriate parameters for the dissimilarities are chosen to give a good performance (after a grid search as suggested in Section 3.5). The best results are obtained with our adaptation of Ward's method with δ_2 and parameters $(u = 3.125, v = 10)$ and k-means after applying δ_1, with parameters $(U = 0, V = 500)$, and a MDS embedding. Results are given in Table 3.5. We generally see that again the balance given by using fairlets is higher than the one obtained with our procedures. However, we see that our procedures produce more identifiable and compact clusters. As a remark, we see that both procedures achieve a nice improvement in fairness compared to the k-means solution in its non fair version.

A plot of some of the clusterings can be seen in Figure 3.5. Visually it is quite clear why the average silhouette index is higher in the attraction-repulsion clustering than in the fair k-median. It also clarifies what we mean by more identifiable and compact clusters.

Figure 3.5: First row: k-means for 2 and 4 clusters in the unperturbed (original) data. Second row: k-means for 2 and 4 clusters in the MDS setting with δ_1. Third row: Ward's method for 2 and 4 clusters with δ_2. Forth row: fair k-median as in [Chierichetti et al., 2017] with $t' = 5, \tau = 18.61958$ for 2 and 4 clusters. Circles represent not white individuals; squares represent white individuals.

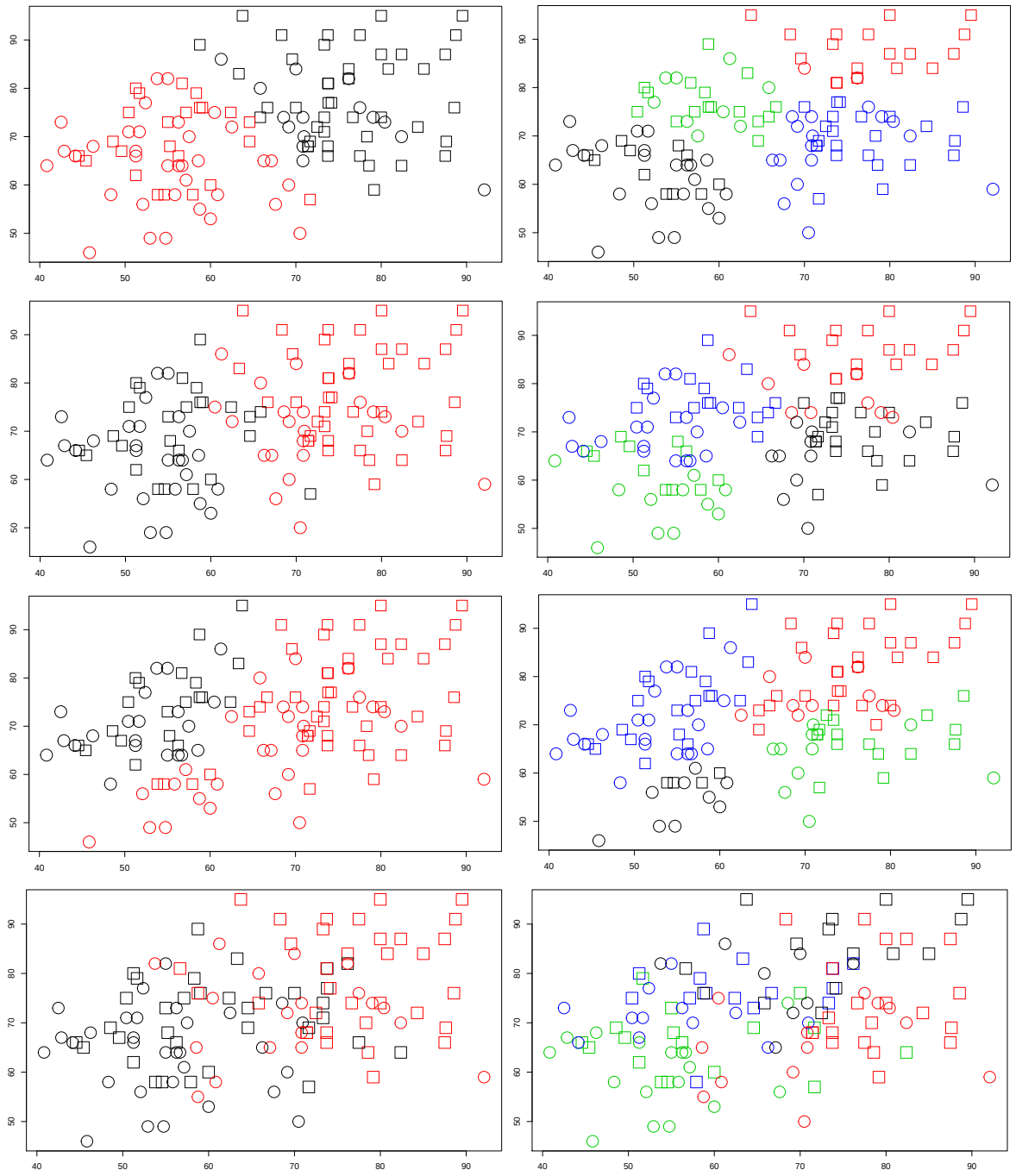
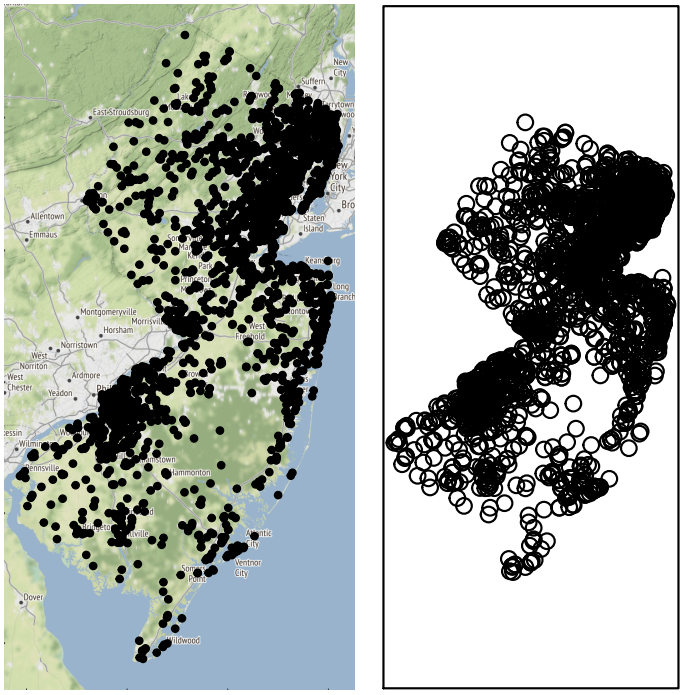


Figure 3.6: Left: location of schools in the state of New Jersey. Right: mds-embedding of the straight-line distances between the schools.



3.6.3 Civil Rights Data Collection

In this section we are going to apply our procedure to the Schools Civil Rights Data Collection (CRDC) for the year 2015-2016 which is available for download in the following link https://ocrdata.ed.gov/DownloadDataFile. In particular we are going to work with the data for entry, middle level and high schools in the state of New Jersey.

From the CRDC data we can collect the number of students in each school that belong to six distinct categories: *Hispanic, American Indian/Alaska Native, Asian, Native Hawaiian/Pacific Islander, Black, White.* An entry of our dataset looks like this.

	LEA_STATE_NAME	LEAID	SCH_NAME	hispanic	native_american	asian	pacific_islander	black
53559	NEW JERSEY	3400004	Chatham High School	46	0	106	2	19
53560	NEW JERSEY	3400004	Chatham Middle School	52	0	91	0	7
53561	NEW JERSEY	3400004	Lafayette Avenue School	28	0	70	0	4
53562	NEW JERSEY	3400004	Milton Avenue School	19	0	37	0	0
53563	NEW JERSEY	3400004	Washington Avenue School	31	0	34	0	0
53564	NEW JERSEY	3400004	Southern Boulevard School	25	0	58	0	4

	white	total	hispanic_frac	native_american_frac	asian_frac	black_frack	pacific_islander_frac	white_frac
53559	1021	1204	0.03820598	0	0.08803987	0.015780731	0.00166113	0.8480066
53560	877	1052	0.04942966	0	0.08650190	0.006653992	0.00000000	0.8336502
53561	526	647	0.04327666	0	0.10819165	0.006182380	0.00000000	0.8129830
53562	274	355	0.05352113	0	0.10422535	0.000000000	0.00000000	0.7718310
53563	337	427	0.07259953	0	0.07962529	0.000000000	0.00000000	0.7892272
53564	349	461	0.05422993	0	0.12581345	0.008676790	0.00000000	0.7570499

Additionally, using *geolocate* of the package *ggmap* in R we are able to extract latitude and longitude coordinates of the schools in New Jersey. Hence we have a precise location for the schools and can calculate distances in a straight line between them. A plot of the locations can be seen in Figure 3.6.

A relevant problem in public management is how to group schools in districts in order to improve decision making. This is a typical problem in governance, whit a famous example being electoral districts (constituencies) in the USA.

Usually districts are made mainly taking into account geographical and political information. Hence, we can fall into disparate impact decisions which is undesirable. However, imposing total fairness seems to be inadequate since geographical information is very relevant. Therefore, we think that our methodology can be successfully applied to this type of problems. In the following we present our results.

First let us state that the total proportion of students with respect to race is

$$p_t = (0.242309362, 0.001795377, 0.098752930, 0.002761563, 0.160439062, 0.474144492).$$

Let $\mathcal{C}$ be a clustering of the schools into K clusters, therefore we have K districts and for each, $1 \leq k \leq K$, there is an associated proportions vector p_k. A simple measure for the unfairness of a partition is

$$\text{unfairness}(\mathcal{C}) = \frac{1}{K} \sum_{k=1}^{K} \|p_k - p_t\|,$$

where $\|\cdot - \cdot\|$ denotes the usual Euclidean distance. Notice that unfairnes$(\mathcal{C}) = 0$ means that the partition is completely fair with respect to the disparate impact doctrine.

Recall that the average silhuette index can be seen as a measure of the compactness of the clusters in a partition. Hence our methodology is to use attraction-repulsion clustering where tuning is done over a mesh of distinct parameters, and the best parameters are those that give the lowest unfairness while keeping the average silhuette index over some treshold τ. In this way we impose the maximum improvement in fairness while keeping part of the geographical information codified by the distance. This procedure can be seen in Algorithm 7.

Codification of the protected attributes in this case is straight forward, it is just the number of students of the school in each category. Hence Catham High School has $S_{C.H.S.} = (46, 0, 106, 2, 19, 1021)$. We want to stress that another easy possibility is to use proportions with respect to the total number of students, i.e. $(0.03820598, 0, 0.08803987, 0.015780731, 0.00166113, 0.8480066)$.

Selecting the grid or mesh of parameters for the different dissimilarities is an important task. For some of them, δ_2 and δ_3, it is mainly an analytical task of selecting the best values. However, in proposing candidates for V in δ_1 and δ_4 we can use available information as we will see below. In this example we will concentrate on δ_1, δ_2 and δ_4. As hierarchical clustering methods, since we only have distances between schools, we will use complete, average and single linkage. Trough a MDS embedding we will use k-means. Hence, cluster.methods = (complete, average, single, k-means) in Algorithm 7.

The grid we use for δ_2 in Algorithm 7 is formed by $parameters = \{(u, v)\}$ with $u = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ and $v = 0.1, 0.3, 0.5, 0.7, 0.9, 1, 3.25, 5.57.75, 10$.

As we stated previously, for proposing values for $V = v_0\tilde{V}$ in δ_1 and δ_4 we are going to use some a priori information that can be corroborated by the data. The most numerous minorities are the Hispanic, Asian and Black communities. Even more, it is well known that poor neighbourhoods have higher concentration of minorities, and therefore schools

Algorithm 7 Tuning

Input: *data, cluster.methods,* δ_i*, parameters,* τ

for *cluster.method* in *cluster.methods* **do**
  $D \leftarrow$ distance matrix computed using unprotected attributes in *data*.
  for *parameter* in *parameters* **do**
    $\Delta \leftarrow$ dissimilarity matrix computed using δ_i, *parameter* and entries of protected and unprotected attributes in *data*
    if *cluster.method* = k-means **then**
      $X \leftarrow$ MDS embedding using Δ
    end if
    $\mathcal{C} \leftarrow$ clustering using *cluster.method*
    $U \leftarrow$ unfairnes($\mathcal{C}$)
    $aS \leftarrow$ average silhouette index for $\mathcal{C}$ using D.
  end for
  param.values $\leftarrow$ all respective touples (U, aS) for the different *parameter* values
  best.parameter $\leftarrow$ *parameter* corresponding to the entry in *param.values* such that $aS \geq \tau$ and with lowest U.
end for

Output: best parameter for dissimilarity δ_i for each clustering method.

in those areas should be representative of that. Hence, this is a major source of unfairness in a mainly geographical clustering of the schools. Since white students are the majority, values of the proportion for the previously mentioned minorities and white students will affect the most our unfairness index. Hence we should thrive to achieve mixing in precisely these groups. Our first three proposals are variations of schemes that should improve mixing in the above mentioned communities.

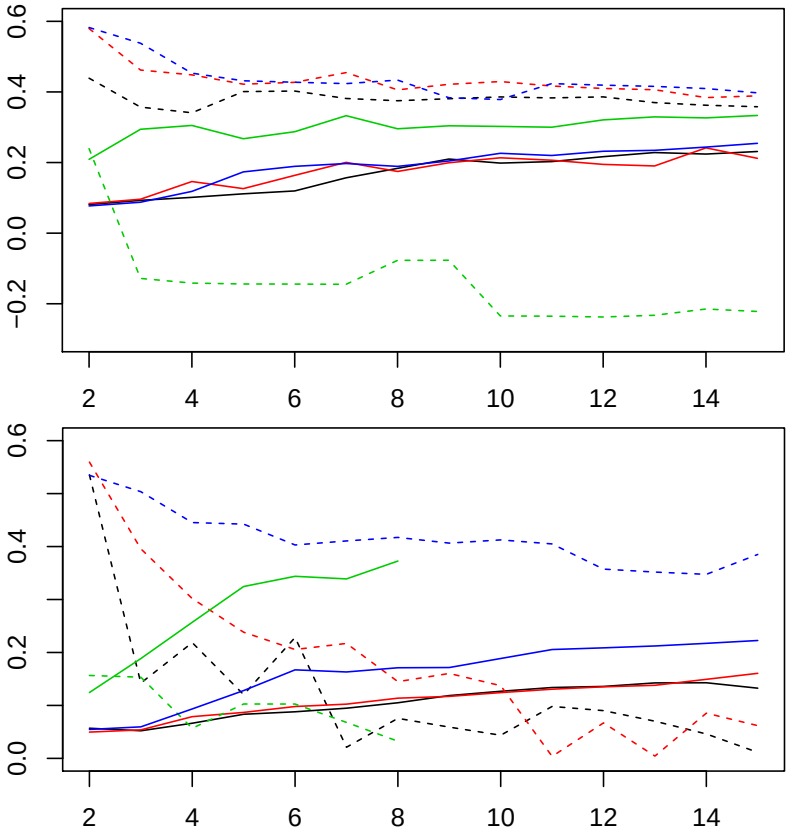
$$\tilde{V} = \left\{ \begin{pmatrix} 1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \right.$$

$$\left. \begin{pmatrix} 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & -1 & -1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \right\}.$$

The forth proposal is the obvious one, which tries to produce mixing in all communities. The fifth and sixth proposals try to impose mixing mainly for the smallest minorities American Indians/Alaska Natives and Native Hawaiians/ Pacific Islanders.

Now we can define the values we will use for the input parameters in Algorithm 7. For δ_1 we have $parameters = \{(U, v_0, V')\}$ where $U = 0_{6\times 6}$, $v_0 = 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1$ and $V' \in \tilde{V}$. For δ_4 another parameter that can incorporate a priori information is w, which in this case tells us how strong should be the influence between schools that are further apart. For the local dissimilarity δ_4 we propose to use $parameters = \{(u, v, w, V)\}$ such that $u = 0.5, 2, 8$, $v = 0.1, 1, 10$, $w = 0.1, 0.5, 0.9, 1, 5.5, 10$ and $V \in \tilde{V}$.

