Attraction-Repulsion clustering with applications to fairness.*

Eustasio del Barrio[1], Hristo Inouzhe[2], and Jean-Michel Loubes[3]

[1,2] *Departamento de Estadística e Investigación Operativa and IMUVA, Universidad de Valladolid, Spain.*

[3] *Université de Toulouse, Institut de Mathématiques de Toulouse, France.*

Abstract

In the framework of fair learning, we consider clustering methods that avoid or limit the influence of a set of protected attributes, S, (race, sex, etc) over the resulting clusters, with the goal of producing a *fair clustering*. For this, we introduce perturbations to the Euclidean distance that take into account S in a way that resembles attraction-repulsion in charged particles in Physics and results in dissimilarities with an easy interpretation. Cluster analysis based on these dissimilarities penalizes homogeneity of the clusters in the attributes S, and leads to an improvement in fairness. We illustrate the use of our procedures with both synthetic and real data.

1 Introduction

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 [12] and references therein for an overview.

Clustering techniques used as 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 [15], [3], [1] or [9] for an overview of such legal issues and mathematical solutions to address them. For instance avoiding discrimination against sensitive characteristics such as sex, race or age can not only be achieved using the naive solution of simply ignoring such protected attribute. Indeed, if the the data at hand reflects a real world bias, machine learning algorithms can pick on this behaviour and emulate it. More precisely, suppose we have 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 clustering on the unprotected data

*Research partially supported by FEDER, Spanish Ministerio de Economía y Competitividad, grant MTM2017-86061-C2-1-P and Junta de Castilla y León, grants VA005P17 and VA002G18.

will result in some biased clusters, therefore, if we classify new instances based on this partition, we will incur in biased decisions.

Through fair clustering our purpose is to avoid or mitigate these situations. Recently, concerns about fairness have received and increasing attention, resulting into two main strategies to address it. 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 ([8], [5]). Another way is to modify the objective functions of the algorithms in a way that eliminates or reduces the unfairness ([21], [13]). In this work we introduce simple and intuitive dissimilarities, with applications to both agglomerative and partitioning cluster algorithms, that aim to reduce homogeneity in the sensitive classes of the groups, but do not impose hard (group proportions) fairness constraints.

In our setting, we have an i.i.d. sample $(X_1, S_1), \ldots, (X_n, S_n) \sim (X, S)$, where $S \in \mathbb{R}^p$ represents the sensitive attributes and $X \in \mathbb{R}^d$ represents the rest of variables of interest. We do not assume that X and S are independent. Assume we know or suspect that this data is biased with respect to the protected class, described by $S_1, \ldots, S_n$. If the biased variables are dominant enough, a clustering on the unprotected data will result in some biased clusters, therefore, if we classify new instances 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, S, are the same in each cluster (hence, the same as the proportions in the whole dataset). Achieving a fair partition is a very demanding computational problem due to the constraints of the proportions in each group (see [2]).

The method we propose, based on *repulsion dissimilarities* that we introduce in Section 2, favours the formation of clusters that are more heterogeneous in the protected variables, since these dissimilarities make bigger the separation between points with the same values of the protected class. The proposed dissimilarities depend on parameters that the practitioner can control and therefore he or she can impose bigger tendency to fairness. The influence of these choices is discussed through some synthetic examples in Section 5.1.1. Repulsion dissimilarities can be combined with some common clustering techniques via an embedding, in particular, with multidimensional scaling (Section 2). Agglomerative hierarchical clustering is well suited for the use of dissimilarities, hence, in Section 3, we show how to adapt our proposals in a computationally efficient way when using this type of clustering. The proposed 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 4 and 5.1.2. Examples of our methodology are given in section 5. We provide a thorough discussion on a synthetic dataset in 5.1.1, while in 5.2 we apply our methods to the Ricci dataset which describes the case of Ricci v. DeStefano of the Supreme Court of the United States [19].

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 tends 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 1 (Repulsion Dissimilarities).

