Exploring Fairness in Density-Based Clustering

Anneka Thiesson, 202005605

Supervisors: Ira Assent & Anna Beer

AARHUS UNIVERSITY

Bachelor's Thesis, Data Science

June 15th, 2023

Abstract

Machine learning (ML) models are increasingly relied upon due to their perceived objectivity and potential for improved decision-making. However, it has been discovered that ML models can exhibit biases, resulting in unfair or discriminatory outcomes. Addressing bias in ML models is crucial to mitigate adverse consequences. While fairness in clustering has gained attention to address this bias, fairness in density-based clustering remains under-explored. The project aims to fill this gap by integrating a novel density-connectivity metric and fair spectral clustering methods to develop a fair clustering algorithm rooted in density-based clustering. We evaluate fairness using established metrics and compare the fairness of the original clustering algorithms with the adapted fairness-aware algorithms. This research contributes to the advancement of fairness in density-based clustering, with potential applications in various domains, including social network analysis and image segmentation.

Contents

Chapter 1

Introduction

Machine learning (ML) models have gained popularity due to the perception that ML algorithms can potentially optimize and improve human decision-making processes. They have the ability to crunch through enormous amounts of data and factor in countless considerations that humans would never be able to process independently. However, it has been revealed that there is no guarantee that these ML models are inherently objective, despite their definitude. Extensive research has shown that machine learning models can exhibit bias, which refers to the presence of unfair or discriminatory behavior in their outputs. Addressing this bias in ML models is crucial, especially when decisions made by biased machine learning models can have significant consequences that can influence individuals, groups, or broader societal outcomes.

Numerous studies in the field of ML fairness have researched various factors contributing to the emergence of bias in machine learning [1] [2]. One significant source of bias lies within the datasets themselves. Bias can be embedded into data due to biased device measurement, flawed reporting, or human bias (also known as prejudice bias). Machine learning algorithms, designed to learn from existing patterns within the data, tend to preserve these biases. Examples of unfair ML algorithms deployed in the real world include the re-offense risk assessment tool COMPAS, which exhibited unfair discrimination against black individuals in the federal US criminal justice system [3], and Amazon's AI hiring and recruitment system, which demonstrated bias against women [4]. Biases can also arise from missing data. This includes missing values or selection biases, resulting in datasets failing to represent the target population accurately. Algorithmic bias is another concern, as biases can stem from algorithmic objectives that prioritize minimizing overall aggregated prediction errors. By prioritizing this objective, ML algorithms may inadvertently favor majority groups over minority groups.

Clustering algorithms are unsupervised ML algorithms and have become fundamental data exploration tools that provide valuable insights and facilitate an increasing number of ML processes that affect the real world. Clustering algorithms group similar data together based on their shared properties, and via this method, can seg-

ment data into meaningful subsets, known as clusters, and in this way identify patterns that might not be immediately apparent. Clustering is particularly useful in domains where the data lacks predefined classes or labels assigned to each point, as it enables researchers to uncover structures and relationships without prior knowledge. In recent years, fairness in clustering has gained traction within the ML community, spearheaded by Chierichetti et al.'s research on fair clustering in 2017 [5]. Fair clustering focuses on achieving fairness in clustering outcomes by adhering to some pre-defined fairness criteria, such as ensuring equal representation of sensitive groups within the identified clusters.

Fairness in density-based clustering [6] has received limited attention compared to other clustering techniques. This can be attributed to the challenges in extending fairness principles to density-based clustering algorithms, which rely heavily on the procedural definition of the clustering technique [7]. However, density-based clustering algorithms are commonly used in various domains, including social network analysis and image segmentation [8]. Hence, ensuring fairness in their clustering outcomes would be beneficial.

The purpose of this research is to explore fairness in density-based clustering. We do this by utilizing a recently-developed density-connectivity metric [7] that links density-based clustering and spectral clustering. As there is research regarding fairness in spectral clustering, this study aims to integrate the two techniques to produce a fair clustering algorithm grounded in density-based clustering. Furthermore, we use existing metrics for fairness evaluation to compare the fairness of the vanilla clustering algorithms and the algorithms adapted to address fairness concerns. Our code recreates every experiment below: GitHub Repository.

Chapter 2

Related Work

Fairness in clustering has emerged as an important research area to address potential biases and discrimination in clustering algorithms. To address fairness concerns, researchers have proposed different techniques [9]. The research into the design of fair algorithms is mainly inspired by the notion of disparate impact [10]. The developed fairness approaches can broadly be categorized into pre-processing, in-processing, and post-processing methods. Pre-processing methods [5] [11] [12] focus on modifying the input data to reduce biases before applying vanilla clustering algorithms. In-processing methods [13] [14] [15] integrate fairness notions and considerations directly into the clustering algorithm, ensuring fair representation and group treatment during the clustering process. Post-processing methods [16] [17], on the other hand, aim to modify the vanilla clustering algorithm's output to achieve fairness.

Chierichetti et al. [5] lays the foundation of incorporating fairness considerations in clustering algorithms using disparate impact [10]. Chierichetti et al.'s focus was on k-median and k-center clustering methods. Specifically, they addressed the binary sensitive attribute scenario, where only two demographic groups exist. To ensure a predetermined level of balance within clusters, they proposed approximation algorithms for finding a clustering solution with the minimum k-median or k-center cost. Rosner and Schmidt [18] extended [5] to allow for multiple protected groups. Chierichetti et al. [5] also proposed the group-level and agnostic fairness notion of balance for a scenario with a binary sensitive attribute. This was later generalized by Bera et al. [17], and has since been used as the fairness metric for much research on fair clustering [11] [9] [12] [13].

Kleindessner et al. [13] uses the balance fairness notion and applies theory from constrained spectral clustering to incorporate prior knowledge about the target clustering. There is a fair amount of literature on constrained spectral clustering [19] [20] [21] [22]. Wang et al. [23] addresses the scalability limitations of the fairness algorithm proposed by [13] by reducing the computation complexity of the algorithm.

The focus of this report, fairness in density-based clustering, has received relatively limited attention in the research community, with sparse contributions focusing on this

specific area.

Chapter 3

Methodology

This section outlines the methodology employed in this project. Firstly, we provide a concise overview of cluster analysis and introduce the two relevant clustering methods. Subsequently, we delve into the concept of fairness within clustering, specifically focusing on the notion of balance and its significance. Additionally, we discuss a novel metric developed for density-based clustering and a constrained spectral clustering algorithm for fairness in clustering, and then propose an integration of the two for fair clustering grounded in density-based clustering.

3.1 Cluster Analysis

Cluster analysis is an unsupervised ML technique that involves grouping similar data points together based on inherent characteristics or similarities. Thus, the task of clustering is to partition a given m-dimensional dataset of n data points $X \in \mathbb{R}^{m \times n}$ into k subsets, also known as clusters $c \in C$. Unlike supervised learning tasks, there are no labels present for the data, so the dataset itself is used for both training and testing.

3.1.1 Density-based clustering: DBSCAN

Density-based clustering is a technique used to identify clusters in data based on the density of data points. The main intuition behind density-based clustering is that clusters are areas of higher density than the surrounding regions of the dataset.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [6] is a popular density-based clustering algorithm. This approach identifies data clusters of arbitrary shapes and will therefore apply to datasets with irregular distributions.