Figure 3.7: Continuous lines represent unfairness while dashed lines represent average silhouette index. The clustering methods are complete linkage in black, k-means in red, single linkage in green and average linkage in blue. Top row is for the unperturbed case while bottom row is for δ_1. The x label indicates the number of clusters which goes from 2 to 15. We stress that single linkage with δ_1, green line in bottom row, does not achieve $\tau > 0$ for $k \geq 9$.



Before showing results for the attraction-repulsion procedures we will cluster the data without taking into account race information, hence only geographical distance is of importance. From the top row of Figure 3.7 we can say that k-means is giving the best performance since it has the lowest unfairness index and a reasonably high average silhouette index. We stress that the k-means procedure is done in the MDS embedding shown on the right in Figure 3.6. In the left column of Figure 3.9 we can see k-means clustering for 5 and 11 clusters. We clearly see that spatial proximity is the driving force of the clustering. The values of unfairness and average silhouette index are respectively $(0.1260654, 0.421962)$ and $(0.2065493, 0.4167931)$.

In order to apply our attraction-repulsion clustering as shown in Algorithm 7 we need to fix the silhouette bound. To do this we take $\tau = 0$, and hence we are not imposing very strong compactness criteria, recall that the silhouette index varies between -1 and 1, but we still want clusters to be relatively compact. In this way we are making a trade-of between reduction in unfairness and spatial coherence. In Figure 3.8 we see the effects of the different best parameters for dissimilarities δ_1, δ_2 and δ_4 for k-means and complete linkage clustering. Generally we see that δ_1 (in red) is the dissimilarity that produces the strongest reduction in unfairness (solid line) and of course this is on behalf of a reduction in average silhouette index (dashed line). Also as expected a local dissimilarity as δ_4 (blue) brings only a modest reduction of unfairness but maintains a high spatial coherence. Hence, if we want to slightly gerrymander districts to improve fairness while maintaining a very high geographical coherence we should use δ_4. On the other hand, if we want the maximum unfairness reduction achievable with our procedure we should use δ_1.

In order to decide which clustering method is the best we use the bottom row of Figure 3.7. There we see that for attracction-repulsion clustering using δ_1, k-means (red) and complete linkage (black) produce similar reduction in unfairness while k-means keeps a relatively higher average silhouette index. Therefore in this case k-means with perturbation δ_1 and a MDS embedding seems to be the best procedure.

In Figure 3.9 we can see a visual comparison between k-means clustering in the different situations. The best parameters for the dissimilarities in the cases we have shown are the following

$$params(\mathcal{C}_5^{\delta_4}) = \left(2, 1, 10, \begin{pmatrix} 1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}\right), params(\mathcal{C}_{11}^{\delta_4}) = \left(2, 0.1, 1, \begin{pmatrix} 1 & -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}\right),$$

$$params(\mathcal{C}_5^{\delta_1}) = \left(0_{6\times 6}, 0.03, \begin{pmatrix} 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 1 \end{pmatrix}\right), params(\mathcal{C}_{11}^{\delta_1}) = \left(0_{6\times 6}, 0.07, \begin{pmatrix} 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 1 \end{pmatrix}\right).$$

What we see is that the matrices V that we introduced on an intuitive bases using a priori information work very well. Hence real world intuitions are compatible with our model. In the case of the parameters for δ_1 again what we expected from intuition is seen, i.e., in order to reduce unfairness for a bigger number of clusters it is necessary to use a stronger perturbation (a higher value for v_0). In the plots we clearly see the behaviour we have previously mentioned. The local dissimilarity δ_4 does a sort of positive gerrymandering while δ_1 imposes a stronger reduction in unfairness and hence alters significantly the clustering results.

Figure 3.8: Continuous lines represent unfairness while dashed lines represent average silhouette index. Top we have k-means and bottom complete linkage clustering. In black we have the unperturbed situation, in red we have the best δ_1 perturbation, in green the best δ_2 and in blue we have δ_4. The x label indicates the number of clusters which goes from 2 to 15.

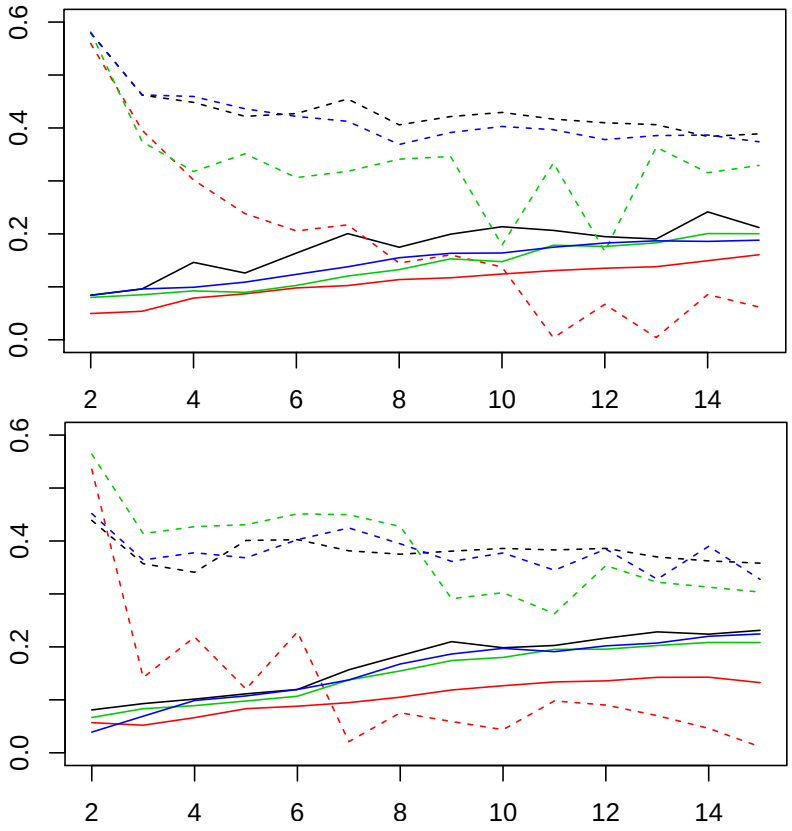
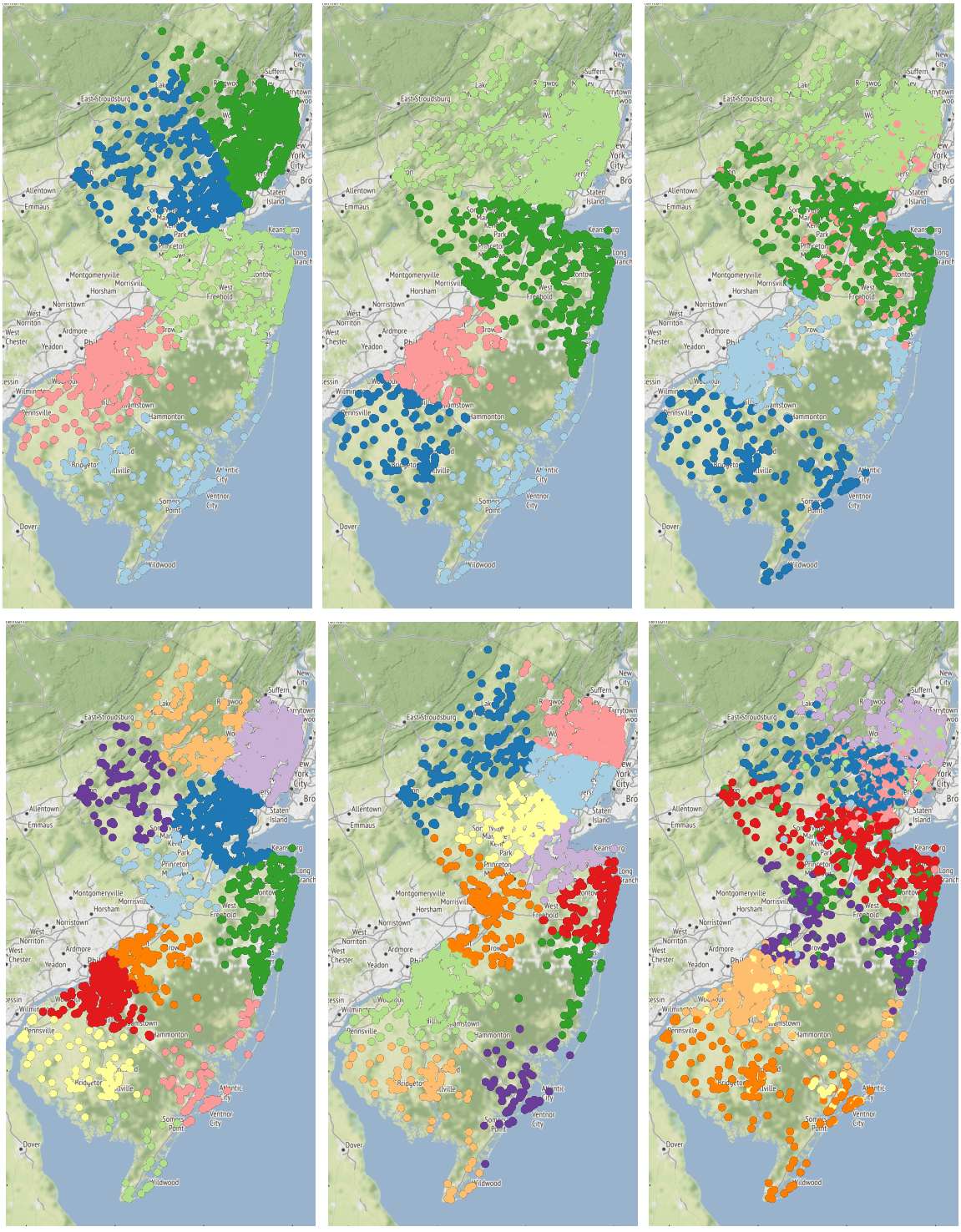


Figure 3.9: Results of k-means clustering, top row looking for 5 clusters and bottom row looking for 11. Left: unperturbed situation. Middle: δ_4 perturbation and MDS embedding. Right: δ_1 perturbation and MDS embedding.



4

A stability heuristic for selecting the number of clusters

4.1 Introduction

We have seen in several instances of this work that in statistical practice one often faces the problem of dividing a dataset $\{y_1, \ldots, y_n\} \subset \mathbb{R}^p$ into disjoint groups, i.e, one faces the problem of clustering a dataset. Immediately, the next two questions that pop into the interested persons mind are: what clustering method should I use and how many groups are really in my data. Hence, that person has arrived to the the two fundamental problems in cluster analysis. As it is often the case with fundamental problems, many different answers and procedures have been proposed for both questions. An extensive compilation of different clustering and cluster number selection procedures can be found in [Hennig et al., 2015].

Probably the most popular way of doing clustering is known as k-means clustering. That is, we are looking for a partition of the data, $\{y_1, \ldots, y_n\}$, into k disjoint sets $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ that minimize the objective function

$$\sum_{i=1}^{k} \sum_{y \in \mathcal{C}_i} \|y - \mu_i\|^2 \tag{4.1}$$

where μ_i is the mean of the points in $\mathcal{C}_i$. k-means is present in many popular software packages and as mentioned in [Hennig et al., 2015], "the algorithm is appealing in many aspects. It is computationally easy, fast, and memory-efficient. Conceptually, this method may be considered a model for the cognitive process of making a data-conditioned typology. Also, it has nice mathematical properties."

However, it is less well known that there are some populational results about the solution of the k-means problem. Let Y be a random variable, then the populational k-means problem can be written as

$$\arg \min_{\mu_j \in \mathbb{R}^p, 1 \leq j \leq k} E \min_{1 \leq l \leq k} \|Y - \mu_l\|^2. \tag{4.2}$$

Analitical solutions for problem (4.2), called principal points, were first given in [Flury, 1990] for Y following an elliptical distribution, with $k = 2$ and for general dimension

p. In [Tarpey et al., 1995] it was shown that principal points are a special case of what is called self-consistent points. Results on principal points for general k for univariate distributions can be found in [Zoppè, 1995, Mizuta, 1998].

We stress that solutions of (4.1) are estimators of the solutions of (4.2) if we assume that $\{y_1, \ldots, y_n\}$ is an i.i.d. sample coming from Y. Of course, solutions of the later are usually unknown and that is why we need approximation algorithms. However, it is quite striking that solutions, analytical or coming from a numerical solution of a system of equations, are available for such a hard problem. In this chapter we propose solutions for the populational problem (we will also call them principal points) for another popular clustering algorithm.

Model based clustering assumes that the data $\{y_1, \ldots, y_n\}$ comes from a random variable that is a mixture of some family of distributions, the most popular family being the Gaussian. Hence, the data come from a random variable Y whose density is $f(y) = \sum_{k=1}^{K} p_k f_k(y)$ where $\{p_k\}_{k=1}^{K}$ are weights and $\{f_k\}_{k=1}^{K}$ are density functions belonging to a certain family. From now on, we will assume that we are in a Gaussian family. There are two popular approaches to model based clustering, called respectively mixture and classification maximum likelihood (see [McLachlan, 1982]), from now on referred as mml and cml.

In mml the objective function to be maximized is

$$\sum_{i=1}^{n} \log \left(\sum_{k=1}^{K} p_k \varphi(y_i; \theta_k) \right) \tag{4.3}$$

over the parameters p_k and θ_k, where θ_k are the parameters for the multivariate normal density φ. In contrast, in cml one looks to maximize

$$\sum_{k=1}^{K} \sum_{y \in \mathcal{C}_k} \log \left(p_k \varphi(y; \theta_k) \right) =$$

$$\sum_{k=1}^{K} \sum_{y \in \mathcal{C}_k} \left(\log \frac{p_k}{(2\pi)^{p/2}} - \frac{(y - m_k)' S_k^{-1} (y - m_k)}{2} - \frac{1}{2} \log |S_k| \right) \tag{4.4}$$

over a partition of the data $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$ and the parameters p_k and $\theta_k = (m_k, S_k)$. Our interest will be in the cml version of model based clustering. Recall that if $S_1 = \cdots = S_k = \mathbb{I}_{p \times p}$ and $p_1 = \cdots = p_K$ then maximizing (4.4) is the same as minimizing (4.1). Hence with cml we have freedom in the elliptical shapes of the clusters, while with k-means sphericity is imposed.

It is interesting to notice that cml is inconsistent (see [Bryant and Williamson, 1978]) in the sense of estimating the true mixture parameters. That is, if the true distribution of the data is a mixture, the parameters given by the cml procedure may not converge to the true parameters. However, it is consistent for the true maximizer of the objective function (again, see [Bryant and Williamson, 1978]). Even more, since we will be concerned only with classification and not with estimation of the mixture parameters this should not be a relevant concern.

Our main objective in this chapter is to justify a conjecture and a subsequent heuristic procedure. In section 4.2, we present a conjecture on the solutions to the restricted population cml problem

$$\underset{(m_k,S_k),1\leq k\leq K}{\operatorname{argmax}} \quad E \max \log \varphi(Y; m_k, S_k) \tag{4.5}$$

when $Y \sim N(\mu, \Sigma)$, with $\mu \in \mathbb{R}^p$ and Σ a covariance matrix in $\mathbb{R}^{p\times p}$, for general K and p. We stress that it is the restricted cml problem, since weights are equal for all the groups and are not a relevant part of the optimization problem. This problem can be understood as an extended k-means where any elliptical shape can be used for the distributions. It is quite shocking that solutions for a problem with so many degrees of freedom seem to be achievable by solving a system of $K-1$ non-linear equations.

From Conjecture 4.1 in Section 4.2 and the results in [Flury, 1990, Tarpey et al., 1995] we can infer that solutions of (4.5) and solutions of (4.2) behave very differently. In particular, see Figure 4.3, principal points for the restricted cml are more unstable (in a certain sense specified later) than principal points for k-means. This behaviour will be useful for determining the number of clusters of restricted cml using stability criteria as shown in Section 4.4.