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

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

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

with $u, v \geq 0$;

$$\delta_3\left((X_1,S_1),(X_2,S_2)\right) = \|X_1 - X_2\|^2 - u\|S_1 - S_2\|^2 \tag{3}$$

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

$$\delta_4\left((X_1,S_1),(X_2,S_2)\right) = \left(1 + \text{sign}(S_1'VS_2)u\left(1 - e^{-v(S_1'VS_2)^2}\right)e^{-w\|X_1-X_2\|}\right)\|X_1 - X_2\|. \tag{4}$$

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 [10] repulsion was introduced modifying the objective function, only taking into account distances between points, to maintain centers of clusters separated). Dissimilarities (1) to (4) 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, therefore, the practitioner with the ability to understand and control the degree of perturbation introduced. Dissimilarity (1) 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) 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 (2) 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 (4) is also a multiplicative perturbation of the Euclidean distance. However, it has a very different behaviour with respect to (1)-(3), 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 (1), while with u we control the amount of maximum perturbation as in (2). 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 1. Let us take $S_1, S_2 \in \{-1, 1\}$. For dissimilarity (1) 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 (2) 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), 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 (4), 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 (1) and (4). 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.

These dissimilarities can then be used directly in some agglomerative hierarchical clustering method, as described in Section 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 [7]) 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 [4], 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 5 for some synthetic and real datasets.

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 [16] 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 [6].

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 [7]) 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 [16]).
However, these last two methods may present some undesirable features (the related dendrograms may present *reversals* that make the interpretation harder, see, e.g., [7]) 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

$$\delta_{W,l}(i,j) = \tfrac{n_i n_j}{n_i+n_j}\delta_l((\tfrac{1}{n_i}\textstyle\sum_{i=1}^{n_i} X_i, \tfrac{1}{n_i}\sum_{i=1}^{n_i} S_i), (\tfrac{1}{n_j}\sum_{j=1}^{n_j} Y_j, \tfrac{1}{n_j}\sum_{j=1}^{n_j} T_j)) \tag{5}$$

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

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 [14], [16] or [7]) 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

$$d_x(i,j) = \big\| \tfrac{1}{n_i} \textstyle\sum_{i=1}^{n_i} X_i - \tfrac{1}{n_j} \sum_{j=1}^{n_j} Y_j \big\|. \tag{6}$$

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

$$d_s(i,j) = \big\| \tfrac{1}{n_i} \textstyle\sum_{i=1}^{n_i} S_i - \tfrac{1}{n_j} \sum_{j=1}^{n_j} T_j \big\|. \tag{7}$$

Proposition 1. *For $\delta_{W,l}$ as in (5), $d_x(i,j)$ as in (6) and $d_s(i,j)$ as in (7) 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)

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

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

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

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

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

$$d^2_{W,x}(i \cup j, k) = \tfrac{n_i+n_k}{n_i+n_j+n_k} d^2_{W,x}(i,k) + \tfrac{n_j+n_k}{n_i+n_j+n_k} d^2_{W,x}(j,k) - \tfrac{n_k}{n_i+n_j+n_k} d^2_{W,x}(i,j) \tag{8}$$

(see, e.g., [7]). The update formulas *ii)* and *iii)* are obtained similarly. We omit details.

□

From Proposition 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 (8). 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*

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

$$d^2_\kappa(x,y) = \kappa(x,x) + \kappa(y,y) - 2\kappa(x,y). \tag{9}$$

Details of this algorithm can be found in [18].

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

$$d_\kappa^2((X_1, S_1), (X_2, S_1)) > d_\kappa^2((X_1, S_1), (X_2, S_2)), \tag{10}$$

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 (10) 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 (1) to (4). 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

$$\delta_{\kappa,1}\left((X_1, S_1), (X_2, S_2)\right) = 1'U1 + S_1'VS_2 + d_\kappa(X_1, X_2)^2, \tag{11}$$

with similar changes for the other dissimilarities. Then we can use an embedding (MDS the simplest choice) as in Section 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 5.1.2.