In DBSCAN, a *core point* is defined as a data point with a sufficient number of neighboring points within a specified distance. The sufficient number of neighboring points and the specified distance are controlled by two parameters minPts and ϵ, respectively. These core points are considered the starting points for cluster formation. *Density-reachability* is a notion that describes the connection between data points

based on their density. A point is considered density-reachable from another point if it is connected with a core point directly within a neighborhood determined by ϵ or indirectly through a chain of core points. If a point is not a core point itself but can be reached by a following path of core points, it is density-reachable.

The DBSCAN algorithm begins by selecting an arbitrary core point and explores its neighborhood determined by ϵ to find all density-reachable points. This process continues until no more density-reachable points are found, and the cluster is considered complete. Points that are neither core points nor density-reachable are marked as noise. DBSCAN repeats this process for other unvisited core points or noise points until all points have been assigned to a cluster or labeled as noise.

3.1.2 Spectral clustering

Spectral clustering [24] is a non-parametric clustering technique based on spectral graph theory, where the approach identifies communities of vertices in a graph based on the edges connecting them. It clusters graph data and can be extended to cluster non-graph data by constructing an affinity matrix that captures the similarity between data points. The primary hyperparameter in spectral clustering is the number of clusters k into which the data will be partitioned. Considering the intuition of clustering, where we want to separate points in different groups according to their similarities, we want to find a partition so that the connectivity between groups is small. Such a partition responds to what is known as a cut in a graph. A good cut is often evaluated based on the number and significance of the edges in the graph that it crosses.

By cutting fewer and less significant edges, a cut can effectively separate distinct clusters, indicating a stronger distinction between them. Ergo, a good cut partitions the graph in such a way that it maximizes the separation between clusters. In graph theory, two important types of cuts are the minimum cut $MinCut$ and the ratio cut $RatioCut$.

A MinCut refers to the cut that minimizes the number of edges cutting it, removing the fewest edges or minimal weighted edges necessary to disconnect the graph. Formally, given a graph G with nodes V and edges E, a MinCut refers to the cut that partitions the graph's nodes into two subsets that minimize the number of edges crossing the cut.

A RatioCut cut aims to partition a graph into two clusters while optimizing for a specific ratio. Given a weight function $w(e)$ that assigns a weight to each edge e in E,

RatioCut seeks to find a cut that minimizes the ratio between the number of edges cutting the cut and the total number of edges in the cut. A good RatioCut then partitions the graph so that there is a small proportion of edges crossing the cut compared to the total number of edges in the cut.

Spectral clustering is derived from the notion of a cut on a graph. It uses the spectral properties obtained from a graph representation of the data. The algorithm begins by constructing a similarity or a weighted adjacency matrix to capture the relationship between data points. The Laplacian matrix is derived from this matrix representation, providing information about the graph's connectivity and other spectral properties. By an eigendecomposition of the Laplacian matrix, the data is transformed into a new representation, where the eigenvalues represent the properties of the Laplacian matrix, and the corresponding eigenvectors provide a low-dimensional feature space of the data. Typically, the k smallest eigenvectors (eigenpairs) are selected since they capture the most informative variations in the data and correspond to the underlying cluster structure. The selected eigenvectors form a new feature space in which the data points are projected. The transformed data can then be clustered using simple, traditional clustering algorithms such as k-means, with the eigenvectors as input to the algorithms.

3.2 Fairness in Clustering

Fairness in clustering aims to address and mitigate biases and disparities that may arise in the clustering process. Fairness in clustering involves ensuring equitable treatment and representation for different groups or individuals within the clustering outcomes.

One key challenge in achieving fairness in clustering lies in the definition of fairness itself. The preference for a specific definition of fairness in clustering can vary depending on the scenario and context requirements. For example, in some situations, individual fairness may be prioritized to ensure similar treatment for similar individuals. In contrast, group fairness may be emphasized in other cases to achieve equitable outcomes for different demographic groups.

Demographic parity (also known as statistical parity or group fairness) notions are derived from the Disparate Impact (DI) doctrine [10] [25] and prohibit discrimination towards any specific demographic group in the algorithm's output predictions. It em-

phasizes that clusters should be formed to guarantee fairness across groups, regardless of individual characteristics. Demographic parity considers sensitive attributes, such as gender, race, or age, and aims to avoid clustering results that disproportionately favor or discriminate against specific groups in the attributes.

The balance fairness notion is an example of a group-level measure, first proposed by [5] and inspired by disparate impact. As is standard with clustering, we are given a set of X points lying in some metric space, and we want to find a partition of X into k different clusters that optimizes some particular objective function.

Regarding the balance notion, we assume that a point $x \in X$ can be split into protected (sensitive) attributes x_p and unprotected (non-sensitive) attributes x_u that are not directly linked to any of the protected attributes. That is, $x = \{x_p, x_u\}$. To illustrate, let the assignment of x_p to a color represent a point's protected label. Then, the concept of disparate impact and fair representation can be understood as the goal of achieving color *balance* within each cluster so that the clusters consist of a similar distribution of colors.

The balance of a cluster $c \in C$ is determined by the distribution of points into sensitive groups $g \in G$ in that cluster, where we must ensure that the number of points from each group in each cluster is proportional to the overall group sizes for the entire dataset. For example, the points could be partitioned into two groups named $orange$ and $blue$, representing the sensitive attributes. Then, according to [5], assuming a uniform distribution of points into the groups, i.e. the groups are of the same size, we can define a balance for a cluster c as:

$$balance(c) = \min\left(\frac{\#orange(c)}{\#blue(c)}, \frac{\#blue(c)}{\#orange(c)}\right) \in [0,1]. \tag{3.1}$$

Furthermore, we can define an evaluation measure for fairness for the full clustering C as:

$$balance(C) = \min_{c \in C} balance(c) \tag{3.2}$$

A cluster c containing an equal number of orange and blue points is considered to have a perfect balance, with a balance value of 1. On the other hand, if a cluster consists entirely of points of a single color (either all orange or all blue), it is considered completely unbalanced, and its balance value is 0. The balance notion presented in [5] is demonstrated in Figure 3.1. The balance measure will be generalized to cases with multiple protected groups in Section 4.1.

