When Correlation Clustering Meets Fairness Constraints

Francesco Gullo[1], Lucio La Cava[2],
Domenico Mandaglio[2], and Andrea Tagarelli[2]

[1] UniCredit, Rome, Italy
gullof@acm.org
[2] DIMES Dept., University of Calabria, Rende (CS), Italy
{lucio.lacava, d.mandaglio, tagarelli}@dimes.unical.it

Abstract. The study of fairness-related aspects in data analysis is an active field of research, which can be leveraged to understand and control specific types of bias in decision-making systems. A major problem in this context is fair clustering, i.e., grouping data objects that are similar according to a common feature space, while avoiding biasing the clusters against or towards particular types of classes or sensitive features. In this work, we focus on a correlation-clustering method we recently introduced, and experimentally assess its performance in a fairness-aware context. We compare it to state-of-the-art fair-clustering approaches, both in terms of classic clustering quality measures and fairness-related aspects. Experimental evidence on public real datasets has shown that our method yields solutions of higher quality than the competing methods according to classic clustering-validation criteria, without neglecting fairness aspects.

1 Introduction

We live in an era where machine learning is increasingly pervasive in our society. Every day we interact with machine learning systems, even without knowing it, and these acquire more and more decision-making power in our lives. For instance, such systems support, or even replace, decision makers in financial [22], medical [21], or legal [17] domains. Given their delicate role, machine learning systems should guarantee correct functioning and not discriminate those who entrust their decisions. In this context, however, a critical aspect emerges: the data used by such systems are often (intrinsically) biased, resulting from incorrect data collection processes. Thus, it is desirable to avoid machine learning algorithms being affected by, or even amplifying, this bias. For instance, in [16], this refers to removing *disparate impact*, according to which no group of individuals should (even indirectly) be discriminated by a decision-making system.

In this respect, and by focusing on an unsupervised machine learning setting, in this work we tackle the problem of *fair clustering*. This corresponds to clustering a set of data objects such that: (i) analogously to the classic clustering scenario, similar objects are assigned to the same cluster, whereas dissimilar objects are assigned to different clusters, and (*ii*) the clusters are not dominated by a specific type of sensitive data class (e.g., people having the same sex).

Our key assumption is that the above problem can be addressed under a *correlation clustering* framework [7]. Correlation clustering is a well-established tool for partitioning the set of vertices of an input graph into clusters, so as to maximize the similarity of the vertices within the same cluster and minimize the similarity of the vertices in different clusters, according to pairwise vertex weights expressing positive and negative types of co-association. Specifically, following our recent work in correlation clustering [20], here we provide insights into its application to the problem of fair clustering, and we compare it to some state-of-the-art approaches in such a context. Furthermore, albeit we do not aim to provide a comprehensive experimental survey on fair clustering, a by-product of our work is that, to the best of our knowledge, it represents a valuable and unprecedented experimental comparison between approaches of fair clustering.

Our contributions in this work are as follows:

(i) We provide a comparison between state-of-the-art methods in the context of fair clustering, belonging to different approaches;
(ii) We show how, by optimizing aspects of fairness, some methods affect their ability to produce clusters that are qualitatively good according to classic clustering-validation criteria;
(iii) We shed light on the capabilities of our recently proposed algorithm [20] to adapt to a fair clustering scenario. We show that it is able to produce better solutions than the competing methods from a clustering perspective, while still accounting for fairness-related aspects.

The remainder of the paper is organized as follows. Section 2 provides related work on fair clustering. Section 3 describes how the fair clustering problem can be solved through a correlation clustering framework. Section 4 presents our approach to fair correlation clustering. Section 5 and Section 6 present experimental methodology, while Section 7 discusses our main experimental findings. Section 8 concludes the paper, also providing pointers for future work.

2 Related Work

Although of relatively recent definition, the problem of fairness in clustering has received considerable attention in the literature [13]. With their seminal work, Chierichetti *et al.* [14] were among the first to formalize the notions around fair clustering and the related problem, following the *disparate-impact doctrine* [16]. Their main contribution is a general pre-processing step, i.e., *fairlets decomposition*, to enable traditional algorithms (e.g., k-center and k-median) meeting fairness principles. Following that forerunner work, fairness has become pervasive in the clustering landscape [8, 9, 23], leading to a fairness-aware declination of numerous traditional clustering formulations, such as k-center [18], k-means [1, 24], k-median [6], spectral clustering [19], and hierarchical clustering [2].

The phenomenon of fairness in clustering has also been extended to alternative approaches, such as correlation clustering. In this regard, Ahmadian *et al.* [3] is the first work to leverage the correlation clustering model for the fair clustering task. More specifically, it takes a complete and undirected graph as input, where vertices are assigned a (single) label representing a given protected class attribute (e.g., sex or ethnicity), and the goal is to provide a fair representation of each considered label in the resulting clusters. Recently, Mandaglio *et al.* [20] proposed to model the fair clustering problem of a relational dataset as a correlation clustering instance. Given a set of objects, defined over a set of features, Mandaglio *et al.* build an associated correlation clustering instance by considering the similarity between the tuples. Although Ahmadian *et al.*'s and Mandaglio *et al.*'s approaches aim to cluster different types of data (graphs and tuples, respectively), both approaches reduce the original problem to a correlation clustering instance. However, Mandaglio *et al.*'s formulation is more general than Ahmadian *et al.*'s one, since the former deals with an arbitrary number of labels (or sensitive attributes), while the latter is limited to a single-label setting.