5 Applications

5.1 Synthetic data

5.1.1 General example

We generate 50 points from four distributions,

$$\mu_1 \sim N((-1, 0.5), \mathrm{diag}(0.25, 0.25)), \mu_2 \sim N((-1, -0.5), \mathrm{diag}(0.25, 0.25));$$

$$\mu_3 \sim N((1, 0.5), \text{diag}(0.25, 0.25)), \mu_4 \sim N((1, -0.5), \text{diag}(0.25, 0.25)),$$

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 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 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 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 (1), 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 (2), where we fix $v = 20$ and set $u = 0, 0.5, 1, \ldots, 5$. In the last row we work with dissimilarity (4) 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), since in this example it gives results very similar to dissimilarity (1). 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 [17]). The average silhouette index of a group is the average of the silhouette indexes of the members of the group.

What we see top-left and middle-left in Figure 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 (1) and (2) 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 with dissimilarity (4). 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 interest 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 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 (1) we take $V = 1.32$, for (2) $u = 1$ and for (4) 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 in the original space.

For the same values as the previous paragraph we present Table 1, where we look for 2,3 and 4 clusters with MDS and k-means, but also using the approximation-free complete linkage

Figure 1: Top row: dissimilarity (1). Middle row: dissimilarity (2). Bottom row: dissimilarity (4). 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.

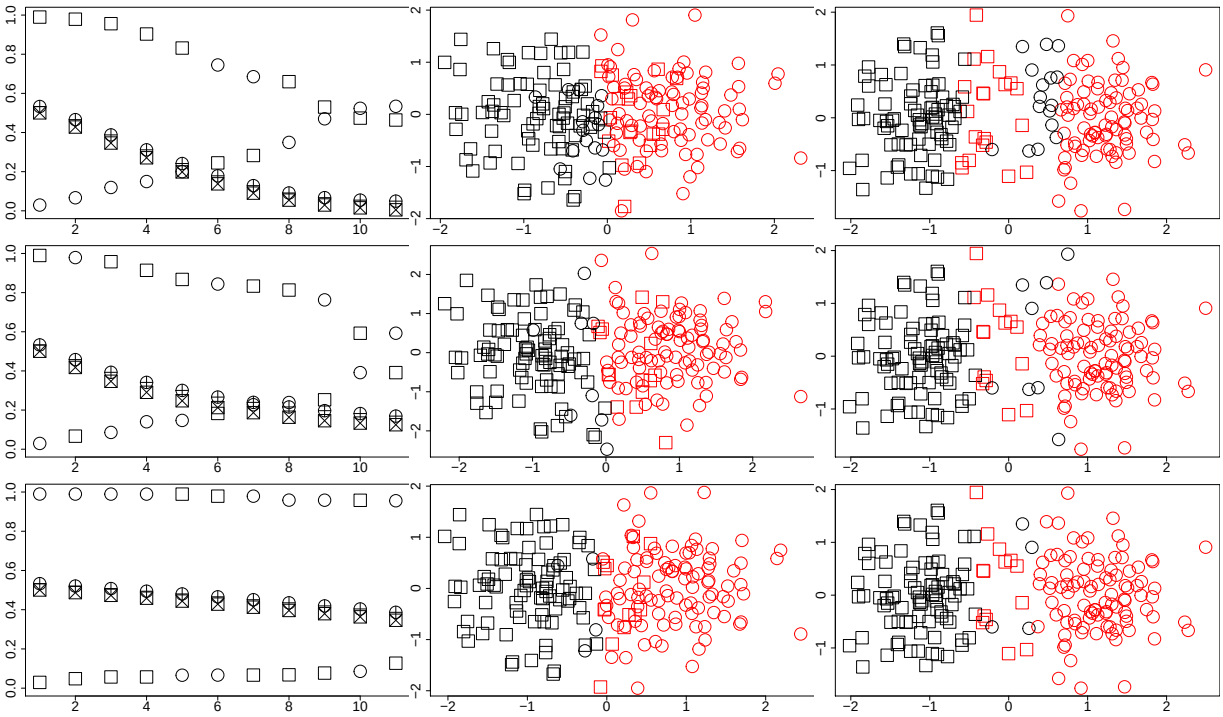


Table 1: Proportion of class S = 1 in every group in different clustering procedures.