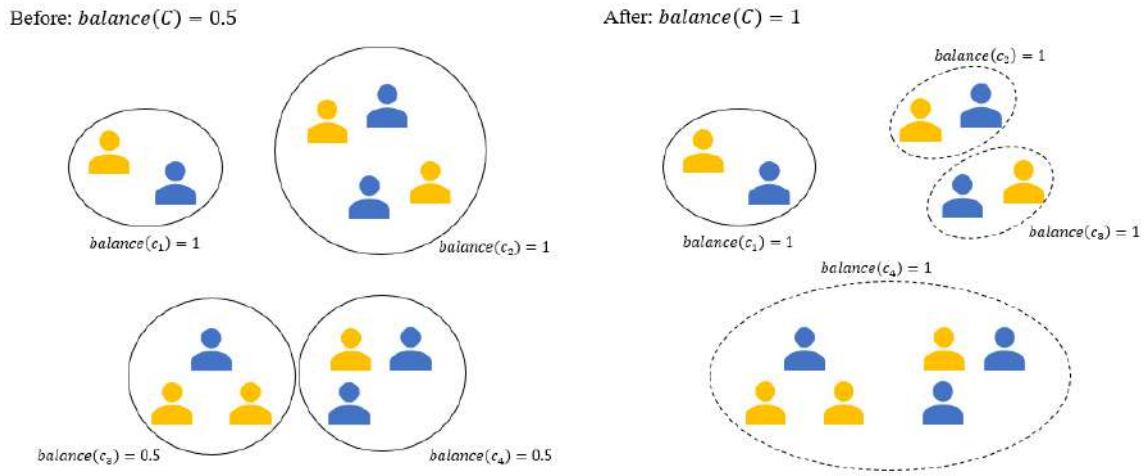

Figure 3.1: Example scenario for understanding balance fairness notion for clustering. To the left, we have a vanilla clustering algorithm. To the right, a balance fairness constraint has been imposed via a fair clustering algorithm.

Figure 3.1 depicts a scenario in which we have a group of two types of experts, orange and blue in, for example, a panel discussion. The goal is to group the experts to organize each panel discussion. Each panel discussion should contain a fair representation of each expert to not produce bias in the audience's opinion regarding a subject. When applying vanilla clustering, we might find the clusters represented by the solid circles on the left hand side of Figure 3.1. Some clusters have a balance of 1, others a balance of 0.5. The overall balance for the clustering is $balance(C) = 0.5$. In certain situations, a fairness balance of 0.5 could be fine; however, in this situation, we want a completely fair representation of both types of experts. Therefore, we find an alternate clustering given a clustering algorithm with fairness constraints. This is represented on the right-hand side of Figure 3.1 by the dashed lines and produces an overall clustering balance of $balance(C) = 1$, with 50-50 expert opinions as an output. However, in applying fairness constraints, we may have lost some amount of clustering quality. The trade-off between fairness and cluster quality is application specific to whether it is valuable to have this loss in quality for fairness.

3.3 Density-Connectivity Distance

The density-connectivity distance (*dc-dist*) [7] is a measure of the distance between two nodes in a dataset. It is computed by finding the minimax distance of a graph, where

only *dense paths* are considered, that is, where all codes are *core nodes*. To construct the graph, we use the *mutual reachability distance*, which combines the Euclidean distances between two points and the value at which both points are considered *core nodes*. DBSCAN [6] connects points that are close to each other based on their mutual reachability distance. The mutual reachability distance represents how easily two points can be reached from each other within a certain neighborhood threshold ϵ. If two points have a mutual reachability distance less than or equal to ϵ, they are considered connected. Since the connections in a cluster are based on the mutual reachability distance, it means that each step in the path connecting the points has a mutual reachability distance less than or equal to ϵ. Therefore, even the longest step in the path of connected points is still less than ϵ. Consequently, the mutual reachability distance between any point in one cluster and any point in another cluster is greater than ϵ, as the clusters have no direct connections between them.

Since DBSCAN clusters require that the intra-cluster dc-dist values are less than ϵ, we can apply a thresholding operation to a matrix M that represents the pairwise distances between points using the dc-dist. In this matrix, a value of 1 indicates that the distance between a pair of points is less than or equal to ϵ, and a value of 0 means the distance is greater than the neighborhood threshold ϵ. This thresholding operation based on ϵ identifies which points are within an ϵ-ball of each other. The construction of ϵ-balls of points forms the core of the DBSCAN clusters. Thus, by adjusting ϵ, we control the size and density of the clusters. So matrix M, derived from dc-dist values, is a reliable tool to determine clusters in DBSCAN. The dc-dist then, through its influence on the mutual reachability distance, gives the smallest ϵ value needed for two points to be core points of the same DBSCAN cluster. Furthermore, [7] shows that DBSCAN clusterings are equivalent under the Euclidean and dc-dist distance metrics.

In [7], it is shown that the ultrametric [26] nature of the dc-dist allows us to find an optimal clustering solution, which corresponds to cuts in a graph. These cuts can be obtained by analyzing the spectrum of the graph using our distance measure. As a result, spectral clustering using the dc-dist produces the exact same clustering results as DBSCAN. In the paper, the minimum cut MinCut was regarded as the partitioning type, where a similarity matrix S based on the distance measure dc-dist (d_{dc}) is defined as $S_{[i,:]} := 1 - \frac{d_{dc_{[i,:]}}}{||d_{dc_{[i,:]}}||}$. The spectral clustering applied to the distance measure dc-dist is referred to as *Ultrametric Spectral Clustering* (USC).

3.4 FairSC - Fair Spectral Clustering Algorithms

In their work, Kleindessner et al. [13] propose a novel approach to spectral clustering that incorporates the balance fairness constraint as defined by [5] and described in Section 3.2. By introducing balance fairness measures and incorporating them as constraints in the clustering algorithm, the paper aims to ensure that the resulting clusters not only capture the inherent structure of the data but also adhere to the fairness criterion. Their approach can be likened to various versions of constrained spectral clustering [20] [27] that aim to include must-link constraints, where specific vertices A and B of a graph should be assigned to the same cluster. Constrained spectral clustering [28] refers to the process of incorporating additional constraints into the algorithm to adhere to some criterion. In [13], the fairness constraints are integrated into the spectral clustering algorithm to ensure balance in the clustering outcome.

The paper introduces a way to mathematically express and enforce this fairness constraint onto a clustering. It states that for each cluster $c \in C$ in the dataset X and a given set of sensitive groups G, the cluster is fair if it contains roughly the same fraction of data points from each sensitive group $g \in G$ as the full dataset. The paper incorporates this notion into the RatioCut objective function as a linear constraint on the matrix H, representing the cluster assignments, which gives rise to Algorithm 1.

Algorithm 1 Unnormalized SC with fairness constraints (FairSC)

Input: weighted adjacency matrix $W \in \mathbb{R}^{n \times n}$; $k \in \mathbb{N}$; group-membership vectors G
Output: a clustering of $[n]$ into k clusters

1: compute Laplacian matrix $L = D - W$
2: let F be a matrix with columns $G - \frac{|g|}{n} \cdot \mathbf{1}_n$
3: compute a matrix Z whose columns form an orthonormal basis of the nullspace of F^T
4: compute the k smallest (respecting multiplicities) eigenvalues of $Z^T L Z$ and the corresponding orthonormal eigenvectors (written as columns of Y)
5: apply k-means clustering to the rows of $H = ZY$

Algorithm 1 takes as an input a weighted adjacency matrix W, the number of clusters k in the spectral clustering algorithm, and the group-membership vectors G that indicate the membership of each data point in the specific group g. Recall that n is the number of data points in dataset X and let h denote the number of groups in G. The algorithm begins by computing the Laplacian matrix $L \in \mathbb{R}^{n \times n}$. Then, it constructs the fairness matrix $F \in \mathbb{R}^{n \times (h-1)}$ using group-membership vectors G as columns, adjust-

ing for group sizes. The proportion of the group in the dataset $\frac{|g|}{n}$ is subtracted from each group-membership vector G, ensuring that the resulting matrix F reflects the deviations from the average representation of each group. Next, it computes a matrix $Z \in \mathbb{R}^{n\times(n-h+1)}$, whose columns form an orthonormal basis of the nullspace of F^T. The nullspace of the matrix consists of all vectors that, when multiplied by the matrix, result in the zero vector [29]. In this case, the null space represents the subspace where the fairness constraints are satisfied, as the vectors within this subspace contribute equally to each group's representation. The orthonormal basis ensures that the basis vectors in the null space are independent and orthogonal to each other. The algorithm then proceeds to perform eigendecomposition for the k smallest eigenvalues and their corresponding eigenvectors of Z^TLZ, which are denoted as columns of $Y \in \mathbb{R}^{(n-h+1)\times k}$. Finally, it applies k-means clustering to the matrix rows $H = ZY$. The output is a clustering of the dataset into k clusters.