3 Fairness Constraints in Correlation Clustering

3.1 Background on Correlation Clustering

The correlation clustering problem, originally introduced by Bansal *et al.* [7], consists of clustering the set of vertices of a graph whose edges are assigned two nonnegative weights, named positive-type and negative-type weights, respectively. Such weights express the advantage of putting any two connected vertices into the same cluster (positive-type weight) or into separate clusters (negative-type weight). The objective is to partition the vertices so as to either minimize the sum of the negative-type weights between vertices within the same cluster plus the sum of the positive-type weights between vertices in separate clusters (Min-CC), or maximize the sum of the positive-type weights between vertices within the same cluster plus the sum of the negative-type weights between vertices in separate clusters (Max-CC). Both the formulations are **NP**-hard [7, 25] and they are equivalent in terms of optimality. However, the available algorithms for Max-CC [10, 26] are inefficient and poorly usable in practice since they are not able to output more than a fixed number of clusters (i.e., six). Conversely, Min-CC admits approximation algorithms [4, 11] that do not suffer from the limitations of the maximization counterpart. For these reasons, in this work we focus on the minimization formulation of correlation clustering:

*Problem 1 (*Min-CC *[5]).* Given an undirected graph $G = (V, E)$, with vertex set V and edge set $E \subseteq V \times V$, and weights $w^+_{uv}, w^-_{uv} \in \mathbf{R}^+_0$ for all edges $(u, v) \in E$, find a clustering $\mathcal{C} : V \longrightarrow \mathbf{N}^+$ that minimizes:

$$\sum_{(u,v)\in E,\, \mathcal{C}(u)=\mathcal{C}(v)} w^-_{uv} \quad + \sum_{(u,v)\in E,\, \mathcal{C}(u)\neq\mathcal{C}(v)} w^+_{uv}. \tag{1}$$

Min-CC is **APX**-hard [11], but admits approximation algorithms [5, 7, 11, 12, 27] with guarantees depending on the type of input graph. On general graphs and weights, the best known approximation factor is $\mathcal{O}(\log |V|)$ [11, 15], provided by a linear programming approach. Conversely, constant-factor approximation algorithms are possible if the graph is complete and edge weights satisfy the *probability constraint*, i.e., $w^+_{uv} + w^-_{uv} = 1$ for all $u, v \in V$. Among these, the one which provides the best trade-off between efficiency and theoretical guarantees is the Pivot algorithm [5], which simply picks a random vertex u, builds a cluster as composed of u and all the vertices v such that an edge with $w^+_{uv} > w^-_{uv}$ exists, and removes that cluster from the graph. The process is repeated until the graph has become empty. This algorithm has $\mathcal{O}(|E|)$ time complexity and it achieves a factor-5 expected guarantee for Min-CC under the *probability constraint* or if a *global weight bound* holds on the overall edge weights [20].

Next we discuss how a clustering problem with fairness constraint can be profitably solved through a Min-CC approach.

3.2 Problem Statement

Let $\mathcal{X} = \{X_1, \cdots, X_n\}$ be a set $\mathcal{A}$ of n objects defined over a set of attributes. The latter is assumed to be divided into two sets, $\mathcal{A}^F$ and $\mathcal{A}^{\neg F}$. The $\mathcal{A}^F$ set contains *fairness-aware*, or *sensitive*, attributes such as those identifying sex, race, religion, relationship status in a citizen database and any other attribute over which fairness is to be ensured. $\mathcal{A}^{\neg F}$ denotes the attributes that are relevant to the task of interest, and thus can be regarded as *non-sensitive*. In both cases, we assume that part of the attributes might be numerical, and the others as categorical (binary or multi-value). We use subscripts N and C to distinguish the two types, therefore $\mathcal{A}^F = \mathcal{A}^F_N \cup \mathcal{A}^F_C$ and $\mathcal{A}^{\neg F} = \mathcal{A}^{\neg F}_N \cup \mathcal{A}^{\neg F}_C$.

We consider a clustering task whose goal is to partition the input objects with a twofold objective: (i) minimize the inter-cluster similarity according to the non-sensitive attributes $\mathcal{A}^{\neg F}$; (ii) minimize the intra-cluster similarity according to the sensitive attributes $\mathcal{A}^F$. The former objective corresponds to the typical clustering objective, since dissimilar objects should belong to different clusters. Pursuing the second objective, instead, would help distribute objects that are similar in terms of sensitive attributes across different clusters, thus fostering the formation of clusters that are equally represented in terms of the sensitive attributes. This is beneficial to ensure that the distribution of groups defined on sensitive attributes within each cluster approximates the distribution across the dataset. Formally, the problem we tackle in this work is:

*Problem 2 (*Fair-CC*).* Given a set of objects $\mathcal{X}$, two subsets of attributes $\mathcal{A}^F$ and $\mathcal{A}^{\neg F}$, and an object similarity function $sim_S(\cdot)$ defined over the subspace S of the attribute set, find a clustering $\mathcal{C}^*$ to minimize:

$$\sum_{u,v \in \mathcal{X},\, \mathcal{C}(u)=\mathcal{C}(v)} sim_{\mathcal{A}^F}(u,v) \quad + \sum_{u,v \in \mathcal{X},\, \mathcal{C}(u)\neq\mathcal{C}(v)} sim_{\mathcal{A}^{\neg F}}(u,v) \tag{2}$$

The objective in Eq. (2) corresponds to solving a complete MIN-CC instance where the set of vertices corresponds to the objects in $\mathcal{X}$ and, for each pair of vertices u and v, the positive-type (resp. negative-type) correlation-clustering weight corresponds to the similarity score between the two vertices according to the non-sensitive (resp. sensitive) attributes.

We remark that the FAIR-CC problem, as stated above, is introduced here for the first time, while in our previous study in [20] we tackled a different problem: given a set of objects defined over sensitive and not-sensitive attributes, find two attribute subsets that lead to pairwise similarity scores satisfying a certain global condition on the correlation-clustering edge weights. The focus in [20] was to show that the global condition can guide the selection of subsets of features that lead to edge weights expressing the best trade-off between an accurate representation of objects' vectors (i.e., discarding not too many features), and the way how the weights facilitate the downstream correlation-clustering algorithm performing well, i.e., by making it achieve approximation guarantees [20]. Instead, in this work, the set of attributes, over which the similarity scores are computed, are given as input in the FAIR-CC problem, and hence they are not needed to be discovered. This is also a more realistic scenario for fair clustering, where the set of sensitive attributes is provided by the specific application scenario.