Stability concerns can be intuitively justified by the question "If we cluster data from a new sample from the same DGP [data generating process], how likely are we getting a similar partition as the current one?" (see [Hennig et al., 2015]). Since, in practice the DGP is unknown we have to resort to resampling techniques such as bootstrapping or sub-sampling. In this way we may take different sub-samples of the original data, do a clustering procedure on them and compare the different partitions via some clustering similarity criteria. If the clusterings are similar this means that at least we are detecting stable structures in the data. Hence when some instability appears we may say that something has went wrong. This is, loosely speaking, the intuition behind stability used for detecting the optimal number of clusters.

Stability can be measured in a global sense, comparing different partitions between them, or locally, meaning that we compare how stable are clusters individually. We will be interested in the later version since it makes more sense in view of Conjecture 4.1. In Section 4.4 we introduce our second main result, a procedure for cluster-wise stability measurement based on the Wasserstein k-barycenters (see (14)). The advantage of this procedure is that it is well suited for model based clustering and that it can be used for partitions coming from different data in a straight forward fashion. We see that our method is competitive with state of the art cluster-wise stability procedures in a synthetic study.

4.2 Behaviour of the solutions of the restricted cml problem

In this section we are interested in providing a solution for problem (4.5). That is, we want to find vectors $\{m_k\}_{k=1}^K \in \mathbb{R}^p$ and covariance matrices $\{S_k\}_{k=1}^K \in \mathbb{R}^{p\times p}$ that maximize the

objective function

$$E\max\left(\log\frac{e^{\frac{-D^2(Y,m_1,S_1)}{2}}}{(2\pi)^{p/2}|S_1|^{1/2}},\ldots,\log\frac{e^{\frac{-D^2(Y,m_K,S_K)}{2}}}{(2\pi)^{p/2}|S_K|^{1/2}}\right) \tag{4.6}$$

where $D^2(y,m_k,S_k)=(y-m_k)'S_k^{-1}(y-m_k)$ is the squared Mahalanobis distance and $|S_k|$ is the usual notation for the determinant of a matrix S_k. As before, we refer to problem (4.6) as (the population version of) the restricted (since cluster weights are equal and hence omitted) cml problem.

It is straightforward that an equivalent formulation of the problem is

$$\max_{(m_k,S_k),1\le k\le K} E\max\left(-\frac{D^2(Y,m_k,S_k)}{2}-\frac{1}{2}\log|S_k|\right)$$

and therefore we may concentrate on

$$\min_{(m_k,S_k),1\le k\le K} E\min\left(D^2(Y,m_k,S_k)+\log|S_k|\right).$$

Taking $Y=\Sigma^{1/2}Z+\mu$ and $Z=(Z_1,\ldots,Z_p)'\sim N(0,\mathbb{I})$, where $\mathbb{I}$ is the identity matrix of dimension p, we can rewrite the expectation as

$$E\min\Bigg(D^2\left(Z,\Sigma^{-1/2}(m_1-\mu),\Sigma^{-1/2}S_1\Sigma^{-1/2}\right)+\log|\Sigma^{-1/2}S_1\Sigma^{-1/2}|,\ldots,$$

$$D^2\left(Z,\Sigma^{-1/2}(m_K-\mu),\Sigma^{-1/2}S_K\Sigma^{-1/2}\right)+\log|\Sigma^{-1/2}S_K\Sigma^{-1/2}|\Bigg)-\log|\Sigma^{-1}|.$$

The previous results imply that we may restrict ourselves to solving

$$\operatorname*{argmin}_{(m_k,S_k),1\le k\le K} E\min\left(D^2(Z,m_k,S_k)+\log|S_k|\right) \tag{4.7}$$

with $Z=(Z_1,\ldots,Z_p)'\sim N(0,\mathbb{I})$.

A useful reformulation of the problem is the following. Let us define

$$G(m_1,S_1,\ldots,m_K,S_K)=E\min\left(D^2(z,m_1,S_1)+\log|S_1|,\ldots,D^2(z,m_K,S_K)+\log|S_K|\right) \tag{4.8}$$

and the regions

$$R_k=\left\{z\in\mathbb{R}^p: D^2(z,m_k,S_k)+\log|S_k|\le D^2(z,m_j,S_j)+\log|S_j|,\forall j\ne k,1\le j\le K\right\}.$$

We have

$$\begin{aligned}G_R(m_1,S_1,\ldots,m_K,S_K)&=\sum_{k=1}^K\left(EI_{R_k}(Z)D^2(Z,m_k,S_k)+\log|S_k|EI_{R_k}(Z)\right)\\&=\sum_{k=1}^K\left(P(R_k)ED^2(X_k,m_k,S_k)+\log|S_k|P(R_k)\right)\\&=\sum_{k=1}^K P(R_k)\left(ED^2(X_k,m_k,S_k)+\log|S_k|\right)\end{aligned} \tag{4.9}$$

where $X_k \sim Z|R_k$.

In order to use a well known result, let us write $ED^2(X_k, m_k, S_k) = t_k$ for some $t_k > 0$, then we have

$$E(X_k - m_k)' \left(\frac{t_k}{p} S_k\right)^{-1} (X_k - m_k) = p. \tag{4.10}$$

From [Grübel, 1988] we know that the pair $(m_k, \frac{t}{p} S_k)$ that fulfils (4.10) and achieves the minimum determinant is the pair $(EX_k, \mathrm{Cov(X_k)})$. For short, we write $\mathrm{Cov}(X_k) = \Sigma_{X_k}$. Therefore $S_k = \frac{p}{t_k}\Sigma_{X_k}$ has the minimum determinant when $ED^2(X_k, m_k, S_k) = t_k$. Furthermore, we have

$$t_k + \log\left|\frac{p}{t_k}\Sigma_{X_k}\right| = t_k + p\log\frac{p}{t_k} + \log|\Sigma_{X_k}| \geq p + \log|\Sigma_{X_k}| \tag{4.11}$$

since $t_k/p + \log(p/t_k) - 1 \geq 0$ for $p/t_k \geq 0$. From the fact that $ED^2(X_k, EX_k, \Sigma_{X_k}) = p$, inequality (4.11) implies that the minimum of the objective function (4.9), for fixed regions $\{R_k\}_{k=1}^K$, is achieved on $((m_1, S_1), \ldots, (m_K, S_K)) = ((EX_1, \Sigma_{X_1}), \ldots, (EX_K, \Sigma_{X_K}))$. Hence we introduce

$$G_{R,m} = G_R(EX_1, \Sigma_{X_1}, \ldots, EX_K, \Sigma_{X_K}) = p + \sum_{k=1}^{K} P(R_k)\log|\Sigma_{X_k}| \tag{4.12}$$

whit $G_{R,m} \leq G_R$. Hence, the reformulation in terms of regions has been quite useful allowing us to obtain minimizers for fixed sets of regions.

Another useful relation can be obtained in the case where $K = 2$. From the equality $EZ = P(R_1)EX_1 + P(R_1^c)EX_2$, we obtain that $EX_2 = -\frac{P(R_1)}{P(R_1^c)}EX_1$. Furthermore,

$$EZ_iZ_j = P(R_1)\mathrm{Cov}(X_{1i}, X_{1j}) + P(R_1^c)\mathrm{Cov}(X_{2i}, X_{2j}) + P(R_1)EX_{1i}EX_{1j} + P(R_1^c)EX_{2i}EX_{2j},$$

and we can write

$$\Sigma_Z = \mathbb{I} = P(R_1)\Sigma_{X_1} + P(R_1^c)\Sigma_{X_2} + P(R_1)EX_1(EX_1)' + P(R_1^c)EX_2(EX_2)'. \tag{4.13}$$

From this we obtain an useful relation (that will be used in the next section), just taking traces on both sides of (4.13), where we use $q = P(R_1)$ to simplify notation,

$$q\mathrm{tr}(\Sigma_{X_1}) + (1-q)\mathrm{tr}(\Sigma_{X_2}) = p - \|EX_1\|^2\frac{q}{1-q}. \tag{4.14}$$

From the discussion above, solving the population restricted cml problem (4.7) is equivalent to finding and adequate partition of the space $\mathbb{R}^p$ given by $\{R_k^*\}_{k=1}^K$ that minimizes (4.12). However, for now, let us turn to a slightly simpler problem whose relevance will be shown later.

Let us focus on the real line $\mathbb{R}$ and fix the regions $\{R_k\}_{k=1}^K$ as $R_k = [c_{k-1}, c_k] \subset \mathbb{R}$ for $k = 1, \ldots, K$ where $c_0 = -\infty < c_1 < \cdots < c_{K-1} < c_K = \infty$. In this setting, minimizing (4.9) is the same as minimizing

$$H(c_1, \ldots, c_{K-1}) = \sum_{k=1}^{K} q_k \log\sigma_k^2$$

where

$$q_k = \int_{c_{k-1}}^{c_k} \varphi(x)dx, \quad A_k = \int_{c_{k-1}}^{c_k} x^2\varphi(x)dx, \quad B_k = \int_{c_{k-1}}^{c_k} x\varphi(x)dx \tag{4.15}$$

and

$$\mu_k = \frac{B_k}{q_k} \quad \text{and} \quad \sigma_k^2 = \frac{A_k}{q_k} - \mu_k^2. \tag{4.16}$$

Since we are interested in a minimum, the following system of equations is pertinent

$$\partial_{c_k} H = 0 \quad \text{for} \quad 1 \leq k \leq K-1.$$

Taking partial derivatives and after some cumbersome calculations we arrive at the following system of equations

$$\partial_{c_k} H = \varphi(c_k)\left(\frac{1}{\sigma_k^2}\left(c_k - \mu_k\right)^2 - \frac{1}{\sigma_{k+1}^2}\left(c_k - \mu_{k+1}\right)^2 + \log\frac{\sigma_k^2}{\sigma_{k+1}^2}\right) = 0 \quad \text{for} \quad 1 \leq k \leq K-1. \tag{4.17}$$

Now we have everything to introduce our main result, which is given as a conjecture.

Conjecture 4.1. *Let $\{c_1^*, \ldots, c_{K-1}^*\}$ be a solution of the system (4.17) with corresponding μ_k^* and $\sigma_k^{2^*}$, given by (4.15) and (4.16), for $k = 1, \ldots, K$. Let u be an unitary vector indicating a direction in $\mathbb{R}^p$. Let $\mathcal{B} = \{u, u^\perp\}$ be a basis of $\mathbb{R}^p$, where $u^\perp$ is an orthonormal basis of the orthogonal complement of u.*

We have that $\{(m_k^*, S_k^*)\}_{k=1}^K$ given by

$$m_k^* = \mu_k^*(1, 0, \ldots, 0)', \quad S_k^* = \mathrm{diag}(\sigma_k^{2^*}, 1, \ldots, 1)$$

are a solution for (4.7) in the basis $\mathcal{B}$. Hence, in the standard basis for $\mathbb{R}^p$, the regions, $\{R_k^*\}_{k=1}^K$, minimizing (4.9) are delimited by $K-1$ hyperplanes orthogonal to u which contain respectively the points $c_k^* u$.

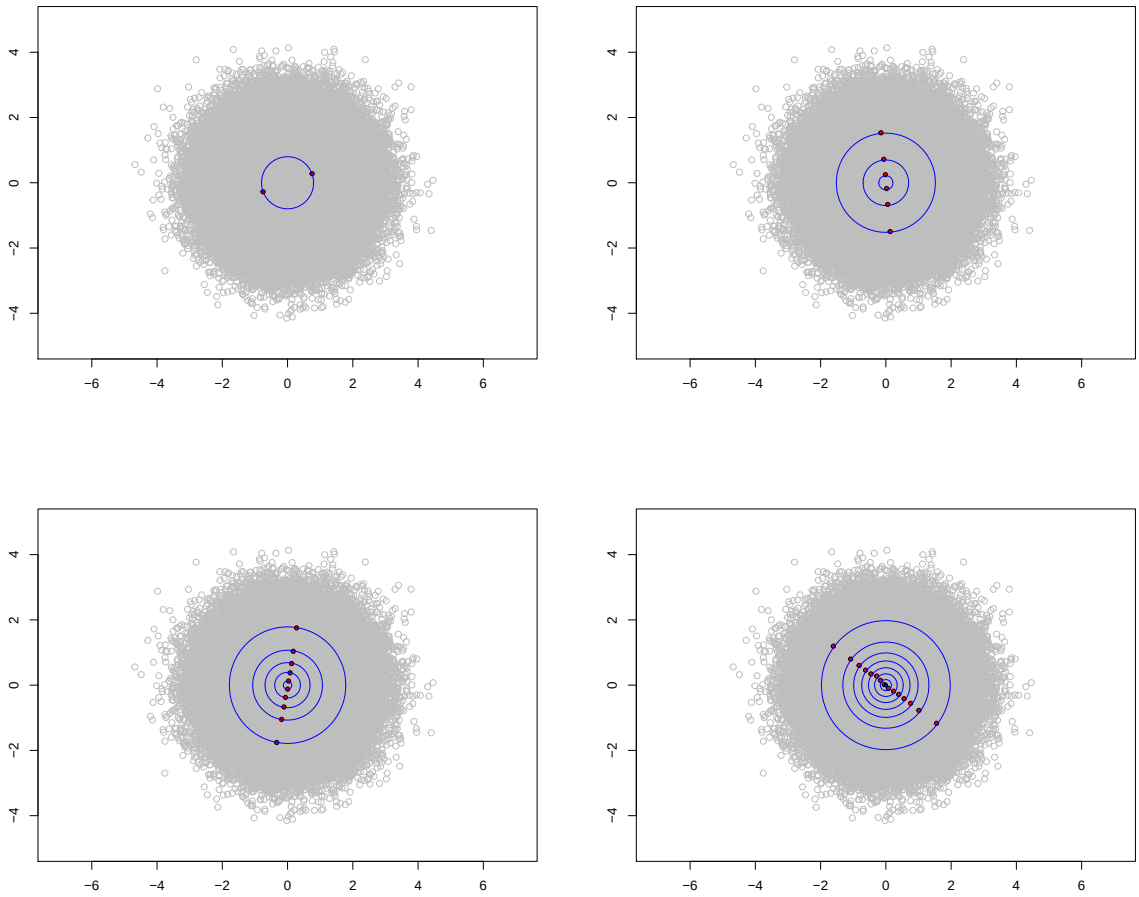
If we call B the change of basis matrix associated with $\mathcal{B}$, we have that the solutions of the problem (4.6), written in the standard basis of $\mathbb{R}^p$ are

$$m_k^* = \mu_k^* \Sigma^{1/2} u + \mu, \quad S_k^* = \Sigma^{1/2} B \mathrm{diag}(\sigma_k^{2^*}, 1, \ldots, 1)\Sigma^{1/2}.$$

In words, Conjecture 4.1 means that the solution of the general problem (4.6) reduces to that of problem (4.7). Even more, problem (4.7) reduces essentially to solving a problem in the real line given by (4.17). It is remarkable that for any dimension p, any direction u and any number of clusters $K \geq 2$ there are always K points, $\{m_1^*, \ldots, m_K^*\}$, that lie along u and are solutions for the restricted cml problem. This is in sharp contrast with the case of principal points when the distribution of Y is not a standard normal. We think that this is a result of the extra freedom of choosing covariance matrices in the cml problem compared to the k-means problem.

Let us see our conjecture at work. We start with Figure 4.1, where we are in $\mathbb{R}^2$ with $Y \sim N(0, \mathbb{I}_{2\times 2})$ and we take a sample of 2×10^5 points. We are going to look for $k = 2, 6, 10, 15$ clusters and use *tclust* in order to obtain a local optimum for the empirical restricted cml problem. The centers of the clusters obtained by *tclust* are shown as red points. Since Conjecture 4.1 indicates that the centers should lie on a line with separation given by appropriate ellipses (circles) we draw them in blue. Indeed, what we see is that

Figure 4.1: Behaviour of the centers m_k^*, for $k = 2, 6, 10, 15$, obtained by *tclust* (in red) with respect to the prediction of Conjecture 4.1 (in blue). The underlying distribution is $N(0, \mathbb{I}_{2\times 2})$ and we draw a sample of size 2×10^5 depicted in gray.



the empirical solutions, in red, lie on an approximately straight lines with separation that is very close to the theoretical one, given in blue.