			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

Table 2: Effect of varying the intensity u of the local dissimilarity (12), 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

hierarchical clustering and our Ward's-like method. We see that there is an improvement in heterogeneity, regardless of the method used. However, the more clusters we want the smaller the improvement. This is associated with our small perturbation of the geometry, if we want more fairness for bigger number of clusters we have to use more intense perturbations.

5.1.2 Kernel trick example

Let us explore the adaptation of the kernel trick explained in Section 4. We consider the data in the top-left image of Figure 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 [11]). 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 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 $\kappa(x,y) = x_1^2y_1^2 + x_2^2y_2^2$, which corresponds to a transformation $\phi((x_1,x_2)) = (x_1^2, x_2^2)$. The data in the transformed space is depicted in the top-right of Figure 2. Our adaptation to the kernel trick uses d_κ as defined in (9) and dissimilarity (4) in the form

$$\delta_{\kappa,4}((X_1,S_1),(X_2,S_2)) = \big(1+\text{sign}(S_1'VS_2)u(1-e^{-v(S_1'VS_2)^2})e^{-wd_\kappa(X_1,X_2)}\big)d_\kappa(X_1,X_2), \quad (12)$$

for X_1, X_2 in the original two dimensional space, as described in Section 4.

Taking into account the discussion at the end of Section 2 we use dissimilarity (12) 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 (12) to calculate the dissimilarity matrix Δ and use MDS, essentially, we will be in the space depicted top-right on Figure 2. Looking for two clusters with tclust, allowing groups with different sizes, we get the result depicted middle-right in Figure 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

Figure 2: Top row: data in the original space (left) and after transformation ϕ (right). Middle row: k-means in the original space (left) and clusters obtained by tclust 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).