4 Algorithm

The FAIR-CC problem requires a function to measure the similarity between two objects with respect to a set of attributes. Following [20], we quantify the degree of similarity between two objects u and v, according to the set of sensitive and non-sensitive attributes, by means of the following $sim_{\mathcal{A}^{\neg F}}(u,v)$ and $sim_{\mathcal{A}^{F}}(u,v)$ measures, respectively:

$$sim_{\mathcal{A}^{\neg F}}(u,v) := \psi^{+}\Big(\alpha_N^{\neg F} \cdot sim_{\mathcal{A}_N^{\neg F}}(u,v) + (1-\alpha_N^{\neg F}) \cdot sim_{\mathcal{A}_C^{\neg F}}(u,v)\Big), \quad (3)$$

$$sim_{\mathcal{A}^{F}}(u,v) := \psi^{-}\Big(\alpha_N^{F} \cdot sim_{\mathcal{A}_N^{F}}(u,v) + (1-\alpha_N^{F}) \cdot sim_{\mathcal{A}_C^{F}}(u,v)\Big), \quad (4)$$

where $\alpha_N^F = |\mathcal{A}_N^F|/(|\mathcal{A}_N^F| + |\mathcal{A}_C^F|)$ and $\alpha_N^{\neg F} = |\mathcal{A}_N^{\neg F}|/(|\mathcal{A}_N^{\neg F}| + |\mathcal{A}_C^{\neg F}|)$ are coefficients to weight similarities proportionally to the number of involved attributes, and $\psi^{+} = exp(|\mathcal{A}^F|/(|\mathcal{A}^F|+|\mathcal{A}^{\neg F}|)-1)$ and $\psi^{-} = exp(|\mathcal{A}^{\neg F}|/(|\mathcal{A}^F|+|\mathcal{A}^{\neg F}|)-1)$ are smoothing factors to penalize correlation-clustering weights that are computed on a small number of attributes. The latter is reasonable as, in a fair clustering task, we usually have fewer sensitive attributes, and it should be avoided that negative-like weights can dominate the positive-like ones. The exponential function enables a mild smoothing, which is desirable.

As FAIR-CC is an instance of MIN-CC, it can be solved by MIN-CC algorithms. Specifically, although it was originally devised for a slightly different problem (as previously explained in Section 3), here we borrow the algorithm

Algorithm 1 CCBounds [20]

Input: Set of objects $\mathcal{X}$, sensitive attributes $\mathcal{A}^F$, non-sensitive attributes $\mathcal{A}^{\neg F}$, Min-CC algorithm A

Output: Clustering $\mathcal{C}$ of $\mathcal{X}$

1: compute $sim_{\mathcal{A}^{\neg F}}(u,v), sim_{\mathcal{A}^F}(u,v)$, $\forall u,v \in \mathcal{X}$, as in Eqs. (3)–(4)
2: build the instance $I = \langle G = (\mathcal{X}, \mathcal{X} \times \mathcal{X}), \{sim_{\mathcal{A}^{\neg F}}(u,v), sim_{\mathcal{A}^F}(u,v)\}_{u,v \in \mathcal{X}\times\mathcal{X}} \rangle$
3: $\mathcal{C} \leftarrow$ run A on I

proposed in [20] and adapt it to solve the Fair-CC problem. This algorithm, dubbed CCBounds [3] and presented in Algorithm 1, consists of building a Min-CC instance with vertices as the input data objects and edge weights as the similarity scores, and then running a Min-CC algorithm A on such a Min-CC instance.

Theoretical remarks. Let $T_{\mathsf{A}}(\mathcal{X})$ be the running time of the algorithm A on the set of data objects $\mathcal{X}$. CCBounds runs in $\mathcal{O}(|\mathcal{X}|^2|\mathcal{A}|+T_{\mathsf{A}}(\mathcal{X}))$ time complexity since it needs to compute a similarity score, over $\mathcal{A}$ attributes, for each pair of objects in $\mathcal{X}$, and then solve the resulting Min-CC instance through algorithm A. Also, the space complexity of CCBounds is $\mathcal{O}(|\mathcal{X}|^2)$ for storing the similarity scores in memory. The specific Min-CC algorithm A used in CCBounds is the one proposed in [4], since it provides (under the probability constraint or the global weight bound stated in [20]) constant-factor approximation guarantee in expectation. Also, taking linear time in the size of the input graph, to the best of our knowledge, it is the most efficient algorithm in the Min-CC literature. As a result of this choice, the time complexity of CCBounds becomes $\mathcal{O}(|\mathcal{X}|^2|\mathcal{A}|)$.

Another appealing aspect of the fact that Fair-CC is an instance of Min-CC is that Fair-CC inherits the following theoretical result:

Theorem 1 ([20]). *If the condition* $\binom{|\mathcal{X}|}{2}^{-1} \sum_{u,v\in\mathcal{X}}(sim_{\mathcal{A}^{\neg F}}(u,v)+sim_{\mathcal{A}^F}(u,v)) \geq \max_{u,v\in\mathcal{X}} |sim_{\mathcal{A}^{\neg F}}(u,v) - sim_{\mathcal{A}^F}(u,v)|$ *holds on the similarity scores and the oracle* A *is an* α*-approximation algorithm for* Min-CC*,* CCBounds *is an* α*-approximation algorithm for* Fair-CC*.*

The above theorem provides approximation guarantee on the Fair-CC objective (cf. Eq. (2)), which combines the cluster quality measure (first summation) and the fairness-related objective (second summation). It is not known how this quality guarantee translates into the single objective, e.g., the fair objective. This is a challenging open question which we defer to future studies.