We recall that the solution given by *tclust* is an estimator of the population restricted cml problem, so we should expect that convergence depends on the number of sample points, on the dimension of the space and on the number of clusters. Our numerical experiments clearly reflect this. We see in Figure 4.1 that for $k = 2, 6$ solutions are extremely close to the behaviour described in our conjecture, while for $k = 10, 15$ they are slightly further.

Dimension effects are studied in Figure 4.2. Here we are in $\mathbb{R}^3$ with $Y \sim N(0, \mathbb{I}_{3\times 3})$ and again a sample of 2×10^5 points. Once again, we see that *tclust* solutions are a good estimate of the theoretical solutions for $k = 6, 10$, however they seem to be worst estimates than for the same case in two dimensions. Even more in 2d the estimates become quite bad (do not lie on an almost straight line) around $k = 20$ while in 3d this happens around $k = 16$. For 2×10^5 points coming from $Y \sim N(0, \mathbb{I}_{6\times 6})$ estimates become bad around $k = 6$. As usual, an increase in the sample size allows to observe the expected behaviour for a bigger number of clusters.

We have previously mentioned that Conjecture 4.1 and the results on principal points in [Flury, 1990, Tarpey et al., 1995] imply that behaviour of restricted cml and k-means should be very different. We have a glimpse on that difference in Figure 4.3. We are going to look for $k = 2, 3, 4$ clusters with *tclust* and k-means in 60 different samples of 10^4 points drawn from a $N(0, \text{diag}(2^2, 1))$ (left column) and $N(0, \text{diag}(3^2, 1))$ (right column) respectively. In green we have the centers of the clusters given by k-means and in red the centers given by *tclust*. In blue is shown the prediction of Conjecture 4.1. First, let us recall how close the empirical estimate, i.e., the results from *tclust*, are to the theoretical predictions, and how we clearly appreciate that centers are lying on different directions. Even more, we see that there is a much bigger instability present in the solutions of restricted cml that the one present in the solutions of k-means. This will be in the heart of our stability procedure for determining the number of clusters that we will present in Section 4.4. But for now, let us explore some ideas for a possible prove of Conjecture 4.1 in the next section.

4.3 Thoughts on proving the one-dimensional case for k = 2

In this section we provide some insights on our attempts at proving Conjecture 4.1 in the one-dimensional case ($p = 1$) when looking for $k = 2$ clusters. First let us stress that we think that the approach followed by Flury in [Flury, 1990] to produce the principal points for $k = 2$ and general dimension is not applicable for the restricted cml problem. This is due to the change in geometry produced by the scalar product being dependent on the covariance matrices. Second, we think that methods provided in [Zoppè, 1995] for finding principal points in the one dimensional case but for general k are also not suitable for the restricted cml problem. Indeed, this methods rely on a particular parametrization which in turn relies on Euclidean geometry of the k-means problem, which is not the geometry of the cml problem.

Hence a first approach for handling Conjecture 4.1 is to start with the simplest case

Figure 4.2: Behaviour of the centers m_k^*, for $k = 6, 10$, obtained by *tclust* (in red) with respect to the prediction of Conjecture 4.1 (in blue). The underlying distribution is $N(0, \mathbb{I}_{3\times 3})$ and we draw a sample of size 2×10^5.

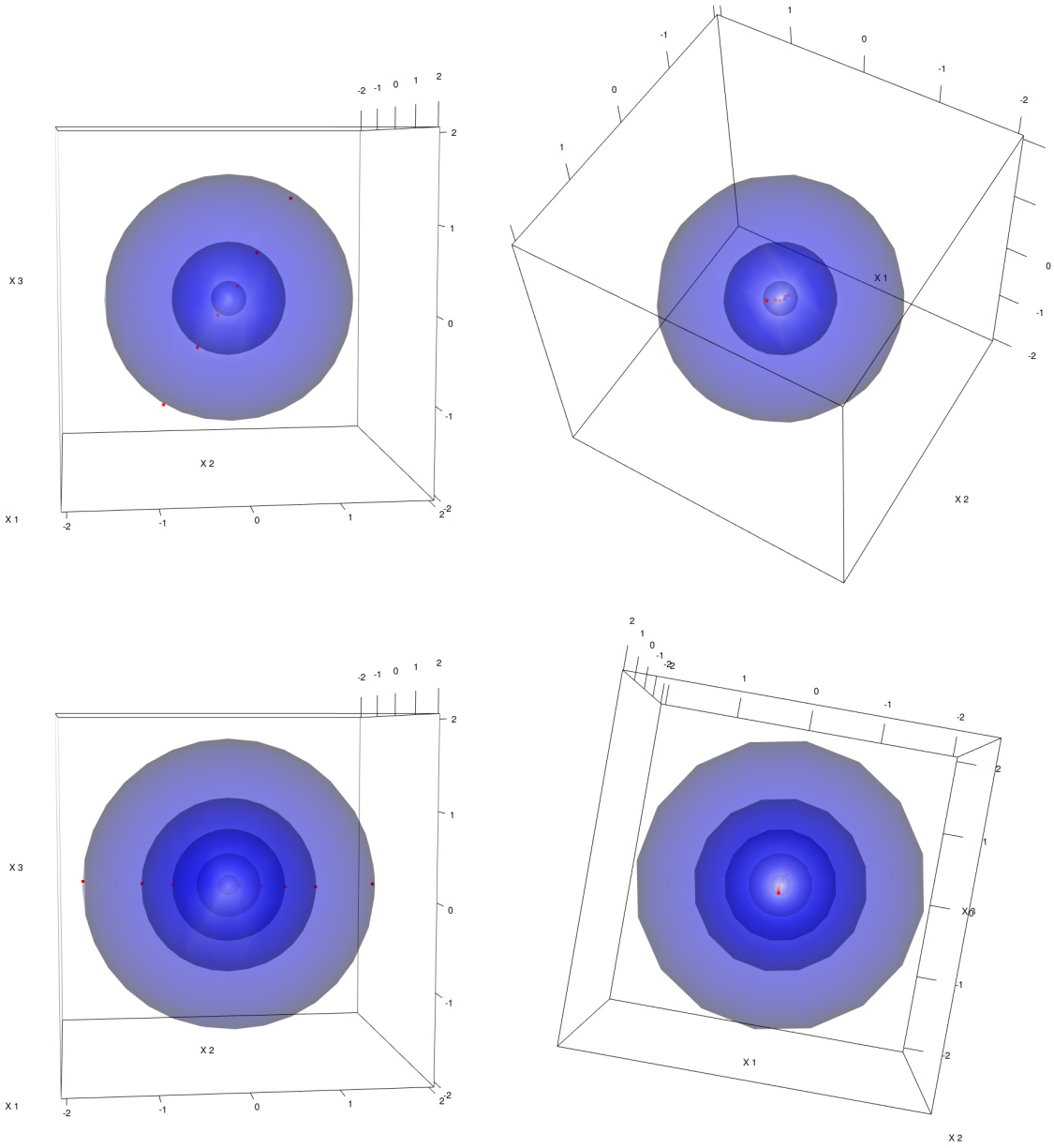
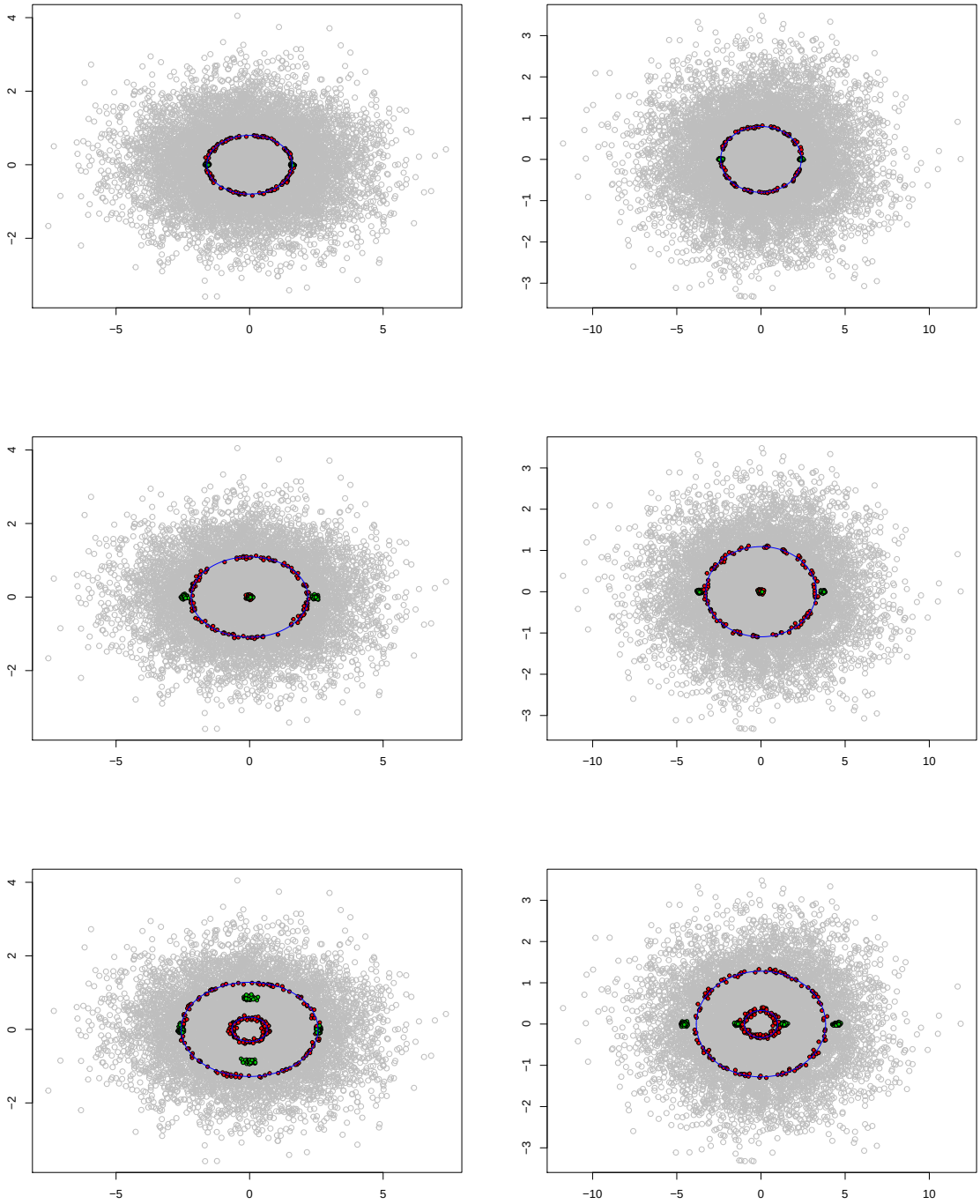


Figure 4.3: Comparison of the behaviour of restricted cml and kmeans centers, in red and green respectively, for $k = 2, 3, 4$. In blue we have the predicted cml behaviour given by Conjecture 4.1. Left column corresponds to 60 samples of 10^4 points taken from a $N(0, \mathrm{diag}(2^2, 1))$ and the right column to 60 samples from a $N(0, \mathrm{diag}(3^2, 1))$. In gray is shown a particular sample for reference.



and try to do the necessary calculations. Hence we will fix $p = 1$ and $k = 2$. We stress that solving the corresponding problem for k-means is fairly simple. The discussion bellow may be an indication of the cml problem being more involved than the k-means problem.

Since in the one-dimensional problem and for $k = 2$ the optimal region R_1 in (4.7) must be of type (a, b) or a complement of it (and then $R_2 = R_1^C$ is of the last type), we can equivalently focus on the functional

$$\tilde{H}(a,b) = \min_{\mu_1,\mu_2,\sigma_1^2,\sigma_2^2} E\Big[\big(\tfrac{(X-\mu_1)^2}{\sigma_1^2} + \log\sigma_1^2\big) I_{(a,b)}(X) + \big(\tfrac{(X-\mu_2)^2}{\sigma_2^2} + \log\sigma_2^2\big) I_{(a,b)^C}(X)\Big]$$

for $-\infty \le a \le b \le \infty$. Once the region R_1 is fixed, the optimal location and scale parameters are the conditional mean and variance. This means that $\tilde{H}(a,b) = 1 + H(a,b)$ with

$$H(a,b) = (\Phi(b) - \Phi(a)) \log\sigma^2_{(a,b)} + (1 - \Phi(b) + \Phi(a)) \log\sigma^2_{(a,b)^C},$$

$$\sigma^2_{(a,b)} = \frac{1}{\Phi(b) - \Phi(a)} \int_a^b x^2\varphi(x)dx - \Big(\frac{1}{\Phi(b) - \Phi(a)} \int_a^b x\varphi(x)dx\Big)^2,$$

$$\sigma^2_{(a,b)^C} = \frac{1}{1 - \Phi(b) + \Phi(a)}\Big(1 - \int_a^b x^2\varphi(x)dx\Big) - \Big(\frac{-1}{1 - \Phi(b) + \Phi(a)} \int_a^b x\varphi(x)dx\Big)^2.$$

We define now

$$A(a,b) = \int_a^b x^2\varphi(x)dx, \quad B(a,b) = \int_a^b x\varphi(x)dx, \quad q(a,b) = \Phi(b) - \Phi(a).$$

Then

$$H(a,b) = q(a,b) \log\sigma_1^2(a,b) + (1 - q(a,b)) \log\sigma_2^2(a,b),$$

with

$$\sigma_1^2(a,b) = \tfrac{A(a,b)}{q(a,b)} - \Big(\tfrac{B(a,b)}{q(a,b)}\Big)^2, \quad \sigma_2^2(a,b) = \tfrac{1-A(a,b)}{1-q(a,b)} - \Big(\tfrac{B(a,b)}{1-q(a,b)}\Big)^2.$$

From now on, for simplicity, we omit the explicit dependence of a and b. Taking partial derivatives and after some algebra we get

$$\begin{aligned} \partial_a H &= \log(\frac{\sigma_1^2}{\sigma_2^2})\partial_a q + \frac{q}{\sigma_1^2}\partial_a\sigma_1^2 + \frac{1-q}{\sigma_2^2}\partial_a\sigma_2^2 \\ &= \varphi(a)\left(\frac{(a + \frac{B}{1-q})^2}{\sigma_2^2} - \frac{(a - \frac{B}{q})^2}{\sigma_1^2} - \log(\frac{\sigma_1^2}{\sigma_2^2})\right) \end{aligned} \tag{4.18}$$

$$\begin{aligned} \partial_b H &= \log(\frac{\sigma_1^2}{\sigma_2^2})\partial_b q + \frac{q}{\sigma_1^2}\partial_b\sigma_1^2 + \frac{1-q}{\sigma_2^2}\partial_b\sigma_2^2 \\ &= \varphi(b)\left(\log(\frac{\sigma_1^2}{\sigma_2^2}) + \frac{(b - \frac{B}{q})^2}{\sigma_1^2} - \frac{(b + \frac{B}{1-q})^2}{\sigma_2^2}\right) \end{aligned} \tag{4.19}$$

Suppose that we are out of the region $-\infty < a < b < \infty$, hence we are in the boundary. The boundary consists of points where $a = b$ or $(a, b) = (-\infty, \infty)$ which have the same

value for H and are easy to handle taking limits in H. The other boundary points are of the type $(a, b) = (-\infty, b)$ with $-\infty < b < \infty$, hence, (4.17) is a necessary condition for the minimum of H if it exists. The case when $(a, b) = (a, \infty)$ with $-\infty < a < \infty$ is equivalent to the previous. If we are able to prove that $\partial_a H = 0 = \partial_b H$ is never fulfilled in $-\infty < a < b < \infty$, then only the boundary is relevant for extreme points. This is what we attempt below.