3.5 Extension of FairSC - The Insertion of dc-dist

By combining the research of [7] and the fair spectral clustering algorithm FairSC [13], we can attempt to produce a fair clustering method rooted in density-based clustering. The input to the algorithm includes a dataset X, the desired number of clusters k, group-membership vectors G representing the demographic groups, and a parameter $minPts$. The output is a clustering of the data into k clusters.

The algorithm begins by computing the ultrametric dc-distance matrix $D_{dc} \in \mathbb{R}^{n\times n}$ from the dataset X. This is done through the dc-dist formula [7] described in Section 3.3. Inspired by [13], we then construct a fairness matrix $F \in \mathbb{R}^{n\times(h-1)}$, where each column represents a group-membership vector G minus the proportion of the corresponding group proportion $\frac{|g|}{n}$. Next, we calculate the matrix $Z \in \mathbb{R}^{n\times(n-h+1)}$, which contains an orthonormal basis of the null space of F^T. By computing Z^+, the Moore-Penrose pseudoinverse [30] of Z, the algorithm prepares for the subsequent steps.

The ultrametric distance matrix contains information describing the data. By multiplying Z onto both sides of D_{dc}, we receive a $Z^TD_{dc}Z \in \mathbb{R}^{(n-h+1)\times(n-h+1)}$ matrix that is one dimension less than the full dataset, ergo does not fully represent the full dataset with the fairness constraint. One way to solve this is to use the pseudo-inverse to reconstruct the original matrix [31]. By using the pseudoinverse, we can reconstruct the original matrix from its transformed version by multiplying the transformed matrix

by its pseudoinverse. This operation effectively "undoes" the transformation, and the result approximates the original matrix as closely as possible.

The original dimensions of the matrix D_{dc}, now with the balance fairness constraints injected, is computed via $(Z^+)^T(Z^T D_{dc} Z)Z^+ \in \mathbb{R}^{n \times n}$. Then, the algorithm computes the k smallest eigenvalues (taking multiplicities into account) and the corresponding orthonormal eigenvectors of the matrix $(Z^+)^T(Z^T D_{dc} Z)Z^+$. These eigenvalues and eigenvectors are stored in the matrix Y, where each column represents an eigenvector. Finally, the algorithm applies k-means clustering to the rows of Y, assigning each data point to one of the k clusters based on their similarities in the transformed space, as can be seen in Algorithm 2.

Algorithm 2 Unnormalized SC with fairness constraints and dc_dist (FairDC_dist)

Input: dataset $X \in \mathbb{R}^{n \times n}; k \in \mathbb{N}; k \in \mathbb{N}$;group-membership vectors$G$, minPts
Output: a clustering of $[n]$ into k clusters

1: compute D_{dc} from X
2: let F be a matrix with columns $G - \frac{|g|}{n} \cdot \mathbf{1}_n$
3: compute a matrix Z whose columns form an orthonormal basis of the nullspace of F^T
4: compute Z^+
5: compute the k smallest (respecting multiplicities) eigenvalues of $(Z^+)^T(Z^T D_{dc} Z)Z^+$ and the corresponding orthonormal eigenvectors (written as columns of Y)
6: apply k-means clustering to the rows of Y

Chapter 4

Experiments

In this chapter, we cover the experiments carried out on two synthetic datasets and two real-life datasets using the methodology presented in section 3.5. We also describe the evaluation metrics for both the degree of balance within the clusterings and the clustering quality.

4.1 Evalutation Metrics

For this project, we evaluate the clustering via two metrics: the quality of clustering and balance. Balance, in this context, serves as a measure of fairness.

To define the fairness measure, there is a generalized approach of balance [5] defined in [17]. In this paper, two intermediate values, r, associated with a group g in the entire dataset, and $r_g(c)$, associated with a group g in a cluster c, are defined. They are defined as: $r_g := \frac{|g|}{|X|}$ and $r_g(c) = \frac{|g(c)|}{|c|}$, respectively. Thus, the balance for a cluster is defined as

$$balance(c) = \min_{g \in G} balance_g(c), \tag{4.1}$$

where

$$balance_g(c) = \min\left(\frac{r_g}{r_g(c)}, \frac{r_g(c)}{r_g}\right) \tag{4.2}$$

As in [5], the balance for the full clustering is:

$$balance(C) = \min_{c \in C} balance(c) \tag{4.3}$$

Notice how the cluster balance in (4.1) is a minimum of a vector of values $\{balance_g(c)\}_{g \in G}$, which becomes important in the discussion.

For the quality of clustering, considering clustering is an unsupervised technique and we, therefore, do not have a ground truth clustering, we can use the silhouette coefficient s_c. The silhouette coefficient is a common measurement used in cluster analysis

to assess the accuracy and consistency of clustering outcomes [32]. It explains how well each data point fits within its assigned cluster, considering its proximity to other points in its own cluster and its separation from points in other clusters. The silhouette coefficient ranges from -1 to 1, with higher values indicating better clustering. Intuitively, a high silhouette coefficient suggests that the data point is well-clustered, with significant cohesion within its cluster and clear separation from neighboring clusters. A low silhouette coefficient indicates that the data point may be closer to points in other clusters, suggesting potential mislabeling or overlap between clusters.

4.2 Synthetic Data

We conduct a series of experiments to evaluate the performance and fairness of our supposedly fair clustering algorithm in a synthetic clustering scenario. The synthetic nature of the dataset allows us to control the properties and characteristics of the clusters, including the degree of separability and the presence of sensitive attributes.

4.2.1 Clean Synthetic Data

To conduct the initial experiment, we generated a synthetic dataset of 200 random samples drawn from a Gaussian distribution arranged in a square-like formation, as seen in Figure 4.1. The data is uniformly distributed amongst the four clusters. We aim to simulate datasets with multiple meaningful ground-truth clusterings, where only one clustering adheres to fairness criteria. The goal is to assess the algorithm's ability to recover the fair ground-truth clustering. If there is only one meaningful ground-truth clustering present, any standard clustering algorithm (such as USC [7] or DBSCAN [6]) without fairness considerations could successfully recover the ground-truth clustering. This scenario is demonstrated in Figure 4.1.