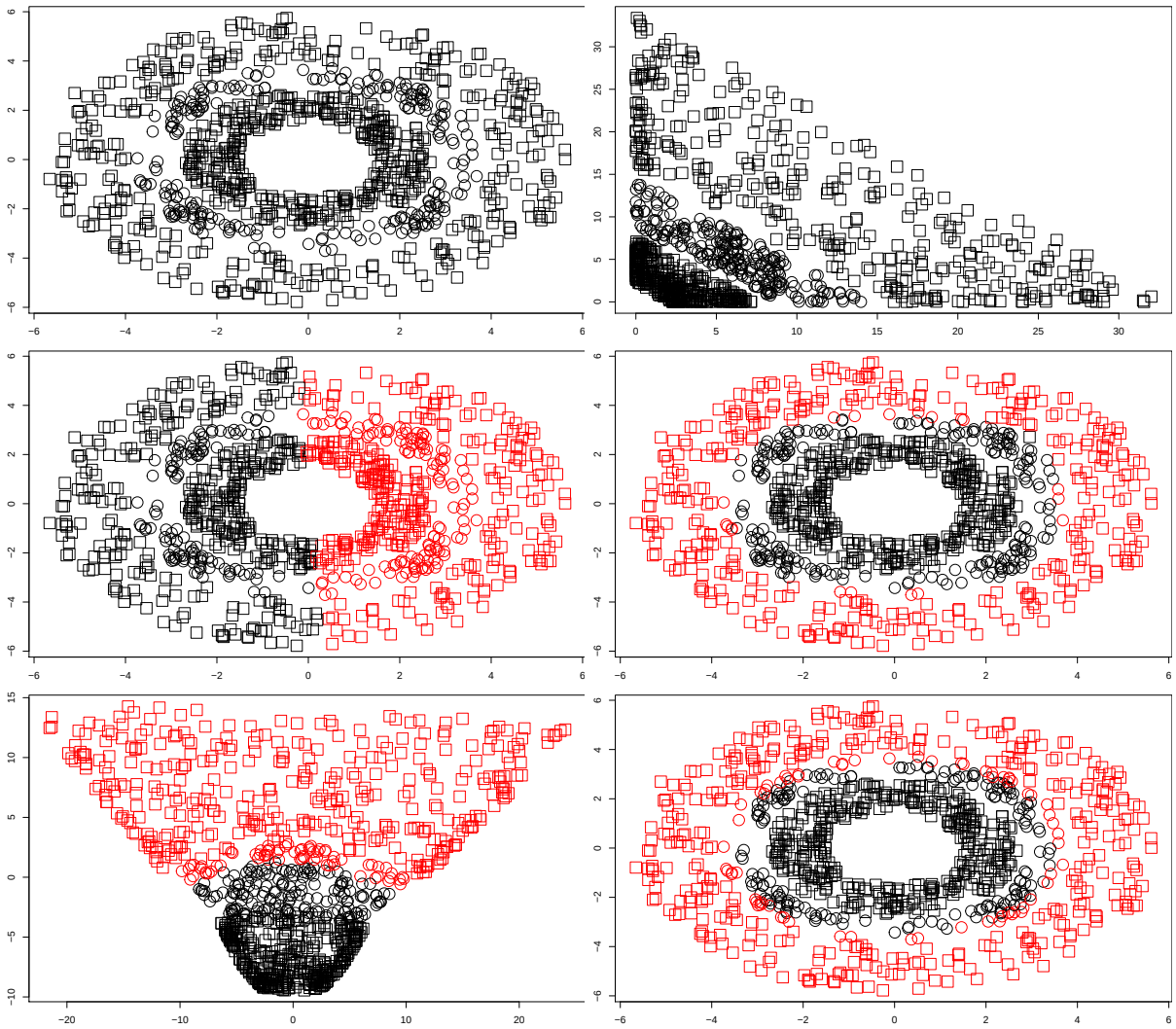
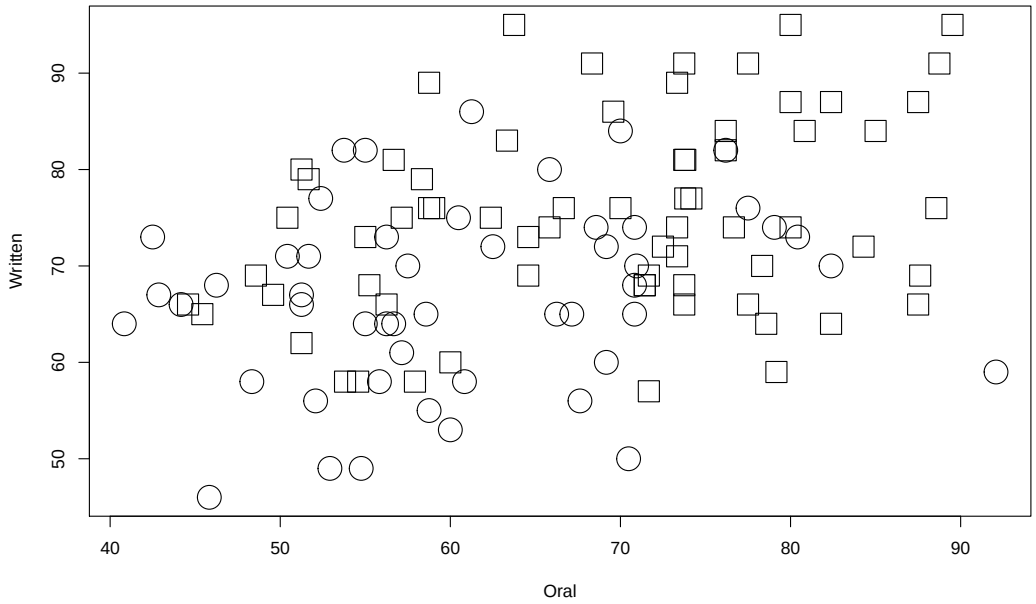