For any $-\infty < a < b < \infty$ we have that $0 < \varphi(a)$ and $0 < \varphi(b)$. Hence from (4.18) and (4.19) we get that $\partial_a H = 0$ and $\partial_b H = 0$ if and only if

$$\frac{(a + \frac{B}{1-q})^2}{\sigma_2^2} - \frac{(a - \frac{B}{q})^2}{\sigma_1^2} - \log(\frac{\sigma_1^2}{\sigma_2^2}) = 0 \tag{4.20}$$

$$\log(\frac{\sigma_1^2}{\sigma_2^2}) + \frac{(b - \frac{B}{q})^2}{\sigma_1^2} - \frac{(b + \frac{B}{1-q})^2}{\sigma_2^2} = 0 \tag{4.21}$$

Therefore it follows that

$$\frac{(a + \frac{B}{1-q})^2}{\sigma_2^2} - \frac{(a - \frac{B}{q})^2}{\sigma_1^2} = -\frac{(b - \frac{B}{q})^2}{\sigma_1^2} + \frac{(b + \frac{B}{1-q})^2}{\sigma_2^2}$$

$$\frac{(b-a)(b+a) - 2(b-a)\frac{B}{q}}{\sigma_1^2} = \frac{(b-a)(b+a) + 2(b-a)\frac{B}{1-q}}{\sigma_2^2}$$

and since $a < b$ we have that

$$\frac{\frac{a+b}{2} - \frac{B}{q}}{\sigma_1^2} = \frac{\frac{a+b}{2} + \frac{B}{1-q}}{\sigma_2^2}. \tag{4.22}$$

Hence (4.22) is a necessary, but not sufficient, condition for an extreme point. With some more algebra we have

$$\frac{a+b}{2}(\sigma_2^2 - \sigma_1^2) = B\left(\frac{\sigma_2^2}{q} + \frac{\sigma_1^2}{1-q}\right) = \frac{B}{q(1-q)}((1-q)\sigma_2^2 + q\sigma_1^2). \tag{4.23}$$

But we also have the relation that comes from (4.14)

$$q\sigma_1^2 + (1-q)\sigma_2^2 = 1 - \frac{B^2}{q(1-q)}.$$

This leads to $\sigma_2^2 = \frac{1}{1-q}\left(1 - \frac{B^2}{q(1-q)}\right) - \frac{q}{1-q}\sigma_1^2$. Therefore substituting in (4.23) and after some manipulation we get

$$\left(\frac{a+b}{2} - \frac{B}{q}\right)\frac{1}{1-q}\left(1 - \frac{B^2}{q(1-q)}\right) = \frac{a+b}{2}\frac{1}{1-q}\sigma_1^2.$$

Since $0 < q < 1$ we get that

$$\sigma_1^2 = \frac{\frac{a+b}{2} - \frac{B}{q}}{\frac{a+b}{2}}\left(1 - \frac{B^2}{q(1-q)}\right) = \left(1 - \frac{2B}{q(a+b)}\right)\left(1 - \frac{B^2}{q(1-q)}\right), \tag{4.24}$$

and

$$\begin{aligned}\sigma_2^2 &= \left(\frac{1}{1-q} - \frac{q}{1-q}\frac{\frac{a+b}{2} - \frac{B}{q}}{\frac{a+b}{2}}\right)\left(1 - \frac{B^2}{q(1-q)}\right)\\ &= \frac{(1-q)\frac{a+b}{2} + B}{(1-q)\frac{a+b}{2}}\left(1 - \frac{B^2}{q(1-q)}\right)\\ &= \left(1 + \frac{2B}{(1-q)(a+b)}\right)\left(1 - \frac{B^2}{q(1-q)}\right).\end{aligned}$$

If we substitute these values in (4.20) we get

$$\left(\frac{(a+\frac{B}{1-q})^2}{\left(\frac{a+b}{2}+\frac{B}{1-q}\right)} - \frac{(a-\frac{B}{q})^2}{\left(\frac{a+b}{2}-\frac{B}{q}\right)}\right)\frac{\frac{a+b}{2}}{\left(1-\frac{B^2}{q(1-q)}\right)} - \log\left(\frac{\frac{a+b}{2}-\frac{B}{q}}{\frac{a+b}{2}+\frac{B}{1-q}}\right) = 0. \tag{4.25}$$

If this equation has no solutions we would have proven that there is no local extrema in $-\infty < a < b < \infty$. Sadly for our purposes this is not the case, i.e., equation (4.25) has solutions, since it can be shown that it changes signs. Therefore we need to show that solutions of (4.25) give a σ_1^2, obtained using (4.24), that contradicts the fact that $\sigma_1^2 = A/q - (B/q)^2$. However this requires solving multiple difficult implicit equations. Therefore this path seems to be inappropriate for achieving an analytical conclusion. Nonetheless, we can check it for any particular solution of (4.25) obtained numerically.

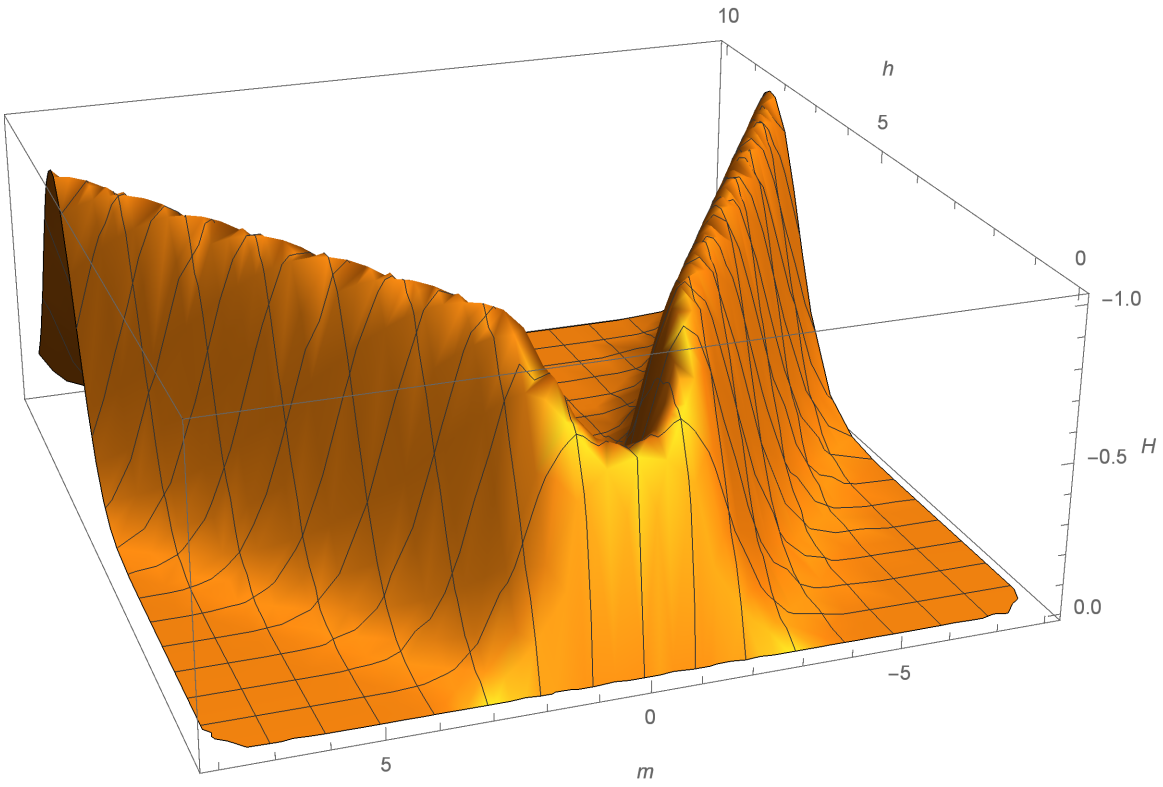
Another possibility is to use a different parametrization given by $a = m-h, b = m+h$ where $m \in [-\infty, \infty]$ and $h \in [0, \infty]$. A plot of $H(m,h)$ can be seen in Figure 4.4. From that plot we see that $H(m_0, h)$ seems to have a single minimum point $h^*_{m_0}$ for any fixed m_0, but that H is decreasing in the direction given by two nearby minima. That is H is decreasing in the direction given by the points $(m_0, h^*_{m_0})$ and $(m_0+\epsilon, h^*_{m_0+\epsilon})$ for $\epsilon > 0$ small enough. Therefore it seems that there are no local minima, and that $\lim_{m\to\infty} H(m,m)$ and $\lim_{m\to\infty} H(-m,m)$ are the absolute minima. However, proving this seems to involve the same type of implicit equations that we have discussed above.

With the previous discussion we hope that we have convinced the reader that proving Conjecture 4.1 is no joke even in the simplest case. We will delay further attempts on the prove for future work and we invite anyone interested in attacking the problem to do so. However in the next section we provide useful applications of the behaviour of the solutions for the restricted cml problem.

4.4 A heuristic for k based on cluster-wise stability

A result of a model based clustering procedure can be characterized as a collection $\{(p_1, \nu_1), \ldots, (p_K, \nu_K)\}$ where $\{p_k\}_{k=1}^K$ are weights and $\{\nu_k\}_{k=1}^K$ are probability measures associated with each cluster. It is common to have a collection of different clusterings for the same K, as a result of using resampling techniques, of using different data sources or of using a parallelization scheme. Hence a collection, $\{\mathcal{C}^i\}_{i=1}^N = \{\{\nu_1^i, \ldots, \nu_K^i\}\}_{i=1}^N$, is available, where we omit the weight information. Let us pool all distributions together to obtain $\mathcal{C} = \{\nu_i\}_{i=1}^{KN}$, and calculate the Wasserstein K-barycenter (recall (12) and (14)),

Figure 4.4: Plot of $H(m, h)$.



$\{\bar{\nu}_k\}_{k=1}^K$, for the collection $\mathcal{C}$. We also recall that associated to the barycenter there is a partition of $\mathcal{C}$ into K groups, $\{\mathcal{S}_k\}_{k=1}^K$. If the partitions $\mathcal{C}^i$ were similar to each other, we should expect that distributions in each cluster, $\mathcal{S}_k$, are similar to each other. Hence they should be similar to the "center" of the cluster $\bar{\nu}_k$. This idea gives us a natural measure of how similar are the distributions in a cluster with respect to the central element

$$V_k^l = \frac{1}{|\mathcal{S}_k|} \sum_{\nu \in \mathcal{S}_k} \mathcal{W}_2^l(\nu, \bar{\nu}_k). \tag{4.26}$$

Recall, that this measure reduces to the mean within-cluster sum of squares in the case of $l = 2$ and k-means if we consider the result of the clustering procedure to be the collection $\{(\frac{1}{K}, \delta_{\mu_k})\}_{k=1}^K$ with $\{\mu_k\}_{k=1}^K$ the centers returned by k-means.

We propose to use V_k^l as a measure of the stability of cluster k. Hence $\{V_k^l\}_{k=1}^K$ will reflect the stability of all the clusters when we look for K groups. Therefore we propose the following scheme for cluster-wise stability assessment:

- Obtain a collection of distributions $\{\mathcal{C}^i\}_{i=1}^N = \{\{\nu_1^i, \ldots, \nu_K^i\}\}_{i=1}^N$ from the respective model based clustering results $\{\{(p_k^i, \nu_k^i)\}_{k=1}^K\}_{i=1}^N$. This can be done by sub-sampling or bootstrapping a data set and then clustering. By clustering different sources of data that are modelled to come from the same generating mechanism. In the rare cases when the generating mechanism is known by sampling from it and clustering.

- Pool together all distributions in $\{\mathcal{C}^i\}_{i=1}^N$ to obtain $\mathcal{C} = \{\nu_i\}_{i=1}^{KN}$. Then obtain the K-barycenter of $\mathcal{C}$, $\{\bar{\nu}_k\}_{k=1}^K$.

- For each $1 \leq k \leq K$ take V_k^l as the stability measure for cluster k.

Inspiration for this procedures comes from the behaviour shown in Conjecture 4.1. Indeed with our procedure we should capture the instability that arises when we divide a solid cluster into several clusters. However the logic of our scheme is not limited to restricted cml clustering and can be used in any model based clustering procedure, or even in combinations of different model based clustering procedures.

For comparison we use the sub-sampling version of the procedure introduced in [Hennig, 2007]. For a data set $\{y_i\}_{i=1}^M \subset \mathbb{R}^p$ we obtain N sub-samples of size m that we denote $Y_1, \ldots, Y_N$. A clustering of the original data set will be denoted as $\mathcal{C}_Y = \{\mathcal{C}_{Y,k}\}_{k=1}^K$ and of the sub-samples as $\mathcal{C}_{Y_i} = \{\mathcal{C}_{Y_i,k}\}_{k=1}^K$ with $\mathcal{C}_{Y,k}, \mathcal{C}_{Y_i,k'} \subset \{y_i\}_{i=1}^M$. The maximum Jaccard agreement for cluster k (in $\mathcal{C}_Y$) in the clustering $\mathcal{C}_{Y_i}$ was defined as

$$s_k^i = \max_{1 \leq k' \leq K} \frac{|\mathcal{C}_{Y,k} \cap \mathcal{C}_{Y_i,k'}|}{|\mathcal{C}_{Y,k} \cup \mathcal{C}_{Y_i,k'}|}.$$

Then the mean value

$$s_k = \frac{1}{N} \sum_{i=1}^N s_k^i$$

is the stability indicator for cluster $\mathcal{C}_{Y,k}$.

For a study of the behaviour of our procedure we will use the following setting. For simplicity and visualization purposes we will be in $\mathbb{R}^2$. Clusters will come from bivariate normal distributions with respective parameters

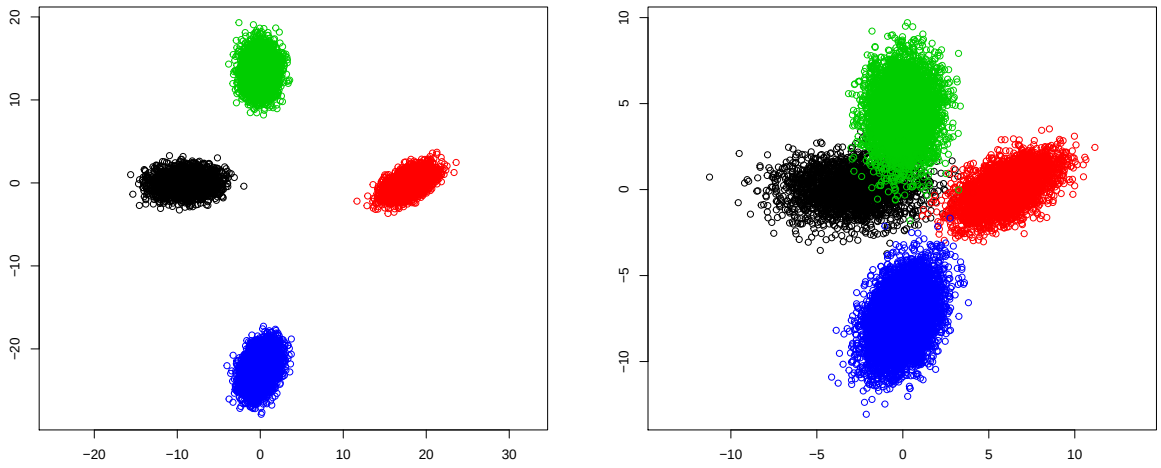
$$\theta_1 = \left((-r, 0), \begin{pmatrix} 2^2 & 0 \\ 0 & 1 \end{pmatrix}\right), \quad \theta_2 = \left((2r, 0), \begin{pmatrix} 2 & 0.8 \\ 0.8 & 1 \end{pmatrix}\right),$$

$$\theta_3 = \left((0, 1.5r), \begin{pmatrix} 1 & 0 \\ 0 & 1.5^2 \end{pmatrix}\right), \quad \theta_4 = \left((0, -2.5r), \begin{pmatrix} 1 & 0.5 \\ 0.5 & 2 \end{pmatrix}\right).$$

We will study two different cases, a very favourable one with $r = 9$ and a less favourable one with $r = 3$. We sample $n_1 = 0.3 \times 8000, n_2 = 0.5 \times 8000, n_3 = 0.7 \times 8000$ and $n_4 = 8000$ points respectively from each normal distribution and obtain a sample $\{y_i\}_{i=1}^{2\times 10^4}$. The samples for both values of r are shown in Figure 4.5. We take 60 sub-samples of size 16000 to obtain $Y_1, \ldots, Y_{60}$. For each sub-sample we look for $K = 2, 3, 4, 5, 6, 7, 8$ clusters with five different clustering procedures: k-means, restricted cml, cml, robust cml (see (2.3)) and *mclust* ([Scrucca et al., 2016]) which solves the mml problem (4.3). For each clustering procedure and each K we calculate $\{s_k\}_{k=1}^K$ and $\{V_k^1\}_{k=1}^K$. We choose V_k^1 since it is more sensitive to small variations in stability. The results for the case $r = 9$ are depicted in Figure 4.6 and for the case $r = 3$ in Figure 4.7.