4.2.2 Non-Convex Synthetic Dataset

In the second synthetic scenario, we utilize the well-known Moons dataset, which features distinct moon-shaped clusters. This dataset presents a challenging scenario for clustering algorithms as the data clusters are not convex. We generate the synthetic dataset using the make_moons function from the scikit-learn library. It consists of 400 samples, evenly distributed among four moon-shaped clusters. These clusters are

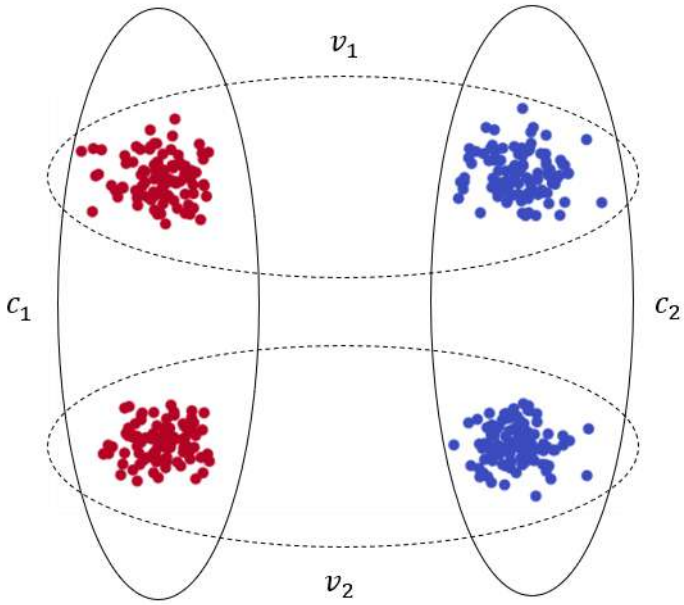

Figure 4.1: Example of synthetic dataset. There are two meaningful ground-truth clusterings into the two clusters: $C = c_1 \cup c_2$ and $C = v_1 \cup v_2$. The latter is considered fair as it contains a balanced representation of each group.

intentionally skewed and aligned with a binary sensitive attribute. The experimental setup mirrors that of the clean synthetic dataset, where our objective is to model datasets with multiple meaningful ground truth clusterings, among which only one adheres to fairness criteria.

4.3 Real Data

4.3.1 Adult Census Dataset

The Adult dataset [33], taken from the USA 1994 Census database, is a popular dataset used for machine learning tasks. It contains demographic information about individuals such as age, education, occupation, marital status, race, sex, native country, and whether they earn more than 50000 per year. The dataset is commonly used to predict whether an individual's income exceeds the 50000 per year threshold based on the other attributes provided. The dataset comprises of 48842 examples, of which there are 15 attributes, 6 numerical, 7 nominal, and 2 binary. It has been acknowledged

that there is bias in the Adult dataset related to the attributes 'sex' against the *female* group [34]. The majority of instances within the Adult dataset are male. The ratio of $male : female$ instances is $32650 : 16192$ ($66.9\% : 33.1\%$).

During data preprocessing, we handle missing values by removing any data points that contain them. We select the same set of numerical attributes as in [5] and [17]. The chosen attributes are age, fnlwgt, education-num, capital-gain, and hours-per-week to represent points in the Euclidean space. The data is standardized, and the relevant sensitive attributes (sex, race) are one-hot encoded. Consequently, we obtain a dataset consisting of 45222 instances. When sex is our sensitive attribute, the dataset consists of 7 attributes, including the sensitive attribute itself. Due to one-hot encoding of the sensitive attribute race, there are 9 attributes, including the sensitive attribute itself. The data is subsampled to 905 data points, with care taken to ensure that the ratio between the groups in the sensitive attributes is upheld to realistically represent the dataset.

4.3.2 Bank Marketing Dataset

The bank marketing dataset [35] contains information related to a direct phone call marketing campaign conducted by a Portuguese banking institution from 2008 to 2013, aiming to promote term deposits among existing customers. The dataset consists of a total of 45211 instances, making it a relatively large and comprehensive dataset for analysis. It encompasses various attributes such as client demographics, economic indicators, and previous marketing campaign outcomes. There are 6 nominal, 4 binary, and 7 numerical attributes. The categorical attribute { *marital status*} is considered sensitive [5] [17]. The sensitive attribute has 3 values: { *married, single, divorced* }. The married group has the majority with a ratio of $married : single : divorced$ at $27214 : 12790 : 5207$ ($60.2\% : 28.3\% : 11.5\%$).

During data preprocessing, we handle missing values by removing any data points that contain them. We select the set of numerical attributes as in [5] [17]. The attributes chosen are age, balance, and duration. The data is standardized, and the relevant sensitive attribute {marital-status} is one-hot encoded. Consequently, we obtain a dataset consisting of 45211 instances described by 6 attributes, including the one-hot encoded sensitive attribute {marital-status}. The data is subsampled to 452 data points, with care taken to ensure that the ratio between the sensitive groups is upheld to realistically represent the dataset.

Chapter 5

Results and Discussion

In this section, we employ our algorithm by performing experiments on both the synthetic and the real data. The goal of our experiments is to first show that the new algorithm that combines the dc-dist and FairSC can uncover meaningful fair clusters and, second, to compare the fairness of the fair algorithms with vanilla clustering algorithms via the balance evaluation metric.

We compare the various clustering algorithm results: (1) the synthetic clustering data having generated cluster labels, (2) our own fairness algorithm with DC_dist, (3) vanilla spectral clustering [24], (4) the FairSC algorithm [13], and (5) the vanilla DBSCAN [6] algorithm without the fairness notion.

5.1 Synthetic Data

5.1.1 Clean Synthetic Data

In Figure 5.1 we see the results of the various algorithms on the clean synthetic dataset with a binary sensitive attribute. The algorithmic hyper-parameters have been set to $k = 2$ for FairSC, vanilla SC, and FairDC_dist. For the vanilla DBSCAN algorithm, the parameters are set to $minPts = 5$ and $\epsilon = 1.8$.

Both the FairDC_dist algorithm and the FairSC algorithm produce similar clustering results. They both produce a perfect balance of 1, meaning that the data points are clustered proportionally to their representation in the dataset. This means that there is a 50-50 distribution for the protected attribute in each cluster. This aligns with the situation for a fair clustering, as described in Section 4.2.1. The silhouette coefficient s_c [32] for each clustering algorithm at $k = 2$ is $s_c = 0.54$. This indicates a somewhat meaningful clustering quality has been maintained while ensuring balance in the sensitive attributes for the FairDC_dist algorithm and the FairSC algorithm. The balance of the VanillaSC algorithm and the DBSCAN algorithm is 0, so their output is an unbalanced clustering.