5 Fairness Evaluation

In this section, we summarize the most-commonly adopted metrics for the evaluation of fairness aspects in clustering. We focus on algorithm-independent measures, i.e., able to generalize across multiple methods, following a *group-level* approach under the *disparate impact doctrine* [16].

[3] `https://github.com/Ralyhu/globalCC`

Balance. It is one of the most adopted evaluation metrics for fairness in clustering, initially proposed by Chierichetti *et al.* [14] in a context with one sensitive attribute with two protected groups. It has been successively generalized to m protected groups by Bera *et al.* [8]. According to the latter, the balance of a clustering solution can formally be defined as follows [13]:

$$balance(\mathcal{C}) = \min_{C \in \mathcal{C}, b \in [m]} \min \left\{ R_{C,b}, \frac{1}{R_{C,b}} \right\} \in [0, 1], \tag{5}$$

where $R_{C,b}$ is the ratio between the proportion of the objects belonging to a given protected group b in the considered dataset and in a given cluster $C \in \mathcal{C}$.

In such a formulation, the lower and upper bounds of a cluster indicate the fully unbalanced and perfectly balanced scenarios, respectively, where the former indicates the case where all the objects in such a cluster pertain to the same protected group, whereas the latter denotes an equal number of objects from each of the protected groups. Therefore, the higher the balance, the better the obtained solution, in terms of equality. Additionally, the considered generalization allows us to obtain a comprehensive evaluation of the balance of our clustering solutions, as it looks at the dataset context, i.e., it will return high scores provided that the balances of the clustering and the input dataset are comparable.

Average Euclidean Fairness. This metric was introduced by Abraham *et al.* [1] to estimate the unfairness by assessing the deviation between the representation of groups obtained focusing on the sensitive attributes in the whole dataset and the given clustering solution. It expresses the cluster-size weighted average of cluster-level deviations (i.e., Euclidean distances) between two frequency (sensitive) attribute vectors, namely $\mathcal{X}_A$, which is computed over the entire set of objects, and C_A, which is computed for each cluster $C \in \mathcal{C}$, focusing on a sensitive attribute $A \in \mathcal{A}^F$. Formally, it is defined as:

$$AE_A(\mathcal{C}) = \frac{\sum_{C \in \mathcal{C}} |C| \times ED(C_A, \mathcal{X}_A)}{\sum_{C \in \mathcal{C}} |C|}, \tag{6}$$

where ED represents the Euclidean distance between the frequency attribute vectors. Since A can be multi-valued, such a formulation is suited to scenarios where there are multiple protected groups. Also, as this measure is a deviation, smaller values correspond to better solutions.

6 Experimental Methodology

6.1 Competing Methods

In the following, we briefly overview the competing methods we included in our experiments. For each of those methods, we used publicly available code, which we adopted "as-is", i.e., without making any changes or optimizations.

Fair Clustering Through Fairlets [14]. This method, here dubbed Fairlets, is one of the pillars of fair clustering. It is based on the notion of *fairlets* *decomposition*, that is a grouping of the input objects into *fairlets*, i.e., minimal subsets of objects that satisfy a given fairness definition, while preserving the clustering objective. Given a good fairlets decomposition, this approach requires traditional clustering algorithms (i.e., k-center or k-median) applied on the centers of the obtained fairlets, to yield the "fair" solutions. FAIRLETS supports two types of fairlets decomposition: an accurate one based on *min cost flow* (MCF), and a more efficient one. We hereinafter refer to those decompositions as *MCF decomposition* and *vanilla decomposition*, respectively. A major limitation of FAIRLETS is that it can handle a single sensitive binary attribute only. We will discuss the impact of such limitations in more detail in Section 7.

We involve FAIRLETS in our experimental evaluation by resorting to the unofficial implementation available online.[4]

HST-based Fair Clustering [6]. This approach, here dubbed HST-FC, focuses on the k-median formulation, and employs a quad-tree decomposition to embed the objects in a a tree metric, called *HST*. By leveraging such a tree, HST-FC computes an approximate fairlets decomposition. A fair clustering is ultimately obtained by running k-median algorithms on the produced fairlets. Like FAIRLETS, HST-FC suffers from the limitation that it deals with one binary sensitive attribute only.

In our experiments, we adopt the official implementation made available by the authors of HST-FC.[5]

Fair Correlation Clustering [3]. This method, here dubbed SIGNED, introduces a fairlet-based reduction for the graph clustering scenario with respect to the problem of correlation clustering, leading to the concept of correlation clustering with fairness constraints. Specifically, given a signed graph, i.e., an undirected graph with edges labeled as positive or negative, the algorithm performs a fairlet decomposition (under different fair settings) over the set of vertices. The produced decomposition is used, together with the original graph, to build a reduced (complete and unweighted) correlation clustering instance, where the vertices correspond to the produced fairlets and the sign of the edges between any two fairlets are built according to the majority sign of the edges between vertices within those two fairlets. A clustering on this reduced correlation clustering instance is computed through local-search optimization starting from all singleton clusters, and then expanded into a solution of the original problem. As a fair setting for the fairlets decomposition, we consider the most common case of fair decomposition where clusters are required not to have a sensitive data class. As the SIGNED method requires a signed graph as input, we perform the following preprocessing step to make the relational data compatible with this format. We derive a complete graph whose vertices are the original data objects and an edge (u, v) is labeled as positive with probability $p^+_{uv} = max\{0, sim_{\mathcal{A}^{\neg F}}(u, v) - sim_{\mathcal{A}^F}(u, v)\}$ and as a negative edge with probability $1 - p^+_{uv}$, where the similarity functions are the ones defined in Eqs. (3)–(4). We point out that, although we can adapt the same weighting strategy as CCBounds

[4] `https://github.com/guptakhil/fair-clustering-fairlets`

[5] `https://github.com/talwagner/fair_clustering`

Table 1: Overview of the datasets involved in our experiments.