In order to get a clear understanding of the results shown below we are going to analyse the particular case of cml clustering in Figure 4.6, i.e., second and forth row left. In the x axes we have the clusters labels, i.e., when we look for $K = 2$ clusters we have labels 1 and 2, when we look for $K = 3$ we have labels 1, 2 and 3 and so on. From the plot, $K = 2, 3, 4$ (black, red, green) are extremely stable giving s_k values of 1 (second row left) and V_k^1 close to 0 (forth row left). However stability decreases greatly when $K = 5, 6, 7, 8$ (blue, cyan, magenta yellow) which is depicted as a noticeable decrease in s_k and a noticeable

Figure 4.5: Data in the very favourable case $r = 9$ (left) and in the less favourable case $r = 3$ (right)



increase in V_k^1. Hence, from a stability point of view it makes sense to choose $K = 4$ as the optimal number of clusters. We stress that visual representations like Figure 4.6 and 4.7 are compact summaries of information. For example, Figure 4.8 represents the information contained in Table 4.1.

Another thing to notice is that cml clustering (also robust cml and mclust) can return empty clusters, that is why the magenta line, corresponding to $K = 7$ (second row left in Figure 4.6), only goes to six clusters. This is handled in a straightforward fashion for s_k. When using V_k^1 we select for the number K of clusters in the K-barycenter the majority vote from the sub-samples, i.e. the clustering procedure has produced a list $\{K_j\}_{j=1}^{60}$ of the number of clusters found in the respective sub-samples and we take K as the majority vote in that list. This explains why some lines end before they are supposed to. This itself is an indication that the number of clusters is smaller than the value we ask for that is provided by the original methods.

Now we should be able to extract several general conclusions from Figures 4.6 and 4.7. We see that both methods give very comparable results, for each type of clustering and for both studied cases. In fact, both methods detect that the optimal number of clusters with respect to stability is $K = 4$ (the correct one), for all methods except k-means. This makes complete sense since the results on principal points mentioned before (see also Figure 4.3) imply that overdivision of clusters should be very stable in the k-means case. This is extremely clear when looking at the first and third row left in Figure 4.7. Another interesting fact is that robust cml, i.e., what *tclust* was introduced for, with only 5% trimming is able to automatically detect that there are 5 or less clusters.

We think that the main strengths of our methodology are: the fact that we do not need to cluster the complete original sample, which sometimes can be computationally demanding, and the fact that implementation is straightforward even when different samples do not share any points. The last point usually requires some adaptation for standard criteria, however it does not make any difference for our methodology.

Figure 4.6: Results for $r = 9$. First two rows correspond to values of s_k, last two rows to values of V_k^1. From left to right the clustering algorithms are: k-means, restricted cml, cml, robust cml and *mclust*. For $K = 2, 3, 4, 5, 6, 7, 8$ we have respectively the colours black, red, green, blue, cyan, magenta and yellow.

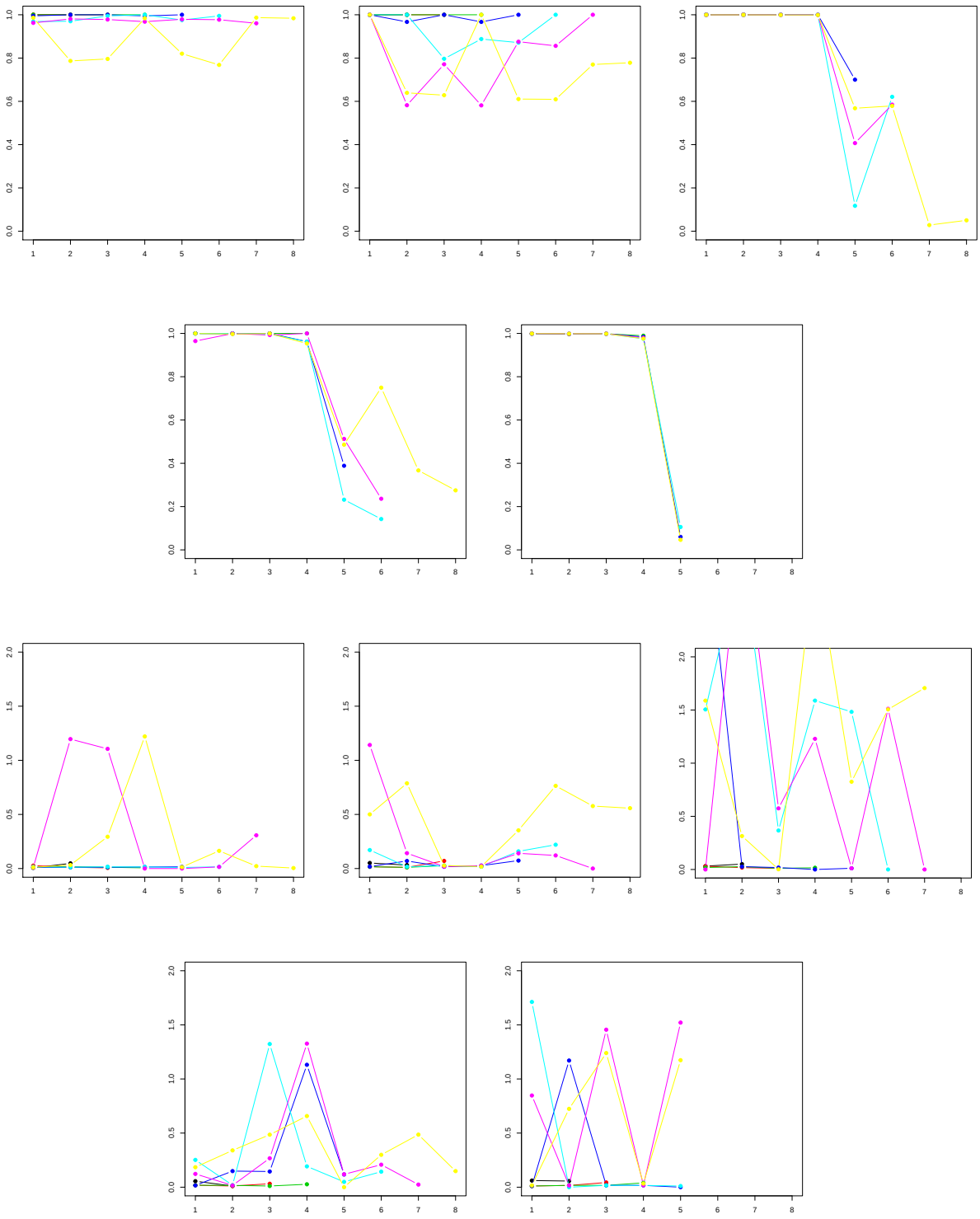


Figure 4.7: Results for $r = 3$. First two rows correspond to values of s_k, last two rows to values of V_k^1. From left to right the clustering algorithms are: k-means, restricted cml, cml, robust cml and *mclust*. For $K = 2, 3, 4, 5, 6, 7, 8$ we have respectively the colours black, red, green, blue, cyan, magenta and yellow.

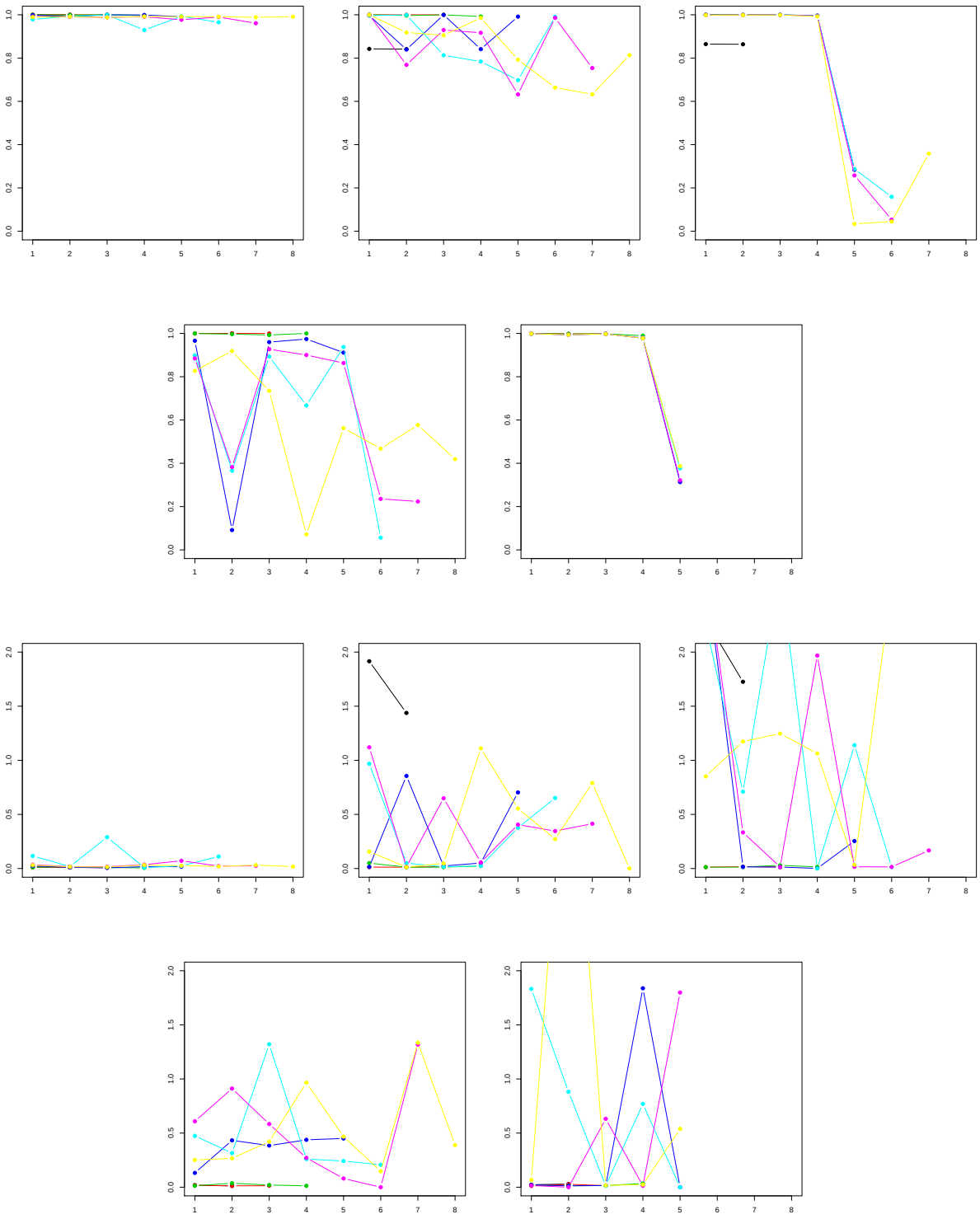
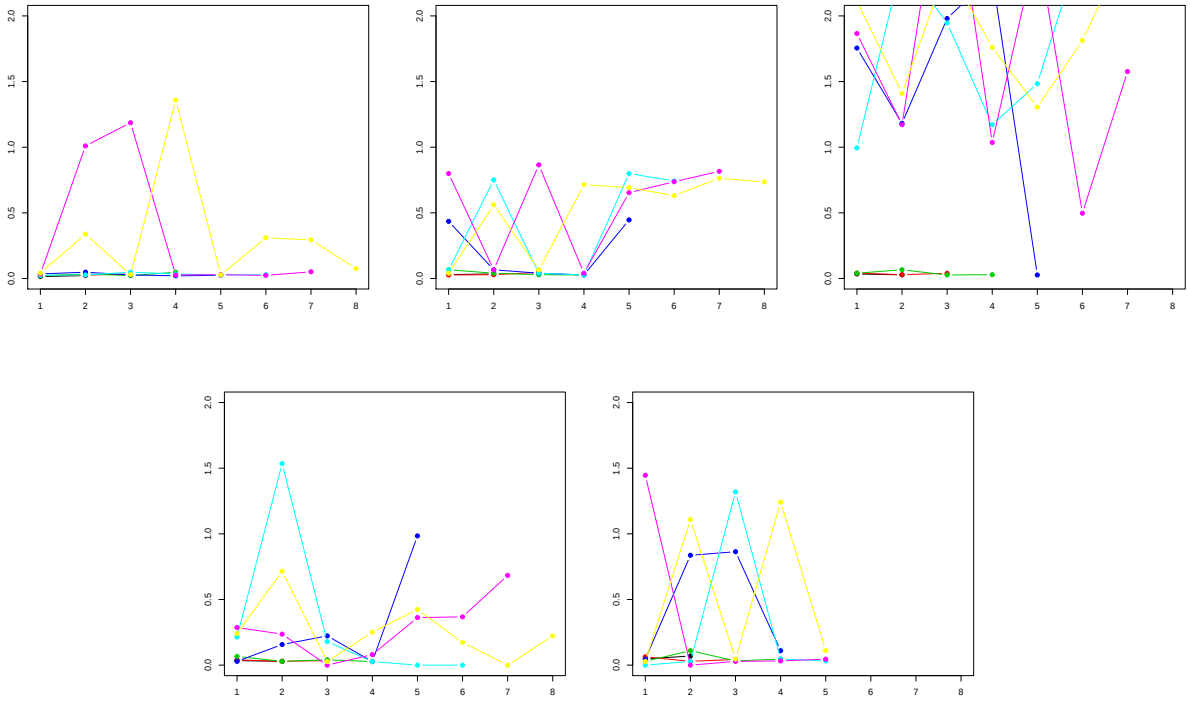


Figure 4.8: Results for $r = 9$ of the values of V_k^1 when drawing independent samples. From left to right the clustering algorithms are: k-means, restricted cml, cml, robust cml and *mclust*. For $K = 2, 3, 4, 5, 6, 7, 8$ we have respectively the colours black, red, green, blue, cyan, magenta and yellow.



To show this last point at work we draw 60 independent samples of 16000 points each with each cluster in the same proportion and coming from the same distribution as in our previous example (fixing $r = 9$). The implementation of our procedure does not change at all, and we give the results in Figure 4.8 and Table 4.1.

Results are very consistent with what we have obtained previously with sub-sampling. Indeed our procedure captures that from a stability point of view $K = 4$ is the best option for all methods except k-means. This can be seen both in the plots of Figure 4.8 or in the values from which we have obtained these plots represented in Table 4.1.

All previous results suggest that cluster-wise stability is an important criteria for choosing the appropriate number of clusters in a model based clustering procedure. We also see that our methodology is very well suited for this kind of clustering and that it provides results competitive with a state of the art procedure.

Table 4.1: Results of V_k^1 for $r = 9$ and 60 independent samples.