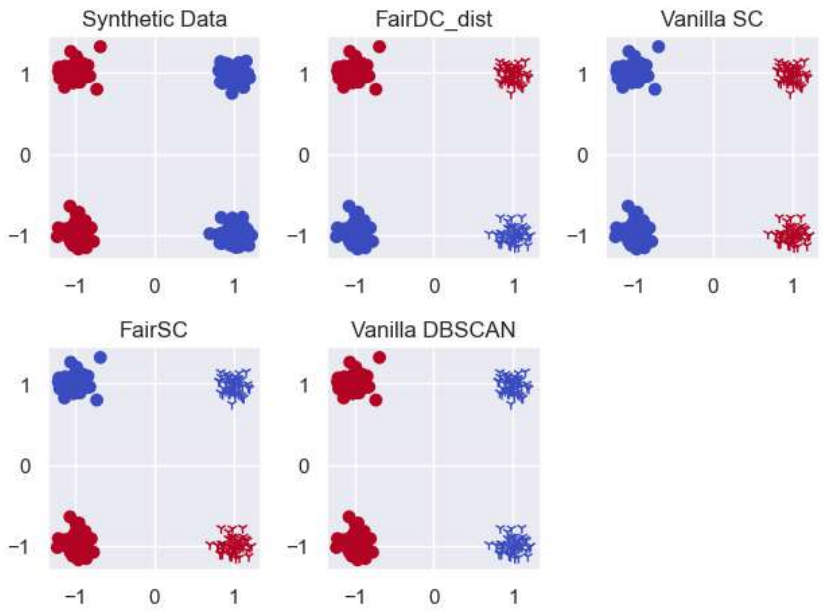

Figure 5.1: The results of the algorithms along with their corresponding silhouette coefficient. The shapes show the original synthetic labels and the colors show the algorithm's clustering outcomes. Balance results: Balance(FairDC_dist) = 1, Balance(SC) = 0, Balance(FairSC) = 1, Balance(DBSCAN) = 0. Silhouette coefficient results: s_c(FairDC_dist) = 0.54, s_c(SC) = 0.54 , s_c(FairSC) = 0.54, s_c(DBSCAN) = 0.54.

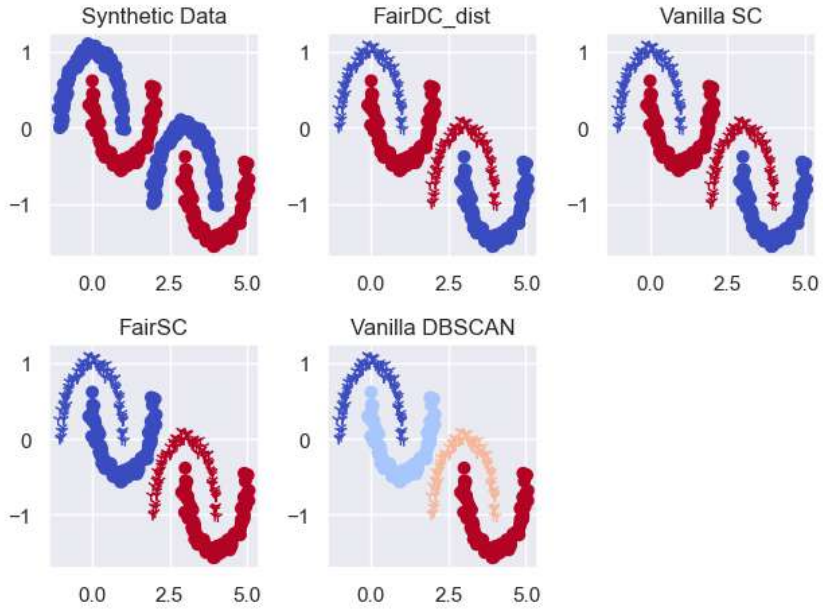

Figure 5.2: The results of the algorithms along with their corresponding silhouette coefficient. The shapes show the original simulated labels, and the colors show the algorithm's clustering outcomes. Balance results: Balance(FairDC_dist) = 1, Balance(SC) = 0, Balance(FairSC) = 1, Balance(DBSCAN) = 0. Silhouette coefficient results: s_c(FairDC_dist) = 0.12, s_c(SC) = 0.12 , s_c(FairSC) = 0.58, s_c(DBSCAN) = 0.28.

5.1.2 Non-Convex Synthetic Data

In Figure 5.2 we see the results of the various algorithms on the Moons synthetic dataset with a binary sensitive attribute. The hyper-parameters have been chosen to be $k = 2$ for FairSC, vanilla SC, and FairDC_dist. For the vanilla DBSCAN algorithm, the parameters are set to $minPts = 5$ and $\epsilon = 1$.

The FairDC_dist algorithm, FairSC algorithm and vanilla SC algorithm produce a perfect balance of 1. However, the silhouette coefficient for FairDC_dist algorithm indicates poor clustering. From the plots in Figure 5.2, we can see that the fair outcome of the FairDC_dist algorithm is meaningless. This brings us to an important observation: while FairDC_dist works towards a fair clustering according to balance, it does not prioritize a good clustering quality while upholding the balance constraint like FairSC. On the other hand, the FairSC algorithm's outcome is a somewhat meaningful balanced cluster with a medium silhouette coefficient for the clustering. The results with four clusters for the DBSCAN algorithm indicate that the parameter for ϵ needs to be tuned.

5.2 Real Data

5.2.1 Adult Dataset (Sex)

In Figure 5.3, we see the balance score and the silhouette coefficient of the various algorithms on the Adult dataset with sex as the sensitive binary attribute. With the exception of DBSCAN, we iterate over the number of clusters $k \in [2,6]$. For the vanilla DBSCAN algorithm, the parameters are set to $minPts = 5$, and we tune ϵ to fit the range of k.

Observe that the FairDC_dist and FairSC balance the sensitive groups well across 2 clusters; however, the corresponding silhouette coefficient once again shows that FairDC_dist clusters well for balance but not so well for quality. The vanilla algorithms fare poorly in both cases, showcasing a situation in which there needs to be some form of fairness constraint enforced to not output biased clusterings.

The DBSCAN algorithm has a constant balance of 0. It seems as if the algorithm is separating outliers into clusters as well, which results in a poor balance and a poor silhouette coefficient. DBSCAN results in an outlier group that, although it is not a cluster, can also be checked for the ratio of the sensitive attribute. It is difficult to obtain a meaningful clustering (with more than one cluster) from DBSCAN without creating a large number of outliers. Therefore, the DBSCAN algorithm performs poorly in balancing the clusters as it tends to group outliers as smaller, meaningless clusters. Thereby, the algorithm generates additional outlier groups that do not represent a valid cluster but can still be evaluated for the ratio of the sensitive attribute.

5.2.2 Bank Marketing Dataset (Marital)

In the Adult dataset, the sensitive attribute is binary. We wish to test whether the algorithm is also effective for sensitive attributes with $|G| > 2$. We see the results in Figure 5.4. As before, with the exception of DBSCAN, we iterate over the number of clusters k. For the vanilla DBSCAN algorithm, the parameters are set to $minPts = 5$ and ϵ is tuned to fit the range of $k \in [2,6]$.

