Fair Associative Co-Clustering

Federico Peiretti[0000−0001−7648−162X] and Ruggero G. Pensa[0000−0001−5145−3438]

University of Turin, Dept. Computer Science, Turin, Italy
{federico.peiretti,ruggero.pensa}@unito.it

Abstract. Co-clustering is a powerful data mining tool that extracts summary information from a data matrix, by simultaneously computing row and column clusters that provide a compact representation of the data. However, if the matrix contains data about individuals, the co-clustering results may be influenced by the societal biases that are reproduced in the data. Consequently, subsequent tasks such as recommendation systems may also be influenced by these biases, thereby compromising the fairness and integrity of the overall knowledge discovery or machine learning process. Despite the extensive research on fairness considerations in clustering, this issue has not been addressed in the context of co-clustering algorithms. In addressing this critical gap in the literature, this paper proposes a novel fair co-clustering algorithm. The proposed algorithm is based on an associative measure derived from the Goodman-Kruskal's *tau*, which has demonstrated good convergence properties. This ensures optimal clustering and fairness performance by implementing an in-process rebalancing mechanism inspired by the fair assignment problem. An extensive experimental validation is provided to demonstrate the efficacy of our approach, also in comparison to a state-of-the-art method that uses co-clustering for fair recommendation.

Keywords: Clustering · Fairness · High-dimensional data.

1 Introduction

Clustering results, as well as those of any other machine learning tasks, can be affected by the presence of any sort of bias in the data. When the data are related to human beings, and clustering is used to drive some critical decision process, such bias could lead to unfair or discriminatory outcomes towards minority groups or protected categories, a situation known as disparate impact. To address this issue, fair clustering has recently emerged as a solution aimed at mitigating the effects of existing biases in the data [13]. Some examples of fair methods for clustering include the balanced representation [14,7,8], the proportionally fair clustering [11] and the equitable distance fairness [10]. These algorithms has shown their effectiveness in identifying a trade-off between fairness and cluster quality in standard scenarios with relatively low-dimensional data samples. However, when dealing with high-dimensional data, most distance-based clustering techniques struggle to identify actual patterns in the data, due to the effects of

the well-known phenomenon of the curse of dimensionality. To cope with this issue, several classes of solutions have been proposed, including resorting to more robust definitions of distances, using some dimensionality reduction approach such as PCA or non-negative matrix factorization, learning a lower-dimensional representation or adopting clustering methods specifically tailored for large data matrices. Among the latter, co-clustering (the simultaneous partitioning of rows and columns of a data matrix) has shown its effectiveness in many challenging scenarios, with different forms of data distributions and matrix sparsity [6]. Co-clustering has another advantage: the partition on columns provides explanatory patterns for the row clustering, and vice-versa, thus making co-clustering an intrinsic interpretable unsupervised task. Unfortunately, co-clustering is even more seriously concerned by fairness issues than clustering. In fact, biases could affect either the row or the column partitioning, or even both. Consider, for instance, a user $\times$ movie matrix recording the ratings given by each user to some movie. Co-clustering can be used to group together similar users (exhibiting similar preferences) and similar movies (liked by similar users). If the outcome of the co-clustering are used to perform movie recommendation to users, suggestions might reflect societal biases present in the data and, consequently, be deeply unfair. Worse than that, such suggestions may contribute to the reinforcement of prejudices on demographic categories of people, thus making data even more biased. Although fair recommendation has been extensively addressed [30], it is worth noting that co-clustering is a more general technique that can be used in different data analysis pipelines or knowledge discovery processes, such as text mining [12], transfer learning [29], object detection, image segmentation and scene categorization [26]. Despite its wide employment, to our knowledge, the problem of bias mitigation in co-clustering has never been studied as such. The only most similar approach uses co-clustering within a fair recommendation framework [19]. However, while the whole process ensures unbiased recommendations, the preliminary co-clustering process is not entirely fair.