Figure 3: Scores of white individuals in squares and of black and hispanic individuals in circles.



columns 2 and 3 of Table 2 (ideally they should be close to 0.754). In order to gain diversity in what is referred as cluster 2 in Table 2 (in red in the plots), we vary the intensity u of our local dissimilarity, with the other parameters set as indicated in Table 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 2. It is clear that we have been able to capture the geometry of the groups and to produce relatively fair clusters.

5.2 A real data example

We analyse 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, where the combined score of the exam, which was a threshold for a promotion, was the question at issue.

We codify white individuals as $S = 1$ and black and hispanic individuals as $S = -1$. A representation of the data can be seen in Figure 3. From it, it is clear that the region delimited by oral scores higher than 72 is heavily dominated by white individuals. We may see this as a bias in this variable. In any case, a standard clustering method on this data may result in groups that are more homogeneous with respect to race than the data as a whole. We stress that the proportion of black and hispanic people in the data set is approximately 0.42.

In Table 3 we test our methods which look to decrease homogeneity, and possibly increase fairness. In this sense, we want to have groups that have a proportion of black and hispanic members close to 0.42. We will use both transforming the data via MDS and applying k-means, the alternative straightforward use of complete linkage hierarchical clustering and also

Table 3: Proportion of class S = -1 in every group in different clustering procedures. For δ_1 we fix $U = 0, V = 500$; for δ_2 we fix $u = 3.125, v = 10$; for δ_3, $u = 333$ and for δ_4, $V = 1$, $u = 0.99$, $v = 10$, $w = 1$.