The FairDC_dist seems to produce balanced clusters, however, upon closer inspection of the silhouette coefficient, we observe the same trends as with the Adult dataset as within the Bank Marketing dataset, namely that the FairDC_dist algorithm clusters well for balance but poorly for clustering quality. Due to outliers separated into

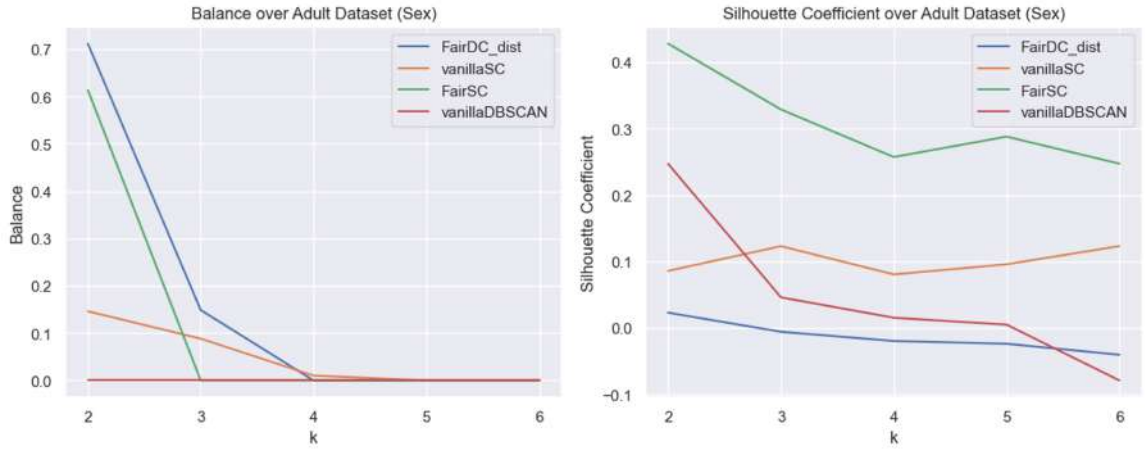

Figure 5.3: Balance scores results of the algorithms along with their corresponding silhouette coefficient for each k. Note that the number of clusters k in DBSCAN is determined by tuning the ϵ to fit in the range of clusters as defined in the figure.

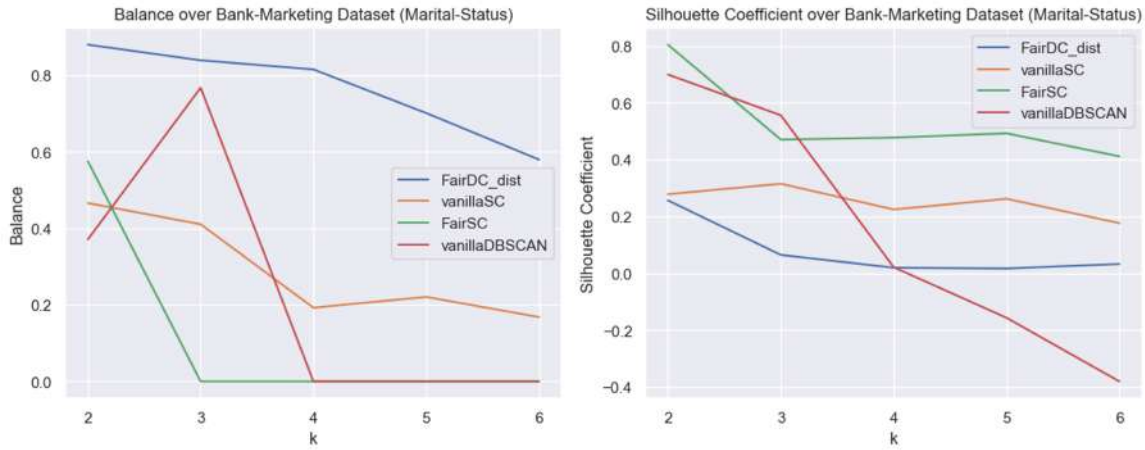

Figure 5.4: Balance scores results of the algorithms along with their corresponding silhouette coefficient for each k. Note that the number of clusters k in DBSCAN is determined by tuning the ϵ to fit in the range of clusters as defined in the figure.

clusters, the FairSC algorithm clusters poorly with regard to balance when $k \geq 3$. Surprisingly, the vanilla DBSCAN algorithm outputs clusters that are relatively balanced and have a higher silhouette coefficient when the epsilon parameter is tuned to produce 3 clusters. This can most likely be attributed to the uncleanliness of the data and the way DBSCAN handles outliers.

5.2.3 Discussion

Upon observation, taking into account the issue of outliers, we can conclude that the FairDC_dist algorithm, despite its emphasis on fairness, sacrifices clustering quality. This trade-off between fairness and clustering quality is important to consider, as prioritizing fairness might overlook natural grouping patterns in the data, leading to sub-optimal clustering results, e.g., Figure 5.3 where the FairDC_dist produces a well-balanced but poorly clustered clustering outcome. It is difficult to pinpoint exactly which aspect of the fairDC_dist algorithm affects the algorithm's attempt to cluster for quality. One potential candidate could be the absence of the Laplacian, as spectral clustering utilizes the smallest eigenpairs of the Laplacian matrix to embed the data points into a lower-dimensional space. By removing the Laplacian from the equation, the spectral embedding step would be missing, which could lead to a loss of information necessary for clustering [24].

Regarding balance, there are multiple ways in which that can be defined. In fact, the balance measure used in [17] is not a strictly proper generalization of the measure used in [5], in contrast to the general acceptance as the preferred generalization in the literature. This is because [17] utilizes a one-vs-all ratio, whereas [5] uses a one-vs-all-remaining ratio. To illustrate the difference, using the one-vs-all ratio for the example in Figure 3.1 would result in the cluster balances $balance(c_3) = \frac{2}{3}$ and $balance(c_4) = \frac{2}{3}$ as opposed to the balances of 0.5. One could also consider an all-pairs comparison as a strictly proper generalization of the balance measure in [5].

Measuring the balance accurately when we have a higher cardinality of sensitive groups, such as in Section 4.3.2, becomes more difficult. When one of the sensitive groups is not proportional to the remaining sensitive groups, we receive a very low balance using (4.3), even if the remaining groups are balanced well. Symmetric KL-divergence [36] between normalized versions of $\{balance_g(c)\}_{g \in G}$ and $\{balance_g\}_{g \in G}$ may correct for this issue, but it has to be explored. It is worth noting that the algorithms used in this report employed default parameters, which may not be optimized

for every dataset. Tuning the parameters according to the dataset characteristics can significantly improve the performance and accuracy of the algorithms.

Clean data is crucial for reliable clustering results, especially when dealing with outliers. Outliers can significantly impact the clustering process, making it important to identify and handle them appropriately. For instance, in the Adult dataset, it was discovered that there were outliers present within the data, skewing the clusterings and, in particular, the fairness results, as an outlying 'cluster' with only one data point per definition results in a fairness measure of 0. For example, in the Adult dataset, we discovered points with a 'capital gain' value equal to 99999. Upon further investigation, it was discovered that this was encoded as a warning message but not removed from the dataset. To address this issue and improve the fair clustering performance, it would be advisable to thoroughly clean the data by identifying and appropriately treating outliers.

Chapter 6

Conclusion

In this project, we focused on exploring fairness in density-based clustering and developing a fair-clustering algorithm inspired by previous work in the area. Fairness considerations in density-based clustering have been relatively understudied compared to other clustering methods. To bridge this gap, we adopted the dc-dist metric [7], a novel density-connectivity metric, which establishes a connection between density-based clustering and spectral clustering. We then adapted a fair spectral clustering algorithm [13] and integrated the dc-dist to develop a fair clustering algorithm rooted in density-based clustering. We sought to assess the fairness of the resulting algorithm by employing established fairness evaluation metrics and comparing the fairness outcomes of the original clustering algorithms with the adapted fair clustering algorithms.