K	k-means							
2	0.0150	0.0221						
3	0.0221	0.0254	0.0349					
4	0.0205	0.0349	0.0221	0.0482				
5	0.0349	0.0482	0.0272	0.0205	0.0242			
6	0.0242	0.0270	0.0482	0.0349	0.0272	0.0293		
7	0.0332	1.0099	1.1861	0.0268	0.0291	0.0237	0.0510	
8	0.0412	0.3378	0.0295	1.3586	0.0269	0.3104	0.2944	0.0757
K	restricted cml							
2	0.0303	0.0346						
3	0.0271	0.0291	0.0462					
4	0.0664	0.0405	0.0291	0.0271				
5	0.4351	0.0664	0.0405	0.0271	0.4462			
6	0.0664	0.7512	0.0405	0.0271	0.7993	0.7429		
7	0.7995	0.0664	0.8654	0.0405	0.6539	0.7369	0.8164	
8	0.0405	0.5613	0.0664	0.7143	0.6917	0.6314	0.7637	0.7353
K	cml							
2	0.0346	0.0278						
3	0.0410	0.0291	0.0400					
4	0.0405	0.0664	0.0271	0.0291				
5	1.7551	1.1819	1.9802	2.3092	0.0271			
6	0.9947	2.4048	1.9467	1.1717	1.4838	2.5749		
7	1.8658	1.1715	3.3650	1.0363	2.5498	0.4975	1.5761	
8	2.1112	1.4105	2.3760	1.7603	1.3048	1.8129	2.5374	
K	robust cml							
2	0.0483	0.0691						
3	0.0633	0.0290	0.0430					
4	0.0286	0.1103	0.0319	0.0465				
5	0.0464	0.8366	0.8639	0.1099				
6	0.0000	0.0285	1.3196	0.0466	0.0318			
7	1.4460	0.0000	0.0285	0.0322	0.0454			
8	0.0284	1.1086	0.0461	1.2416	0.1104			
K	mclust							
2	0.0344	0.0291						
3	0.0405	0.0291	0.0346					
4	0.0664	0.0291	0.0405	0.0271				
5	0.0291	0.1569	0.2228	0.0271	0.9842			
6	0.2142	1.5346	0.1793	0.0271	0.0000	0.0000		
7	0.2866	0.2362	0.0000	0.0800	0.3622	0.3677	0.6834	
8	0.2411	0.7151	0.0271	0.2514	0.4239	0.1729	0.0000	0.2217

Conclusion (English)

Throughout Chapter 1 of this thesis we showed that the Kolmogorov distance, the credibility index bounds and the tK-index of fit, provide an intuitive and easy to understand comparison between models. The Kolmogorov distance between a contamination model and a generator gives a straightforward way of comparing accepted or rejected models and, further, allows the use of the other two indexes in the case of rejection. The credibility index bounds provide a summary of which model is closest to the data and give an idea of the region in which the model agrees well with the data. The tK-index of fit provides a single summary that can be widely used and can have attached some informative tolerance region. The procedure we have followed to calculate the normal family tolerance region can be more or less directly extended to other absolutely continuous distributions. We have also provided an efficient algorithm for computing $d_K(F_0, R_\alpha(F_n))$ which makes possible the implementation of all the previous procedures.

With these tools we elaborate on the idea that rejecting a model does not mean that the model is useless. Our testing procedure and asymptotic results allow different applications of this idea. As showed in our toy example in Section 1.6, we can use them to assess how some known generating mechanism produces data compatible with some fixed model when we allow some "small" contamination. In this way we may obtain some useful (hopefully simpler or faster to implement) generators for some range of sample sizes. These tools allow also to compare different data sets, from unknown generators, to a contamination model and rank how well the model agrees with the data.

Future work can be the extension of some of the results to a general dimension. Achieving an algorithm for the computation of a suitable multidimensional version of the trimmed Kolmogorov distance would be a very beautiful and useful result. However both of these problems are difficult due to the nature of the Kolmogorov distance.

Chapter 2 presents a methodology based on clustering of clusterings, supervised classification methods and the Wasserstein distance between probabilities to classify a new sample when a database of classified data has big intrinsic variability. We have built a freely available R package called *optimalFlow* to implement our methodology with the special focus on Flow Cytometry analysis.

Working with real data in collaboration with the Cancer Research Center in Salamanca we have obtained state of the art precision when using our methods to classify new data. Even more, an useful by-product of our method is the obtention of artificial prototypes for groups of similar cytometries. These prototypes can be used as a reference point for comparing different cytometries and hence allow to detect atypical behaviour. Another useful application can be to reduce the size of a databases since only prototypes can be used for classification.

As future work we would like to apply our methodology to a variety of data obtained from fields where conditions are similar to the ones present in Flow Cytometry. This should be a validation of the fact that our methodology is quite general. We are also working in some Bayesian techniques that we think should be competitive in the field of Flow Cytometry but also in general.

In Chapter 3 we have consider the problem of clustering data taking into account an extra variable S which models some sensible information such as age, sex, race. A so-called fair clustering means that the cluster labels should not enable any inference on the values of S, hence promoting clusters which are not homogeneous with respect to the values of S. We have considered an algorithm based on attraction-repulsion dissimilarities, which offers some advantages over previous methods dealing with proportion constrained clustering approaches.

First, our method is flexible in the sense that it is possible to vary the amount of *fairness* we impose on the clusters and to consider the trade-off between fairness and the geometrical structure of the original data. Furthermore, the method is extremely simple to implement, something shared with the methods proposed in [Chierichetti et al., 2017], but not being the case for the other algorithms implementing fair clustering procedures that have been mentioned previously. For instance, we may use any clustering algorithm, since we are not restricted to clustering algorithms that minimize an objective function of some distance. In particular, this allows the use of hierarchical clustering. As a consequence, this method of fair clustering can be extended to the case of non Euclidean data, opening the field of applications for this method. For instance, fairness corrections can be included when clustering probability distributions or some other objects that present geometrical shapes. This can be achieved by replacing the squared euclidean distance $\|X_1 - X_2\|$ in dissimilarities (3.2) - (3.5) with an appropriate distance for the objects involved.

Finally, from a practitioners point of view it should be straightforward to use and easy to program with standard tools as R or Python as provided in `https://github.com/JMLToulouse/FairLearning`.

The field of fair learning and particularly fair clustering is very young and hence there are many opportunities for novel future work. We think that this field will be a bright star in the machine learning universe for many years to come.

In the last chapter of this thesis we have presented some of the results of a work in progress. We had set our minds on achieving the ambitious goal of proving the existence of populational solutions for a popular mixture model based clustering procedure. At the time of writing these lines we still have a long way to go to fully achieve this goal.

However, we think that we have discovered the behaviour of the solutions of the restricted classification maximum likelihood problem in some particular conditions and we have presented it as Conjecture 4.1. We think that a similar behaviour should be generalizable to elliptical distributions but this goes out of the scope of the present work. We have provided several numerical examples that support our conjecture. It is also remarkable how the behaviour of restricted cml and of k-means is very different in the same conditions.

Inspired by the behaviour of restricted cml and based on Wasserstein k-barycenters we have proposed a cluster-wise stability measure tailor made for model based clustering procedures. We have shown that our measure is compatible with other state of the art measures and that it can be used to obtain the optimal number of clusters in a

straightforward fashion.

The challenges that lay ahead are clear, to try to prove Conjecture 4.1 and to apply our methods successfully to meaningful real data examples.

Conclusión (Castellano)

A lo largo del Capítulo 1 de esta tesis hemos mostrado que la distancia de Kolmogorov, las cotas para el índice de credibilidad y el índice de ajuste tK proporcionan una comparación intuitiva y fácil de entender entre modelos. La distancia de Kolmogorov entre un modelo de contaminación y un generador ofrece una forma sencilla de comparar modelos aceptados o rechazados y, además, permite el uso de los otros dos índices en caso de rechazo. Las cotas para el índice de credibilidad proporcionan un resumen en cuanto a que modelo es el más cercano a los datos y dan una idea de la región en la que el modelo concuerda bien con los datos. El índice de ajuste tK proporciona un único valor resumen que puede ser ampliamente utilizado y puede tener asociada alguna región de tolerancia informativa. El procedimiento que hemos seguido para calcular la región de tolerancia para la familia normal puede extenderse más o menos directamente a otras distribuciones absolutamente continuas. También hemos proporcionado un algoritmo eficiente para calcular $d_K(F_0, R_\alpha(F_n))$ que hace posible la aplicación de todos los procedimientos anteriores.

Con estas herramientas profundizamos en la idea de que rechazar un modelo no significa que el modelo sea inútil. Nuestro procedimiento de contraste y nuestros resultados asintóticos permiten diferentes aplicaciones de esta idea. Al igual que en el ejemplo de juguete en la Sección 1.6, podemos usarlos para evaluar cómo algún mecanismo de generación conocido produce datos compatibles con algún modelo fijo cuando permitimos alguna contaminación "pequeña". De esta manera podemos obtener algunos generadores útiles (posiblemente más sencillos o más rápidos de implementar) para algún rango de tamaños de muestra. Estas herramientas permiten comparar diferentes conjuntos de datos, procedentes de generadores desconocidos, a un modelo de contaminación, y proporciona una jerarquía de como de bien el modelo concuerda con los distintos datos.

El trabajo futuro puede consistir en la extensión de algunos de los resultados a una dimensión general. Lograr un algoritmo para el cálculo de una versión adecuada de la distancia de Kolmogorov recortada en dimensión general sería un resultado muy bonito y útil. Sin embargo, ambos problemas son difíciles debido a la naturaleza de la distancia de Kolmogorov.

El capítulo 2 presenta una metodología basada en el clustering de particiones, en métodos de clasificación supervisada y en la distancia de Wasserstein entre probabilidades para clasificar una nueva muestra cuando una base de datos de muestras clasificada tiene una gran variabilidad intrínseca. Hemos construido un paquete de R disponible gratuitamente llamado *optimalFlow* para implementar nuestra metodología con un enfoque especial en el análisis de Citometrías de Flujo.

Trabajando con datos reales en colaboración con el Centro de Investigación del Cáncer de Salamanca hemos obtenido una precisión de última generación a la hora de utilizar

nuestros métodos para clasificar nuevos datos. Además, un subproducto útil de nuestro método es la obtención de prototipos artificiales para grupos de citometrías similares. Estos prototipos pueden ser utilizados como punto de referencia para comparar diferentes citometrías y, por lo tanto, permiten detectar comportamientos atípicos. Otra aplicación útil puede ser reducir el tamaño de las bases de datos, ya que sólo se utilizarían los prototipos para la clasificación.

Como trabajo futuro nos gustaría aplicar nuestra metodología a una variedad de datos obtenidos de campos donde las condiciones son similares a las presentes en la citometría de flujo. Esto debería ser una validación del hecho de que nuestra metodología es bastante general. También estamos trabajando en algunas técnicas Bayesianas que creemos que deberían ser competitivas en el campo de la citometría de flujo, pero también en problemas más generales.

En el Capítulo 3 hemos considerado el problema de agrupar los datos teniendo en cuenta una variable extra S que modela alguna información sensible como la edad, el sexo, la raza. La llamada partición justa requiere que las etiquetas de los clusters no deberían permitir ninguna inferencia sobre los valores de S, promoviendo así agrupaciones que no son homogéneas con respecto a los valores de S. Hemos considerado un algoritmo basado en disimilaridades de atracción-repulsión, que ofrece algunas ventajas sobre métodos anteriores basados en métodos de clustering con restricciones en las proporciones.

En primer lugar, nuestro método es flexible en el sentido de que es posible controlar la cantidad de justicia que imponemos a los clusters y considerar un equilibrio entre la justicia y la estructura geométrica de los datos originales. Además, el método es extremadamente simple de implementar, algo compartido con los métodos propuestos en [Chierichetti et al., 2017], pero no es el caso de los otros algoritmos que implementan procedimientos de clustering justo que se han mencionado anteriormente. Por ejemplo, podemos utilizar cualquier algoritmo de clustering, ya que no estamos limitados a algoritmos de clustering que minimicen una función objetiva de cierta distancia. En particular, esto permite el uso de clustering jerárquico. En consecuencia, este método de agrupación equitativa puede extenderse al caso de datos no viven en el espacio Euclídeo, abriendo el campo de aplicación de este método. Por ejemplo, las correcciones de equidad pueden incluirse al agrupar distribuciones de probabilidad o algunos otros objetos que presentan formas geométricas. Esto puede lograrse reemplazando la distancia Euclídea $\|X_1 - X_2\|$ en las disimilaridades (3.2) - (3.5) por una distancia apropiada para los objetos involucrados.

Por último, desde el punto de vista de los usuarios, debería ser sencillo de usar y fácil de programar con herramientas estándar como R o Python, como se indica en `https://github.com/JMLToulouse/FairLearning`.

El campo del aprendizaje justo y, en particular, del clustering justo es muy joven y, por lo tanto, existen muchas oportunidades para un trabajo futuro novedoso. Creemos que este campo será una estrella brillante en el universo del machine learning durante muchos años.

En el último capítulo de esta tesis hemos presentado algunos de los resultados de un trabajo en progreso. Nos habíamos propuesto alcanzar el ambicioso objetivo de probar la existencia de soluciones poblacionales para un procedimiento de clustering popular basado en un modelo de mezcla. En el momento de escribir estas líneas todavía nos queda un largo camino por recorrer para alcanzar plenamente este objetivo.

Sin embargo, creemos que hemos descubierto el comportamiento de las soluciones del

problema restringido de máxima verosimilitud de clasificación en algunas condiciones particulares y lo hemos presentado como la Conjetura 4.1. Pensamos que un comportamiento similar debería ser generalizable a distribuciones elípticas, pero esto se sale del alcance del presente trabajo. Hemos proporcionado varios ejemplos numéricos que apoyan nuestra conjetura. También es notable cómo el comportamiento del cml restringido y de las k-medias es muy diferente en las mismas condiciones.

Inspirados por el comportamiento de los cml restringidos y basados en los k-baricentros de Wasserstein, hemos propuesto una medida de estabilidad de clústeres hecha a medida para los procedimientos de clústering basados en modelos. Hemos demostrado que nuestra medida es compatible con otras medidas de última generación y que se puede utilizar para obtener el número óptimo de clústers de forma sencilla.

Los retos que tenemos por delante son claros, tratar de probar la Conjetura 4.1 y aplicar nuestros métodos con éxito a ejemplos de datos reales y significativos.

Bibliography

[Aghaeepour et al., 2013] Aghaeepour, N., Finak, G., Hoos, H., Mosmann, T., Brinkman, R., Gottardo, R., and Scheuermann, R. (2013). Critical assessment of automated flow cytometry data analysis techniques. *Nat Methods*, 10:228–238.

[Aghaeepour et al., 2011] Aghaeepour, N., Nikolic, R., Hoos, H., and Brinkman, R. (2011). Rapid cell population identification in flow cytometry data. *Cytometry A*, 79:6–13.

[Agueh and Carlier, 2011] Agueh, M. and Carlier, G. (2011). Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43:904–924.

[Alpaydin, 2014] Alpaydin, E. (2014). *Introduction to Machine Learning*. MIT Press.

[Álvarez-Esteban et al., 2008] Álvarez-Esteban, P., del Barrio, E., Cuesta-Albertos, J., and Matrán, C. (2008). Trimmed comparison of distributions. *J. Amer. Statist. Assoc.*, 103:697–704.

[Álvarez-Esteban et al., 2011] Álvarez-Esteban, P., del Barrio, E., Cuesta-Albertos, J., and Matrán, C. (2011). Uniqueness and approximate computation of optimal incomplete transportation plans. *Annales de l'Institut Henri Poincaré - Probabilités et Statistiques*, 47:358–375.

[Álvarez-Esteban et al., 2012] Álvarez-Esteban, P., del Barrio, E., Cuesta-Albertos, J., and Matrán, C. (2012). Similarity of samples and trimming. *Bernoulli*, 18:606–634.

[Álvarez-Esteban et al., 2016] Álvarez-Esteban, P., del Barrio, E., Cuesta-Albertos, J., and Matrán, C. (2016). A contamination model for approximate stochastic order. *TEST*, 25:751–774.

[Álvarez Esteban et al., 2016] Álvarez Esteban, P., del Barrio, E., Cuesta-Albertos, J., and Matrán, C. (2016). A fixed-point approach to barycenters in wasserstein space. *J Math Anal Appl*, 441:744–762.

[Álvarez Esteban et al., 2018] Álvarez Esteban, P., del Barrio, E., Cuesta-Albertos, J., and Matrán, C. (2018). Wide consensus aggregation in the wasserstein space. application to location-scatter families. *Bernoulli*, 24:3147–3179.

[Azad et al., 2012] Azad, A., Pyne, S., and Pothen, A. (2012). Matching phosphorylation response patterns of antigen-receptor-stimulated t cells via flow cytometry. *BMC Bioinformatics*, 13:S10.