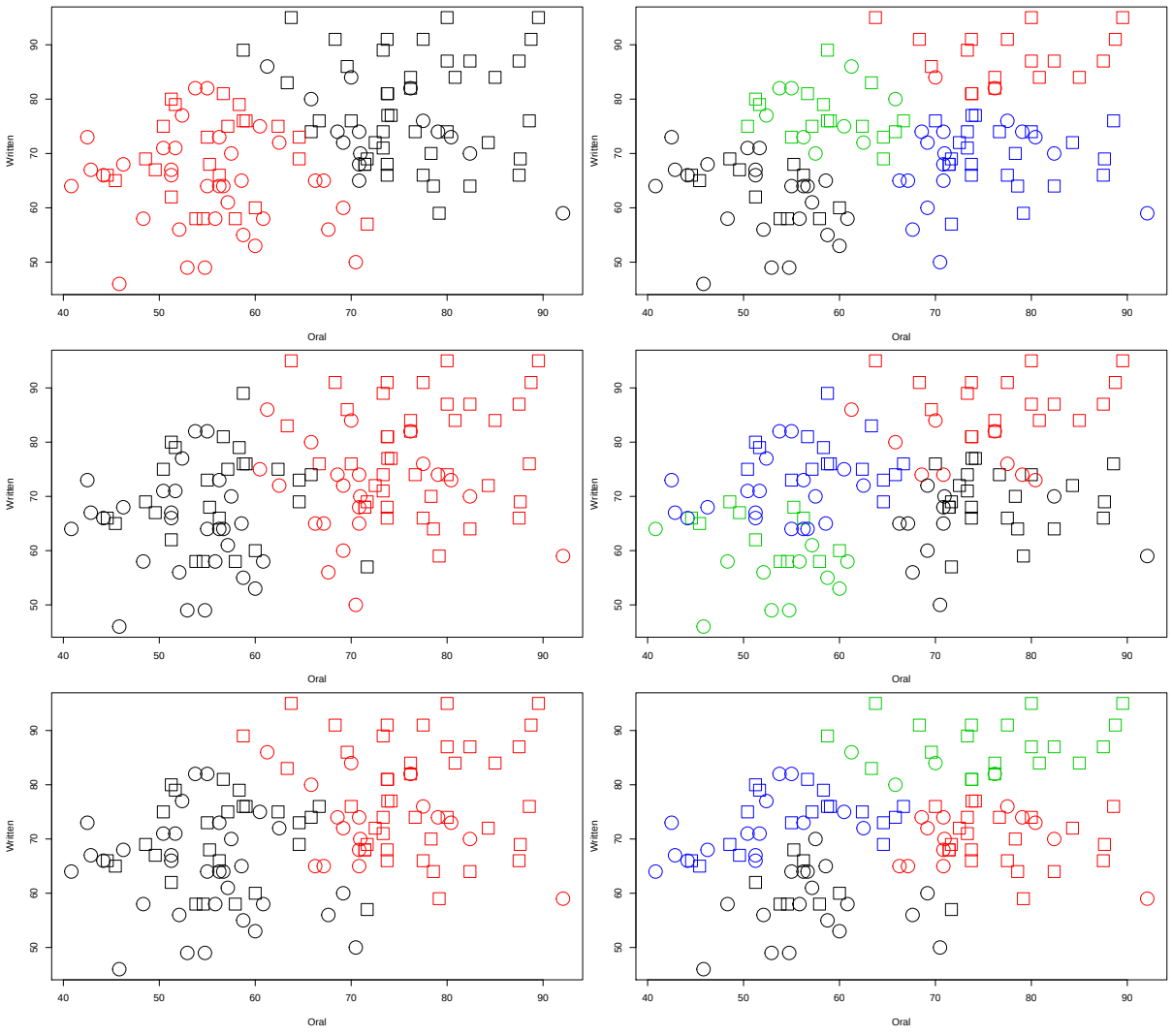
			Proportion of black and hispanic people in the group								
			K = 2		K = 3			K = 4			
k -means	Unperturbed		0.59	0.25	0.59	0.37	0.17	0.68	0.36	0.10	0.41
	MDS	δ_1	0.34	0.52	0.35	0.52	0.35	0.54	0.50	0.33	0.32
		δ_2	0.54	0.32	0.39	0.31	0.52	0.41	0.50	0.56	0.18
		δ_3	0.35	0.51	0.51	0.39	0.32	0.33	0.54	0.50	0.32
		δ_4	0.25	0.59	0.59	0.17	0.37	0.68	0.10	0.41	0.36
Complete Linkage	Unperturbed		0.29	0.60	0.29	0.29	0.60	0.00	0.29	0.37	0.60
	δ_1		0.27	0.48	0.27	0.44	0.55	0.00	0.42	0.44	0.55
	δ_2		0.30	0.59	0.30	0.27	0.59	0.30	0.27	0.50	0.70
	δ_3		0.23	0.49	0.23	0.48	0.51	0.23	0.43	0.51	0.50
	δ_4		0.30	0.62	0.22	0.43	0.62	0.00	0.43	0.29	0.62
Ward's Method	Unperturbed		0.56	0.29	0.37	0.17	0.56	0.69	0.37	0.17	0.45
	δ_1		0.55	0.33	0.13	0.38	0.55	0.50	0.60	0.13	0.38
	δ_2		0.58	0.31	0.38	0.26	0.58	0.69	0.54	0.38	0.26
	δ_3		0.52	0.34	0.18	0.43	0.52	0.18	0.59	0.47	0.43

Ward's method. The appropriate parameters for the dissimilarities are chosen to give a good performance and are specified in Table 3.

As we expected, applying clustering procedures to the original (unperturbed) data gives clusters that are significantly more homogeneous than the whole data set with respect to the protected class. On the other hand, using the suggested dissimilarities, we see improvements in the heterogeneity of the groups regarding the sensitive variable. Both additive dissimilarities (1) and (3) perform well in conjunction with MDS and k-means, giving proportions closer to 0.42 for 2,3 and 4 clusters (rows 2 and 3 in Table 3). In the hierarchical setting, it seems that dissimilarity (2) gives a nice performance. The local dissimilarity (4) seems to have little effect, which can be interpreted as a need of non-local effects to be able to affect the group formation.