The findings of this research highlighted differences between the investigated algorithms. The fair clustering algorithm, FairDC_dist, prioritized fairness but exhibited lower clustering quality than the original algorithms and the fair spectral clustering algorithm [13]. It brought up the important notion of a trade-off between fairness and clustering quality. The trade-off is important to consider, as overly prioritizing fairness may overlook natural patterns in the data, resulting in poor clustering outcomes. The absence of the Laplacian in FairDC_dist, which is crucial for spectral clustering, could potentially contribute to the loss of clustering information and impact the overall clustering quality.

To assess the fairness of the resulting algorithm, established fairness evaluation metrics were used to compare the fairness outcomes of the original clustering algorithms with the fair clustering algorithms. Evaluating balance in sensitive groups with a higher cardinality presented challenges. Further exploration is needed to establish more accurate and comprehensive approaches for measuring balance in such scenarios, i.e., an exploration into the symmetric KL-divergence of the sensitive groups.

Overall, this research contributes to the understanding of fairness in density-based clustering and highlights the need for further investigation and development of fair clustering methods in density-based clustering. Future work should focus on exploring different optimization techniques that take into consideration clustering quality to fur-

ther enhance our understanding of how fairness can be effectively incorporated in this context. Furthermore, it would be prudent to examine alternative balance measures, especially in the cases of a higher cardinality within the sensitive groups.

Bibliography

[1] A. Chouldechova and A. Roth, "The frontiers of fairness in machine learning," 2018.

[2] F. Martínez-Plumed, C. Ferri, D. Nieves, and J. Hernández-Orallo, "Fairness and missing values," 2019.

[3] A. W. Flores, K. Bechtel, and C. T. Lowenkamp, "False positives, false negatives, and false analyses: A rejoinder to machine bias: There's software used across the country to predict future criminals. and it's biased against blacks," *Fed. Probation*, vol. 80, p. 38, Sep. 2016.

[4] J. Dastin, "Amazon scraps secret ai recruiting tool that showed bias against women," *Ethics of Data and Analytics: Concepts and Cases*, p. 296, 2022.

[5] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii, "Fair clustering through fairlets," 2018.

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," KDD'96, p. 226–231, AAAI Press, 1996.

[7] A. Beer, A. Draganov, E. Hohma, P. Jahn, C. M. Frey, and I. Assent, "Connecting the dots — density-connectivity distance unifies DBSCAN, k-center and spectral clustering," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23)*, (New York, NY, USA), pp. 6–10, ACM, August 2023.

[8] P. Bhattacharjee and P. Mitra, "A survey of density based clustering algorithms," *Frontiers of Computer Science*, vol. 15, pp. 1–27, 2021.

[9] A. Chhabra, V. Vashishth, and P. Mohapatra, "Fair algorithms for hierarchical agglomerative clustering," 2021.

[10] G. Rutherglen, "Disparate impact under title vii: An objective theory of discrimination," *Virginia Law Review*, vol. 73, no. 7, pp. 1297–1345, 1987.

[11] A. Backurs, P. Indyk, K. Onak, B. Schieber, A. Vakilian, and T. Wagner, "Scalable fair clustering," 2019.

[12] C. Rösner and M. Schmidt, "Privacy preserving clustering with constraints," *arXiv preprint arXiv:1802.02497*, 2018.

[13] M. Kleindessner, S. Samadi, P. Awasthi, and J. Morgenstern, "Guarantees for spectral clustering with fairness constraints," 2019.

[14] X. Chen, B. Fain, L. Lyu, and K. Munagala, "Proportionally fair clustering," in *International Conference on Machine Learning*, pp. 1032–1041, PMLR, 2019.

[15] I. M. Ziko, E. Granger, J. Yuan, and I. B. Ayed, "Variational fair clustering," 2020.

[16] M. Kleindessner, P. Awasthi, and J. Morgenstern, "Fair k-center clustering for data summarization," 2019.

[17] S. K. Bera, D. Chakrabarty, N. J. Flores, and M. Negahbani, "Fair algorithms for clustering," 2019.

[18] C. Rösner and M. Schmidt, "Privacy preserving clustering with constraints," 2018.

[19] S. X. Yu and J. Shi, "Grouping with bias," in *Advances in Neural Information Processing Systems* (T. Dietterich, S. Becker, and Z. Ghahramani, eds.), vol. 14, MIT Press, 2001.

[20] S. Yu and J. Shi, "Segmentation given partial grouping constraints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 173–183, 2004.

[21] Z. Lu and M. A. Carreira-Perpinan, "Constrained spectral clustering through affinity propagation," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[22] L. Xu, W. Li, and D. Schuurmans, "Fast normalized cut with linear constraints," pp. 2866 – 2873, 07 2009.

[23] J. Wang, D. Lu, I. Davidson, and Z. Bai, "Scalable spectral clustering with group fairness constraints," 2023.

[24] U. von Luxburg, "A tutorial on spectral clustering," 2007.

[25] M. Feldman, S. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," 2015.

[26] S. Semmes, "An introduction to the geometry of ultrametric spaces," 2007.

[27] J. Kawale and D. Boley, "Constrained spectral clustering using l1 regularization," in *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 103–111, SIAM, 2013.

[28] X. Wang, B. Qian, and I. Davidson, "On constrained spectral clustering and its applications," *Data Mining and Knowledge Discovery*, vol. 28, pp. 1–30, 2014.

[29] S. Boyd, "Lecture notes for ee263." Stanford University, 2008.

[30] R. Penrose, "A generalized inverse for matrices," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 3, p. 406–413, 1955.

[31] C. Zhang, S. Guo, J. Cao, J. Guan, and F. Gao, "Object reconstitution using pseudo-inverse for ghost imaging," *Opt. Express*, vol. 22, pp. 30063–30073, Dec 2014.

[32] M. J. Zaki and W. Meira, *Data mining and analysis: fundamental concepts and algorithms*. Cambridge University Press, 2014.

[33] B. Becker and R. Kohavi, "Adult." UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

[34] T. L. Quy, A. Roy, V. Iosifidis, W. Zhang, and E. Ntoutsi, "A survey on datasets for fairness-aware machine learning," *WIREs Data Mining and Knowledge Discovery*, vol. 12, mar 2022.

[35] P. R. S. Moro and P. Cortez, "Bank Marketing." UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5K306.

[36] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

Appendix A

Density-Connectivity Distance

Algorithm 3 Density-Connectivity Distance

Input: Dataset $X = \{x_0, \cdots, x_n\}, x_i \in \mathbb{R}^d$
Output: Distance matrix **D**

D ← zeros((**n**, **n**));
MST ← MstClas();
for do $x \in X$
    MakeTree(MST, $\{x\}$);
end for
for do $edge(x_i) \in PairwiseDistances(X)\ increasing$
    if then $Tree(x_i) \neq Tree(x_j)$
        Union(MST, Tree(x_i), Tree(x_j))
        for do $u \in Leaves(Tree(x_i))$
            for do $v \in Leaves(Tree(x_j))$
                D$[u, v] \leftarrow distance(x_i, x_j)$;
            end for
        end for
    end if
end for
return D