	#objs.	*sensitive* attribute	*non-sensitive* attributes
Adult	48 842	sex	age, fnlgwt, education_num, capital_gain, hours_per_week
Bank	40 004	marital	age, balance, duration
CreditCard	10 127	sex	customer_age, dependent_count, avg_utilization_ratio, total_relationship_count
Diabetes	101 763	sex	age, time_in_hospital
Student	649	sex	age, study_time, absences

to obtain the edge attributes, we discarded this choice as our experiments showed that it favors the emergence of a degenerated clustering solution (i.e., a single output cluster), due to the strong predominance of positive weights on the edges.

In our evaluation, we use the official implementation made available by the authors of SIGNED.[6]

6.2 Data

We considered five real-world relational datasets, which have been commonly used in the fair clustering literature. The main characteristics of these datasets are summarized in Table 1. As reported in the table, in our evaluation we focused on a smaller subset of the original attributes; note that this is a common practice, which is adopted, among others, by the competing methods outlined above.

Adult.[7] This dataset reports information about the 1994 US Census. For each tuple representing an individual, we considered *age*, *fnlwgt*, *education-num*, *capital-gain* and *hours-per-week* as non-sensitive attributes, and *sex* (i.e., male or female) as a sensitive attribute.

Bank.[7] This provides details on phone calls involving direct marketing campaigns of a Portuguese banking institution to assess whether the bank term deposit will be subscribed or not. We considered attributes *age*, *balance* and *duration* as non-sensitive, and *marital status* (i.e., married or not) as sensitive.

CreditCard.[8] This dataset concerns customer credit card services to estimate customer attrition. We considered attributes *customer_age*, *dependent_count*, *avg_utilization_ratio* and *total_relation ship_count* as non-sensitive, and *sex* as sensitive.

Diabetes.[7] It reports diabetic patient records, for which we considered *age* and *time_in_hospital* as non-sensitive attributes, and *sex* as a sensitive attribute.

---

[6] https://github.com/google-research/google-research/tree/master/correlation_clustering

[7] https://archive.ics.uci.edu/ml/datasets/

[8] https://www.kaggle.com/sakshigoyal7/credit-card-customers

Student.[7] This dataset contains student performances for Mathematics and Portuguese language in secondary education of two Portuguese schools. We considered *age*, *study_time* and *absences* as non-sensitive, and *sex* as sensitive.

6.3 Evaluation Goals

Our evaluation objectives concern both fairness and quality aspects of clustering. In the first case, we use the fairness metrics defined in Section 5, which allow us to have a group-wide overview of how a method behaves in terms of fair principles. In the second case, we assess the quality of clustering by means of intra- and inter-clustering similarity, considering both the sensitive and non-sensitive attributes, as described below. Finally, we evaluate running times.

Intra/Inter-cluster similarity. As stated in Section 3, we take into account the intra-cluster, resp. inter-cluster, similarity among objects to properly distribute them into clusters, either focusing on their sensitive and non-sensitive attributes (cf. Eqs. (3) and (4)). We define the following aggregated scores to have an overall measure of goodness of the clusters:

$$inter(\mathcal{A}^{\neg F}) = \frac{1}{|\Theta|} \sum_{u,v \in \Theta} sim_{\mathcal{A}^{\neg F}}(u,v), \qquad inter(\mathcal{A}^{F}) = \frac{1}{|\Theta|} \sum_{u,v \in \Theta} sim_{\mathcal{A}^{F}}(u,v), \quad (7)$$

$$intra(\mathcal{A}^{\neg F}) = \frac{1}{|\Omega|} \sum_{u,v \in \Omega} sim_{\mathcal{A}^{\neg F}}(u,v), \qquad intra(\mathcal{A}^{F}) = \frac{1}{|\Omega|} \sum_{u,v \in \Omega} sim_{\mathcal{A}^{F}}(u,v), \quad (8)$$

where $\Omega = \{u, v \in \mathcal{X} \,|\, \mathcal{C}(u) = \mathcal{C}(v)\}$, and $\Theta = \{u, v \in \mathcal{X} \,|\, \mathcal{C}(u) \neq \mathcal{C}(v)\}$. In particular, to obtain fair clusters, we need to maximize (resp. minimize) the $inter(\mathcal{A}^F)$, resp. $intra(\mathcal{A}^F)$, scores, so that objects having the same set of *sensitive* attributes will not be clustered together, rather they will be well-distributed across clusters. Conversely, we require to maximize, resp. minimize, the $inter(\mathcal{A}^{\neg F})$, resp. $intra(\mathcal{A}^{\neg F})$, scores, to ensure that objects with the same set of *non-sensitive* attributes will be clustered close with each other and not scattered across different clusters.

Running times. We measure the running times of CCBounds and the competing methods while executing them on the *Cresco6* cluster.[9]

6.4 Hyper-parameters and Configurations

Data sampling and attributes selection. To test the selected competing methods under different conditions, and run even the most computationally expensive approaches, we adopt the sampling strategy proposed in [14]. Specifically, by sampling (without replacement) we extracted 1k or 10k tuples from the original full set of tuples, by preserving some desired ratio between the protected

[9] `https://www.eneagrid.enea.it`

Table 2: Configurations and hyper-parameters used in our evaluations w.r.t. different experimental setups. k_{avg} is the avg. number of clusters that were obtained over ten runs of CCBounds, and k corresponds to the parameter value provided to FAIRLETS and HST-FC.