In Figure 4 we compare the solutions of k-means for 2 and 4 clusters in the original data and in the data transformed via MDS and dissimilarity (1), which we saw gives a good performance. We also represent the solution given by Ward's method and dissimilarity (2). We see that the clusterings are similar, but our perturbation of the Euclidean distance that takes into account the composition of the groups is able to induce change in the borders between groups, making the clusters more heterogeneous in the protected class.

Figure 4: Top row: k-means for 2 and 4 clusters in the unperturbed (original) data. Middle row: k-means for 2 and 4 clusters in the MDS setting with δ_1. Bottom row: Ward's method for 2 and 4 clusters with δ_2. Circles represent not white individuals; squares represent white individuals.



References

[1] Besse, Philippe and Castets-Renard, Celine and Garivier, Aurelien and Loubes, Jean-Michel, (2018). Can everyday AI be ethical. Fairness of Machine Learning Algorithms, *arXiv:1810.01729*.

[2] Chierichetti, F. Kumar, R. Lattanzi, S. and Vassilvitskii, S. (2017) Fair clustering through fairlets. *NIPS*.

[3] Chouldechova, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5:153–163, 2017.

[4] Cox,T. F. and Cox, M. A. A. (2000) *Multidimensional Scaling.*

[5] del Barrio, E. Gamboa, F. Gordaliza, P. and Loubes, J-M.(2018) Obtaining fairness using optimal transport theory. *arXiv:1806.03195*.

[6] Ester, M. Kriegel, H-P. Sander, J. and Xu, X. (1996) A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *AAAI*.

[7] Everitt B.S., Landau, S., Leese, M. and Stahl, D. (2011). *Cluster Analysis, 5th Edition.* Wiley.

[8] Feldman, M. Friedler, S. A. Moeller, J. Scheidegger, C. and Venkatasubramanian, S. (2015) Certifying and removing disparate impact. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[9] Friedler, S. A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E. P., and Roth, D. (2018) A comparative study of fairness-enhancing interventions in machine learning. *arXiv:1802.04422F*.

[10] Ferraro, M. B. and Giordani, P. (2013) On possibilistic clustering with repulsion constraints for imprecise data. *Information Sciences*.

[11] Fritz, H. García-Escudero, LA. and Mayo-Íscar, A. (2012) tclust: An R Package for a Trimming Approach to Cluster Analysis. *Journal of statistical software*.

[12] Hennig, C., Meila, M., Murtagh, F. and Rocci, R. (2015). *Handbook of Cluster Analysis.* CRC Press.

[13] Kehrenberg, T. Chen, Z. and Quadrianto, N. (2018) Interpretable Fairness via Target Labels in Gaussian Process Models. *arXiv:1810.05598v2*.

[14] Lance, G. and Williams, W. (1967) A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems. *The Computer Journal*.

[15] Lum, K. and Johndrow, J. (2016) A statistical framework for fair predictive algorithms. *arXiv:1610.08077L*.

[16] Murtagh, F. and Contreras, P. (2011) Algorithms for hierarchical clustering: an overview. *WIREs Data Mining and Knowledge Discovery*.

[17] Rousseeuw, P. (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*.

[18] Schölkopf, B. Smola, A. and Müller, K-R. (1998) Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*.

[19] Supreme Court of the United States. (2009) Ricci v. DeStefano. 557 U.S. 557, 174.

[20] Woodworth, B. Gunasekar, S. Ohannessian, M. I. and Srebro, N. (2017) Learning Non-Discriminatory Predictors. *arXiv:1702.06081v3* .

[21] Zafar, M. B. Valera, I. Rodriguez, M. G. and Gummadi, K. P. (2017) Fairness Constraints: Mechanisms for Fair Classification. *AISTATS*.