[Backurs et al., 2019] Backurs, A., Indyk, P., Onak, K., Schieber, B., Vakilian, A., and Wagner, T. (2019). Scalable fair clustering. *arXiv:1902.03519*.

[Barron, 1989] Barron, A. (1989). Uniformly powerful goodness of fit tests. *Ann. Statist.*, 17:107–124.

[Bera et al., 2019] Bera, S., Chakrabarty, D., Negahbani, M., and College, D. (2019). Fair algorithms for clustering. *arXiv:1901.02393*.

[Bercea et al., 2018] Bercea, I., Gross, M., Khuller, S., Kumar, A., Rösner, C., Schmidt, D., and Schmidt, M. (2018). On the cost of essentially fair clusterings. *arXiv:1811.10319*.

[Bertsimas and Tsitsiklis, 1997] Bertsimas, D. and Tsitsiklis, J. (1997). *Introduction to Linear Optimization*. Athena Scientific.

[Besse et al., 2018] Besse, P., Castets-Renard, C., Garivier, A., and Loubes, J. (2018). Can everyday AI be ethical. Fairness of Machine Learning Algorithms. *arXiv:1810.01729*.

[Boissard et al., 2015] Boissard, E., Le Gouic, T., Loubes, J.-M., et al. (2015). Distribution's template estimate with wasserstein metrics. *Bernoulli*, 21(2):740–759.

[Bryant and Williamson, 1978] Bryant, P. and Williamson, J. (1978). Asymptotic behaviour of classification maximum likelihood estimates. *Biometrika*, 65:273–281.

[Campello et al., 2013] Campello, R., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. *Advances in Knowledge Discovery and Data Mining. PAKDD 2013*, 7819:160–172.

[Cárcamo et al., 2019] Cárcamo, J., Rodríguez, L., and Cuevas, A. (2019). Directional differentiability for supremum-type functionals: statistical applications. *arXiv:1902.01136*.

[Celeux and Govaert, 1993] Celeux, G. and Govaert, G. (1993). Comparison of the mixture and the classification maximum likelihood in cluster analysis. *Journal of Statistical Computation and Simulation*, 47:127–146.

[Cerioli et al.,2018] Cerioli, A., García-Escudero, L-A., Mayo-Íscar, A. and Riani, M. (2018). Finding the Number of Normal Groups in Model-Based Clustering via Constrained Likelihoods. *Journal of Computational and Graphical Statistics*, 27:404–416.

[Chierichetti et al., 2017] Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. (2017). Fair clustering through fairlets. *Advances in Neural Information Processing Systems*, 30:5029–5037.

[Chouldechova, 2017] Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5:153–163.

[Coen et al., 2010] Coen, M., Ansari, M. H., and Filmore, N. (2010). Comparing clusterings in space. *ICML'10 Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 231–238.

[Cox and Cox, 2000] Cox, T. and Cox, M. (2000). *Multidimensional Scaling*. Chapman and Hall/CRC.

[Cuturi and Doucet, 2014] Cuturi, M. and Doucet, A. (2014). Fast computation of wasserstein barycenters. *PMLR 32*, pages 685–693.

[Davies, 1995] Davies, P. L. (1995). Data features. *Statistica Neerlandica*, 49:185–245.

[Davies, 2018] Davies, P. L. (2018). On p-values. *Statistica Sinica*, 28:2823–2840.

[del Barrio et al., 2019a] del Barrio, E., Cuesta-Albertos, J., Matrán, C., and Mayo-Íscar, A. (2019a). Robust clustering tools based on optimal transportation. *Statistics and Computing*, 29:139–160.

[del Barrio et al., 2018] del Barrio, E., Gamboa, F., Gordaliza, P., and Loubes, J. (2018). Obtaining fairness using optimal transport theory. *arXiv:1806.03195*.

[del Barrio et al., 2019b] del Barrio, E., Inouzhe, H., and Loubes, J.-M. (2019b). Attraction-Repulsion clustering with applications to fairness. *arXiv:1904.05254*.

[del Barrio et al., 2019c] del Barrio, E., Inouzhe, H., Loubes, J.-M., Matrán, C., and Mayo-Íscar, A. (2019c). optimalflow: Optimal-transport approach to flow cytometry gating and population matching. *arXiv:1907.08006*.

[del Barrio et al., 2019d] del Barrio, E., Inouzhe, H., and Matrán, C. (2019d). Box-constrained monotone l_∞-approximations to lipschitz regularizations, with applications to robust testing. *arXiv:1903.08573*.

[del Barrio et al., 2019e] del Barrio, E., Inouzhe, H., and Matrán, C. (2019e). On approximate validation of models: a kolmogorov–smirnov-based approach. *TEST*.

[del Barrio and Matrán, 2013] del Barrio, E. and Matrán, C. (2013). Rates of convergence for partial mass problems. *Probability Theory and Related Fields*, 155:521–542.

[Donoho, 1988] Donoho, D. L. (1988). One sided inference about functionals of a density. *Ann. Statist.*, 16:1390–1420.

[Dost et al., 2011] Dost, B., Wu, C., Su, A., and Bafna, V. (2011). Tclust: a fast method for clustering genome-scale expression data. *IEEE/ACM Trans Comput Biol Bioinform*, 8:808–818.

[Ester et al., 1996] Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231.

[Everitt et al., 2011] Everitt, B., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster Analysis*. Wiley.

[Feldman et al., 2015] Feldman, M., Friedler, S., Moeller, J., Scheidegge, C., and Venkatasubramanian, S. (2015). Certifying and removing disparate impact. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268.

[Ferraro and Giordani, 2013] Ferraro, M. and Giordani, P. (2013). On possibilistic clustering with repulsion constraints for imprecise data. *Information Sciences*, 245:63–75.

[Flury, 1990] Flury, B. (1990). Principal points. *Biometrika*, 77:33–41.

[Fraley and Raftery, 2002] Fraley, C. and Raftery, A. (2002). Model-based clustering, discriminant analysis, and density estimation. *J Am Stat Assoc*, 97:611–631.

[Friedler et al., 2018] Friedler, S., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E., and Roth, D. (2018). A comparative study of fairness-enhancing interventions in machine learning. *arXiv:1802.04422F*.

[Fritz et al., 2013] Fritz, H., García-Escudero, L., and Mayo-Íscar, A. (2013). A fast algorithm for robust constrained clustering. *Computational statistics & data anlysis*, 61:124–136.

[García-Escudero et al., 2008] García-Escudero, L., Gordaliza, A., Matrán, C., and Mayo-Íscar, A. (2008). A general trimming approach to robust cluster analysis. *Ann Statist*, 36:1324–1345.

[Ge and Sealfon, 2012] Ge, Y. and Sealfon, S. (2012). flowpeaks: a fast unsupervised clustering for flow cytometry data via k-means and density peak finding. *Bioinformatics*, 28:2052–2058.

[Genovese and Wasserman, 2004] Genovese, C. and Wasserman, L. (2004). A stochastic process approach to false discovery control. *Ann. Statist.*, 32(3):1035–1061.

[Goetz et al., 2018] Goetz, C., Hammerbeck, C., and Bonnevier, J. (2018). *Flow Cytometry Basics for the Non-Expert*. Springer International Publishing.

[Gordaliza, 1991] Gordaliza, A. (1991). Best approximations to random variables based on trimming procedures. *J. Approx. Theory*, 64(2):162–180.

[Gouic and Loubes, 2017] Gouic, T. L. and Loubes, J. (2017). Existence and consistency of wasserstein barycenters. *Probab Theory Rel*, 168:901–917.

[Grübel, 1988] Grübel, R. (1988). A minimal characterization of the covariance matrix. *Metrika*, 39:49–52.

[Hardt et al., 2016] Hardt, M., Price, E., Srebro, N., et al. (2016). Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323.

[Hennig, 2007] Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics and Data Analysis*, 52:258–271.

[Hennig et al., 2015] Hennig, C., Meila, M., Murtagh, F., and Rocci, R. (2015). *Handbook of Cluster Analysis*. CRC Press.

[Hodges and Lehmann, 1954] Hodges, J. and Lehmann, E. (1954). Testing the approximate validity of statistical hypotheses. *J. R. Statist. Soc. B*, 16(2):261–268.

[Hsiao et al., 2016] Hsiao, C., Liu, M., Stanton, R., McGee, M., Qian, Y., and Scheuermann, R. (2016). Mapping cell populations in flow cytometry data for cross-sample comparison using the friedman-rafsky test statistic as a distance measure. *Cytometry A*, 89:71–88.

[Huber, 1964] Huber, P. J. (1964). Robust estimation of a location parameter. *Ann. Math. Statist.*, 35:73–101.

[Kehrenberg et al., 2018] Kehrenberg, T., Chen, Z., and Quadrianto, N. (2018). Interpretable fairness via target labels in gaussian process models. *arXiv:1810.05598v2*.

[Kuhn, 1995] Kuhn, H. (1995). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97.

[Lance and Williams, 1967] Lance, G. and Williams, W. (1967). A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems. *The Computer Journal*, 9(4):373–380.

[Li et al., 2017] Li, H., Shaham, U., Stanton, K., Yao, Y., Montgomery, R., and Kluger, Y. (2017). Gating mass cytometry data by deep learning. *Bioinformatics*, 33:3423–3430.

[Lindsay and Liu, 2009] Lindsay, B. and Liu, J. (2009). Building and using semiparametric tolerance regions for parametric multinomial models. *Ann. Statist.*, 37:3644–3659.

[Liu and Lindsay, 2009] Liu, J. and Lindsay, B. (2009). Model assessment tools for a model false world. *Stat. Science*, 24:303–318.

[Lo et al., 2008] Lo, K., Brinkman, R., and Gottardo, R. (2008). Automated gating of flow cytometry data via robust model-based clustering. *Cytometry A*, 73:321–332.

[Lo et al., 2009] Lo, K., Hahne, F., Brinkman, R., and Gottardo, R. (2009). flowclust: a bioconductor package for automated gating of flow cytometry data. *BMC Bioinformatics*, 10:145.

[Lum and Johndrow, 2016] Lum, K. and Johndrow, J. (2016). A statistical framework for fair predictive algorithms. *arXiv:1610.08077L*.

[Lux et al., 2018] Lux, M., Brinkman, R., Chauve, C., Laing, A., Lorenc, A., Abeler-Dörner, L., and Hammer, B. (2018). flowlearn: fast and precise identification and quality checking of cell populations in flow cytometry. *Bioinformatics*, 34:2245–2253.

[Massart, 1990] Massart, P. (1990). The tight constant in the dvoretzky-kiefer-wilfowitz inequality. *Ann. Prob.*, 24:303–318.

[McLachlan, 1982] McLachlan, J. (1982). The classification and mixture maximum likelihood approaches to cluster analysis. In Mengersen, K., Robert, C., and Titterington, D., editors, *Mixtures: Estimation and Applications*, volume 2, pages 199–208. North-Holland.

[Meinshausen and Rice, 2006] Meinshausen, N. and Rice, J. (2006). Estimating the proportion of false null hypotheses among a large number of independently tested hypotheses. *Ann. Statist.*, 34(1):373–393.

[Mizuta, 1998] Mizuta, M. (1998). Two principal points of symmetric distributions. In Rizzi, A., Vichi, M., and Bock, H.-H., editors, *Advances in Data Science and Classification*, pages 171–176, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Munk and Czado, 1998] Munk, A. and Czado, C. (1998). Nonparametric validation of similar distributions and assessment of goodness of fit. *J. R. Statist. Soc. B*, 60:223–241.

[Murtagh and Contreras, 2011] Murtagh, F. and Contreras, P. (2011). Algorithms for hierarchical clustering: an overview. *WIREs Data Mining and Knowledge Discovery.*

[Owen, 1995] Owen, A. B. (1995). Nonparametric likelihood confidence bands function for a distribution. *J. Amer. Statist. Assoc.*, 90(430):516–521.

[Pyne et al., 2009] Pyne, S., Hu, X., Wang, K., Rossin, E., Lin, T., Maier, L., Baecher-Allan, C., McLachlan, G., Tamayo, P., Hafler, D., Jager, P. D., and Mesirov, J. (2009). Automated high-dimensional flow cytometric data analysis. *PNAS*, 106:8519–8524.

[Qian et al., 2010] Qian, Y., Wei, C., Lee, F. E.-H., Campbell, J., Halliley, J., Lee, J., Cai, J., Kong, Y., Sadat, E., Thomson, E., Dunn, P., Seegmiller, A., Karandikar, N., Tipton, C., Mosmann, T., Sanz, I., and Scheuermann, R. (2010). Elucidation of seventeen human peripheral blood b cell subsets and quantification of the tetanus response using a density-based method for the automated identification of cell populations in multidimensional flow cytometry data. *Cytometry B Clin Cytom*, 78:S69–82.

[Raghavachari, 1995] Raghavachari, M. (1995). Limiting distributions of kolmogorov-smirnov type statistics under the alternative. *Ann. Statist.*, 1:67–73.

[Rockafellar and Wets, 2009] Rockafellar, R. T. and Wets, R. (2009). *Variational Analysis.* Springer Berlin Heidelberg.

[Rousseeuw, 1987] Rousseeuw, P. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

[Rudas et al., 1994] Rudas, T., Clogg, C., and Lindsay, B. (1994). A new index of fit based on mixture methods for the analysis of contingency tables. *J. R. Statist. Soc. B*, 56(4):623–639.

[Rösner and Schmidt, 2018] Rösner, C. and Schmidt, M. (2018). Privacy preserving clustering with constraints. *45th International Colloquium on Automata, Languages and Programming*.

[Saeys et al., 2016] Saeys, Y., Gassen, S. V., and Lambrecht, B. (2016). Computational flow cytometry: helping to make sense of high-dimensional immunology data. *Nat Rev Immunol*, 16:449–462.

[Schmidt et al., 2018] Schmidt, M., Schwiegelshohn, C., and Sohler, C. (2018). Fair coresets and streaming algorithms for fair k-means clustering. *arXiv:1812.10854*.

[Schölkopf et al., 1998] Schölkopf, B., Smola, A., and Müller, K. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319.

[Scott and Symons, 1978] Scott, A. and Symons, M. (1978). Asymptotic behaviour of classification maximum likelihood estimates. *Clustering methods based on likelihood ratio criteria*, 27:387–397.

[Scrucca et al., 2016] Scrucca, L., Fop, M., Murphy, T., and Raftery, A. (2016). mclust 5: Clustering, classification and density estimation using gaussian finite mixture models. *The R Journal*, 8:289–317.

[Shorack and Wellner, 1986] Shorack, G. R. and Wellner, J. A. (1986). *Empirical Processes with Applications to Statistics*. Classics in Applied Mathematics. SIAM.

[Supreme Court of the United States, 2009] Supreme Court of the United States (2009). Ricci v. DeStefano. *557 U.S. 557*, 174.

[Tarpey et al., 1995] Tarpey, T., Li, L., and Flury, B. (1995). Principal points and self-consistent points of elliptical distributions. *The Annals of Statistics*, 23:103–112.

[Tolan, 2018] Tolan, S. (2018). Fair and unbiased algorithmic decision making: Current state and future challenges. *JRC technical reports (European Comission)*.

[Ubhaya, 1974a] Ubhaya, V. A. (1974a). Isotone optimization. i. *J. Approx. Theory*, 12:146–159.

[Ubhaya, 1974b] Ubhaya, V. A. (1974b). Isotone optimization. ii. *J. Approx. Theory*, 12:315–331.

[Villani, 2009] Villani, C. (2009). *Optimal Transport: Old and New*. Springer.

[Xu et al., 2016] Xu, S., Qiao, X., Zhu, Y., Zhang, Y., Xue, C., and Li, L. (2016). Reviews on determining the number of clusters. *Applied Mathematics & Information Sciences*, 10:1493–1512.

[Zafar et al., 2017] Zafar, M., Valera, I., Rodriguez, M., and Gummadi, K. (2017). Fairness constraints: Mechanisms for fair classification. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 54:962–970.

[Zoppè, 1995] Zoppè, A. (1995). Principal points of univariate continuous distributions. *Statistical and Computing*, 5:127–132.