	p, q	split ratio	k_{avg}	k
Adult-1k	1,2	650/350	3.12	3
Bank-1k	1,2	650/350	3.48	3
Credit-Card-1k	1,6	800/200	5.6	6
Diabetes-1k	1,2	540/460	5.2	5
Student-1k	1,2	266/383	3.88	4
Adult-10k	1,2	6 500/3 500	2.96	3
Bank-10k	1,2	6 500/3 500	3.28	3
Credit-Card-10k	1,6	4 769/5 358	6.32	6
Diabetes-10k	1,2	5 400/4 600	6.44	6
Adult-Full	2,5	32 650/16 192	3.64	4
Bank-Full	2,5	12 790/27 214	3.64	4
Diabetes-Full	1,2	47 055/54 708	OOM	6

classes. The details of the sampling strategy used in our experiments are reported in Table 2, where the selected fair attributes and split ratio (i.e., the fraction of tuples pertaining to different sensitive attribute values) are, whenever possible, the same as [14]. Also, both FAIRLETS and HST-FC require two integers p and q as input, whose ratio p/q corresponds to the minimum balance required by each clusters, yielded by these algorithms. The configuration of the aforementioned parameters, inspired by [14, 8], is reported in Table 2.

We highlight that, as described so far, we focus on a single and binary sensitive attribute to match the minimum requirements that embrace all competing methods. Nonetheless, some approaches (including our CCBounds) can deal with multiple values assigned to a single sensitive attribute.

Number of clusters. While FAIRLETS and HST-FC require a hyper-parameter k in input, denoting the desired number of output clusters, the same does not apply with the correlation clustering-based approaches. Thus, to create a reasonable comparative environment, we use the (rounded) average number of clusters returned by CCBounds in ten iterations as the k parameter for FAIRLETS and HST-FC. Moreover, we inherit the value k from the nearest subset when the correlation clustering-based approaches run out of memory.

7 Results

Table 3 summarizes the results achieved by CCBounds and the competing methods. With the exception of very high running times and out of memory errors (indicated with NA and OOM, respectively), all reported measurements correspond to averages over 10 runs of the tested algorithms. The similarity values (Eqs. (7)–(8)) were obtained by using Euclidean and Jaccard similarities for

Table 3: Summary of results according to the following criteria (columns from left to right): number of clusters, balance score, avg. Euclidean fairness, avg. intra-cluster and inter-cluster similarities according to either the set of selected sensitive attributes or the set of non-sensitive attributes (cf. Table 1), and running time. For each criterion, bold values correspond to the best-performing methods (possibly up to the second decimal point).

		$\#clust.$	balance $\uparrow$	AE $\downarrow$	$intra(\mathcal{A}^{\neg F})\uparrow$	$intra(\mathcal{A}^{F})\downarrow$	$inter(\mathcal{A}^{\neg F})\downarrow$	$inter(\mathcal{A}^{F})\uparrow$	time (s) $\downarrow$
Adult-1k	CCBounds	3.12	0.565	**0.007**	**0.685**	0.524	**0.415**	**0.334**	$<$ **1**
	Fairlets	3	0.805	**0.004**	0.585	**0.319**	0.596	**0.335**	$<$ **1**
	HST-FC	3	**0.971**	0.01	0.616	0.335	0.599	**0.336**	$<$ **1**
	Signed	41	0.66	0.03	0.59	**0.32**	0.60	**0.33**	240
Adult-10k	CCBounds	2.96	0.52	0.03	**0.65**	0.43	**0.43**	**0.33**	3.86
	Fairlets	3	0.82	**0.003**	0.60	**0.32**	0.615	**0.33**	$<$ **1**
	HST-FC	3	**0.98**	**0.006**	0.626	0.336	0.618	**0.336**	3.03
	Signed	NA	NA	NA	NA	NA	NA	NA	$>$ 48h
Adult-Full	CCBounds	3.64	0.56	**0.003**	**0.69**	0.47	**0.42**	0.24	75.5
	Fairlets	4	0.66	0.02	0.59	**0.32**	0.62	**0.34**	**6.5**
	HST-FC	4	**0.96**	**0.008**	0.63	0.34	0.62	**0.34**	72.86
	Signed	NA	NA	NA	NA	NA	NA	NA	$>$ 48h
Bank-1k	CCBounds	3.48	0.565	**0.006**	**0.727**	0.587	**0.441**	**0.369**	$<$ **1**
	Fairlets	3	0.828	**0.002**	0.606	**0.354**	0.613	**0.364**	$<$ **1**
	HST-FC	3	**0.968**	**0.007**	0.621	0.365	0.617	**0.365**	$<$ **1**
	Signed	41	0.7	0.03	0.61	**0.35**	0.63	**0.36**	224
Bank-10k	CCBounds	3.28	0.52	**0.0007**	**0.78**	0.63	**0.45**	**0.36**	4.74
	Fairlets	3	0.7	0.001	0.59	**0.32**	0.63	**0.36**	$<$ **1**
	HST-FC	3	**0.969**	0.004	0.656	0.365	0.656	**0.365**	3.07
	Signed	NA	NA	NA	NA	NA	NA	NA	$>$ 48h
Bank-Full	CCBounds	3.64	0.55	**0.0004**	**0.72**	0.55	**0.45**	**0.37**	51.1
	Fairlets	4	0.68	0.001	0.62	**0.34**	0.65	0.36	**5.3**
	HST-FC	4	**0.94**	0.008	0.66	0.37	0.66	**0.37**	28
	Signed	NA	NA	NA	NA	NA	NA	NA	$>$ 48h
CreditCard-1k	CCBounds	5.6	0.613	0.127	**0.6**	0.497	**0.46**	0.362	$<$ **1**
	Fairlets	6	0.4	0.042	0.485	**0.355**	0.486	**0.375**	$<$ **1**
	HST-FC	6	**0.756**	**0.026**	0.513	0.373	0.481	**0.377**	$<$ **1**
	Signed	171	0.56	0.1	0.56	0.41	0.49	**0.38**	173
CreditCard-10k	CCBounds	6.32	0.496	0.17	**0.6**	0.46	**0.46**	0.32	4.1
	Fairlets	6	0.94	**0.01**	0.497	**0.34**	0.49	**0.337**	$<$ **1**
	HST-FC	6	**0.955**	**0.013**	0.52	**0.337**	0.491	**0.337**	2.52
	Signed	NA	NA	NA	NA	NA	NA	NA	$>$ 48h
Diabetes-1k	CCBounds	5.2	0.45	0.33	**0.622**	0.519	**0.512**	0.352	$<$ **1**
	Fairlets	5	**0.92**	**0.015**	0.537	0.381	0.532	**0.385**	$<$ **1**
	HST-FC	5	0.872	0.05	0.585	0.386	0.529	**0.386**	$<$ **1**
	Signed	106	0.85	0.04	0.58	**0.36**	0.54	**0.38**	257
Diabetes-10k	CCBounds	6.44	0.48	0.22	**0.65**	0.54	**0.5**	0.36	4.72
	Fairlets	6	**0.92**	**0.01**	0.53	**0.38**	**0.53**	**0.39**	$<$ **1**
	HST-FC	6	0.799	0.065	0.59	**0.388**	**0.53**	**0.386**	2.84
	Signed	NA	NA	NA	NA	NA	NA	NA	$>$ 48h
Diabetes-Full	CCBounds	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	Fairlets	6	**0.93**	**0.01**	OOM	OOM	OOM	OOM	**22.2**
	HST-FC	6	0.81	0.06	OOM	OOM	OOM	OOM	761.2
	Signed	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
Student-1k	CCBounds	3.88	0.51	0.10	**0.625**	0.463	**0.471**	0.224	$<$ **1**
	Fairlets	4	0.82	**0.013**	0.528	**0.339**	0.543	**0.357**	$<$ **1**
	HST-FC	4	**0.93**	0.024	0.563	0.357	0.541	**0.358**	$<$ **1**
	Signed	55	0.82	0.04	0.57	**0.34**	0.55	**0.36**	71

numerical and categorical attributes, respectively. Moreover, as for the Fairlets method, as previously discussed in Section 6.1, we report results only for the vanilla fairlets decomposition, since the min-cost-flow (MCF) counterpart has very high running times (more than 7 minutes on the smallest dataset, i.e., *Student-1k*) and produces solutions that are very similar to the vanilla one (results not shown for the sake of brevity).

As for the balance, we notice that, although CCBounds does not match the high scores obtained by "fairness-native" methods (i.e., Fairlets and HST-FC), it is still able to score comparably with its direct competing method, i.e., Signed. Exceptions arise in the case of *Student-1k* and *Diabetes-1k*, where CCBounds sets up to lower scores, and for some large datasets, where Signed does not terminate in reasonable time, while our CCBounds still obtains good results in reasonable time. The paradigm shifts when we consider small yet heavily unbalanced datasets (i.e., *CreditCard-1k*, with an 80:20 ratio); here, although several competing methods struggle to obtain high scores, CCBounds achieves the second-best balance score. Overall, as the balance obtained by CCBounds in all evaluation scenarios ranges from 0.45 to 0.613, we can conclude that it is able of guaranteeing satisfactory balance scores.

In the case of avg. Euclidean fairness, CCBounds obtains very good scores under different scenarios: it is among the best-performer approaches for the *Adult-1k*, *Adult-Full* and *Bank-1k* datasets, and outperforms all the other methods by an order of magnitude on *Bank-10k* and *Bank-Full*. Conversely, CCBounds is unable to match the best scores obtained by some of the competing methods when focusing on the remaining datasets.

Considering the similarity computed on the sensitive attributes, CCBounds does not achieve the best intra-cluster similarity, meaning that it tends to group a few more objects with the same sensitive attribute value than the other methods. Nevertheless, the inter-cluster similarities are comparable with the other methods, thus indicating that CCBounds is still able to properly separate the objects into clusters, when accounting for the sensitive attribute. Instead, when we focus on the similarity computed on the non-sensitive attributes, CCBounds achieves the best performance in all the considered evaluation scenarios, yielding very high-quality clusters.

Finally, we also investigated on running times, spotting Fairlets as the best performer, followed by HST-FC and CCBounds, which both guarantee reasonable running times. Although CCBounds has quadratic time complexity due to pairwise similarity calculations (cf. Section 4), we managed to perform in parallel such time-consuming steps. On the contrary, Signed requires excessively long execution times, often resulting infeasible in practice, along with an abnormal number of clusters produced, which is particularly large even when considering the smallest *1k* datasets. Overall, it should be noted that, albeit the observed running times should be taken with grain of salt due to the (lack of) code optimizations, major remarks are consistent with the time complexities of the corresponding methods.

Discussion. A number of remarks arise from our experimental evaluation. First, although native fairness-aware approaches are able to produce clustering solutions that optimize fairness notions, we found out that such a capability comes with a cost, as the produced clusters are often far from being qualitatively good. On the other hand, CCBounds demonstrated itself to be effective and versatile: it was recognized as the best-in-case approach among the tested ones when it comes to find good-quality clusters, while also being able not to excessively penalize aspects related to fairness.

Second, although we unveiled the weakness in quality shown by the native fair-clustering approaches, we nonetheless shed light on how the approaches based on correlation clustering might suffer from computational issues, by being slower than the other methods, and requiring more memory. This is particularly evident with SIGNED, as it is unable to terminate in all datasets having more than 10k tuples, while it is kept under control in CCBounds, which goes down only in the case of *Diabetes-Full* (containing more than 100k tuples, cf. Section 6.2), thanks to the numerous optimization adopted under the hood. However, such a dataset makes it difficult to calculate similarities even for traditional and more efficient approaches, despite the computing capabilities at our disposal.

Finally, by wearing the lens of our proposed approach, we can state that it is able to provide performance in terms of fairness-aware metrics that are comparable to its direct competitor (i.e., SIGNED), but, at the same time, it manages to overcome all the state-of-the-art competing methods considered in our assessment, when it comes to generating qualitatively good clusters, anyway preserving aspects of fairness as much as possible.

8 Conclusions

In this paper, we analyzed how a correlation clustering method, called CCBounds, can profitably be used for the problem of fair clustering. Experimental evidence on real data has shown the meaningfulness of the clustering solutions produced by CCBounds, also revealing its ability of yielding clusters of higher quality than the considered competing methods, according to classic clustering-validation criteria, without discarding aspects of fairness.

In the future, we plan to further evaluate the performance of CCBounds under other conditions, e.g., multiple protected values. Also, we aim to investigate on alternative definitions of the similarity functions and push forward the capabilities of CCBounds towards more challenging scenarios, such as embracing multiple sensitive attributes with many values, allowing us to align with more realistic use cases, and strengthen the versatility of the correlation clustering under fairness constraints.

References

1. Abraham, S.S., P, D., Sundaram, S.S.: Fairness in clustering with multiple sensitive attributes. In: Proc. EDBT Conf. pp. 287–298 (2020)
2. Ahmadian, S., Epasto, A., Knittel, M., Kumar, R., Mahdian, M., Moseley, B., Pham, P., Vassilvitskii, S., Wang, Y.: Fair hierarchical clustering. In: Proc. NIPS Conf. (2020)
3. Ahmadian, S., Epasto, A., Kumar, R., Mahdian, M.: Fair correlation clustering. In: Proc. AISTATS Conf. pp. 4195–4205 (2020)
4. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: ranking and clustering. In: Proc. ACM STOC Symp. pp. 684–693 (2005)

5. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: Ranking and clustering. JACM **55**(5), 23:1–23:27 (2008)
6. Backurs, A., Indyk, P., Onak, K., Schieber, B., Vakilian, A., Wagner, T.: Scalable fair clustering. In: Proc. ICML Conf. pp. 405–413 (2019)
7. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Mach. Learn. **56**(1), 89–113 (2004)
8. Bera, S.K., Chakrabarty, D., Flores, N., Negahbani, M.: Fair algorithms for clustering. In: Proc. NIPS Conf. pp. 4955–4966 (2019)
9. Bercea, I.O., Groß, M., Khuller, S., Kumar, A., Rösner, C., Schmidt, D.R., Schmidt, M.: On the cost of essentially fair clusterings. In: Proc. APPROX/RANDOM Conf. pp. 18:1–18:22 (2019)
10. Charikar, M., Guruswami, V., Wirth, A.: Clustering with qualitative information. In: Proc. IEEE FOCS Symp. pp. 524–533 (2003)
11. Charikar, M., Guruswami, V., Wirth, A.: Clustering with qualitative information. JCSS **71**(3), 360–383 (2005)
12. Chawla, S., Makarychev, K., Schramm, T., Yaroslavtsev, G.: Near optimal LP rounding algorithm for correlation clustering on complete and complete k-partite graphs. In: Proc. ACM STOC Symp. pp. 219–228 (2015)
13. Chhabra, A., Masalkovaitė, K., Mohapatra, P.: An overview of fairness in clustering. IEEE Access **9**, 130698–130720 (2021)
14. Chierichetti, F., Kumar, R., Lattanzi, S., Vassilvitskii, S.: Fair clustering through fairlets. In: Proc. NIPS Conf. pp. 5029–5037 (2017)
15. Demaine, E.D., Emanuel, D., Fiat, A., Immorlica, N.: Correlation clustering in general weighted graphs. TCS **361**(2-3), 172–187 (2006)
16. Feldman, M., Friedler, S.A., Moeller, J., Scheidegger, C., Venkatasubramanian, S.: Certifying and removing disparate impact. In: Proc. ACM KDD Conf. pp. 259–268 (2015)
17. Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., Mullainathan, S.: Human decisions and machine predictions. The Quarterly Journal of Economics **133**(1), 237–293 (2017)
18. Kleindessner, M., Awasthi, P., Morgenstern, J.: Fair k-center clustering for data summarization. In: Proc. ICML Conf. pp. 3448–3457 (2019)
19. Kleindessner, M., Samadi, S., Awasthi, P., Morgenstern, J.: Guarantees for spectral clustering with fairness constraints. In: Proc. ICML Conf. pp. 3458–3467 (2019)
20. Mandaglio, D., Tagarelli, A., Gullo, F.: Correlation clustering with global weight bounds. In: Proc. ECML-PKDD Conf. pp. 499–515 (2021)
21. Martorelli, M., Jayatilake, S.M.D.A.C., Ganegoda, G.U.: Involvement of machine learning tools in healthcare decision making. J. Healthc. Eng. (2021)
22. Mashrur, A., Luo, W., Zaidi, N.A., Robles-Kelly, A.: Machine learning for financial risk management: A survey. IEEE Access **8**, 203203–203223 (2020)
23. Rösner, C., Schmidt, M.: Privacy preserving clustering with constraints. In: Proc. ICALP Colloq. pp. 96:1–96:14 (2018)
24. Schmidt, M., Schwiegelshohn, C., Sohler, C.: Fair coresets and streaming algorithms for fair k-means. In: Proc. WAOA Work. pp. 232–251 (2019)
25. Shamir, R., Sharan, R., Tsur, D.: Cluster graph modification problems. Discret. Appl. Math. **144**(1-2), 173–182 (2004)
26. Swamy, C.: Correlation clustering: maximizing agreements via semidefinite programming. In: Proc. ACM-SIAM SODA Conf. pp. 526–527 (2004)
27. van Zuylen, A., Williamson, D.P.: Deterministic algorithms for rank aggregation and other ranking and clustering problems. In: Proc. WAOA Work. pp. 260–273 (2007)