To fill this gap in the fair clustering literature, we propose a fair co-clustering algorithm based on an associative measure known as the de-normalized Goodman-Kruskal's τ, that has good convergence properties and does not require the final number of co-clusters to be defined *a priori*. These features can be exploited to adapt the co-clustering results to meet both partitioning quality and fairness requirements, without being too much constrained by a ill-defined number of clusters, and enable us to design an in-process rebalancing mechanism inspired by the fair assignment problem. We show experimentally that our approach is effective in identifying fair co-clusters that mitigate the disparate impact and, at the same time, still preserve a good quality. We also derive some interesting insights on the tradeoff between fairness and clustering performance: by slightly relaxing the balance constraint, our approach enables the achievement of reasonable partitioning results. Additionally, we compare our algorithm with a competitor that performs latent block model for fair recommendation and uses a fairer optimization that could be used, in theory, to obtain unbiased co-clusters.

However, we show that this is not sufficient to pursue our goal, thus making our approach the first truly fair co-clustering method.

2 Related work

Co-clustering is a data mining technique that simultaneously clusters rows and columns in a data matrix, particularly well-suited for high-dimensional datasets. Unlike traditional clustering, co-clustering optimizes a joint objective across both dimensions, revealing and exploiting latent structures. For a more comprehensive overview of co-clustering, we refer the reader to [6]. Despite the many existing extensions of such technique, fairness in co-clustering has not been directly explored in the extant literature. The closest related work, to the best of our knowledge, is a Gaussian latent block model (a class of methods largely used in the co-clustering literature) with an ordinal regression for providing fair recommendations independent of protected groups, thereby ensuring statistical parity [19].

Many studies, instead, have sought to incorporate fairness considerations into clustering methodologies, with a predominant focus on group fairness in center-based clustering. Chierichetti *et al.* [14] pioneered the notion of fairness in clustering by introducing fairlets, minimal sets with a balanced representation of different demographic groups that serve as building blocks for larger fair clusters. Subsequent research by Bera *et al.* [7] has expanded this concept to encompass multiple, non-binary protected attributes, offering approximation algorithms for center-based objectives. Despite the efficacy of these methods, scalability challenges emerged, prompting the development of optimizations such as near-linear time fairlet decomposition [4]. Other works extended fairness to hierarchical [1], spectral [27] and correlation clustering [3]. Alternative notions of fairness include proportionally fair clustering [11], in which every sufficiently large group of points is entitled to its own cluster center. Gupta et al. [22] have proposed the concept of τ-ratio fairness, which aims to achieve a balance between proportionality and efficiency through round-robin algorithms. Several studies have explored alternative fairness constraints and optimization strategies. Some of those [2,7] investigate methods to avoid under- and over-representing any specific group in a cluster. Others [8,16], analyze the cost of essentially fair clustering, providing theoretical bounds on the price of fairness.

The concept of group fairness emphasizes demographic parity across clusters, whereas individual fairness ensures equitable treatment of each individual with respect to the treatment of other's or their own specific needs [15]. Chakrabarti *et al.* [10] proposed α-equitable k-center clustering, ensuring that individuals receive a comparable level of service quality, while Brubach *et al.* [9] defined pairwise fairness and community preservation, capturing the scenario in which individuals benefit from being clustered together. Other approaches include [28], which learns fair and clustering-favorable representations for clustering simultaneously, in the context of visual learning. Finally, Ghodsi *et al.* [20] introduced a symmetric non-negative matrix tri-factorization model with contrastive fairness regularization that achieves balanced and cohesive clusters.

3 Background and motivation

This section delves into fundamental concepts related to fairness and co-clustering, essential for understanding the functionality of our proposed fair co-clustering algorithm. Additionally, we present an example that highlights the necessity of computing co-clustering outcomes in a fair manner.

3.1 Fair clustering

Fair clustering is a rapidly evolving field within algorithmic fairness in unsupervised learning, aiming to prevent clustering algorithms from favoring specific demographics. A prominent fairness notion in clustering is balance, initially introduced by Chierichetti *et al.* for two protected groups (e.g., Male and Female) [14]. Balance ensures that each cluster has an approximately equal number of points from both groups, enforcing the notion of disparate impact. Bera *et al.* generalize the balance to accommodate multiple protected groups by ensuring that the ratio of points from each group in every cluster matches the overall dataset ratio [7]. They define balance as follows:

Definition 1 (Balance). *The balance of a clustering $\mathcal{C}$ is defined as:*

$$balance(\mathcal{C}) = \min_{C_j \in \mathcal{C}, g \in \mathcal{G}} \min \left(\frac{r_g}{r_g(C_j)}, \frac{r_g(C_j)}{r_g} \right) \tag{1}$$

where $\mathcal{G}$ is the set of protected groups, r_g is the ratio of the group $g \in \mathcal{G}$ in the dataset X, $r_g(C_j)$ is the ratio of the group $g \in \mathcal{G}$ in cluster C_j, i.e.,

$$r_g = \frac{|X_g|}{|X|} \quad ; \quad r_g(C_j) = \frac{|X_g(C_j)|}{|X(C_j)|}$$

In this paper, we use the definition given by Gupta *et al.* [22]. They introduce the notion of τ-ratio fairness, which ensures that each cluster contains a predefined fraction of points for each protected attribute value. This approach necessitates *a priori* knowledge of the dataset demographic composition and accommodates multi-valued protected attributes. Notably, τ-ratio fairness represents a strict generalization of the balance property, enabling nuanced adjustments between clustering efficiency and fairness objectives. Its interpretability and direct evaluability from clustering outputs distinguish it from traditional balance fairness, as it prioritizes the maintenance of specific proportional ratios rather than pairwise attribute balancing.

Definition 2 (τ-ratio fairness). *Let $\boldsymbol{\tau} = (\tau_g)_{g=1}^{|\mathcal{G}|}$ be a vector, where $\tau_g \in [0, \frac{1}{k}]$ for all protected groups $g \in \mathcal{G}$. A clustering solution satisfies τ-ratio fairness if, for each cluster C_j and each protected group g, the number of points belonging to the group g in C_j is at least $\tau_g n_g$, where n_g denotes the total number of points belonging to group g in the dataset, i.e.,*

$$|X_g(C_j)| \geq \tau_g n_g \quad with \quad \tau_g \in \left[0, \frac{1}{k}\right] \tag{2}$$

We denote the number of clusters with k. Specifically, when $\tau_g = 1/k$ for all demographic groups g and all clusters have similar size, the definition is equivalent to Definition 1. To avoid any potential ambiguity, we will henceforth refer to τ_g as γ_g.

This notion of fairness is well-suited to high-dimensional data because a solution can be found using greedy round-robin algorithms, which can handle a large number of clusters and data without having to explore too wide a space of solutions with only an additional time complexity $O(kn\log(n))$.

3.2 Fast Co-clustering

Fast-τCC [5] is a recent co-clustering algorithm that has good convergence properties and is also able to identify a congruent number of clusters on rows and columns, starting from an initial overestimation. Given a data matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}_+^{n\times m}$, a co-clustering of $\mathbf{A}$ is a pair $(\mathcal{R}, \mathcal{C})$, where $\mathcal{R}$ is a partition of the rows and $\mathcal{C}$ a partition of the columns of the matrix. The objective function of Fast-τCC is derived from the Goodman and Kruskal's τ [21], and can be defined as follows:

$$\hat{\tau}_{R|C}(\mathcal{R}, \mathcal{C}) = \sum_{k=1}^{|\mathcal{R}|}\sum_{l=1}^{|\mathcal{C}|}\frac{t_{kl}^2}{T \cdot t_{\cdot l}} - \sum_{k=1}^{|\mathcal{R}|}\frac{t_{k\cdot}^2}{T^2} \tag{3}$$

where $\mathbf{T} = (t_{kl})$ is the contingency table associated to the co-clustering $(\mathcal{R}, \mathcal{C})$, where $\mathcal{R} = (\mathcal{R}_1, \ldots, \mathcal{R}_K)$ and $\mathcal{C} = (\mathcal{C}_1, \ldots, \mathcal{C}_L)$, i.e. $t_{kl} = \sum_{i\in\mathcal{R}_k}\sum_{j\in\mathcal{C}_l} a_{ij}$, for $k = 1, \ldots, K$ and $l = 1, \ldots, L$. Following this notation, $t_{k\cdot} = \sum_{l=1}^{L} t_{kl}$, $t_{\cdot l} = \sum_{k=1}^{K} t_{kl}$ and $T = \sum_{k=1}^{K}\sum_{l=1}^{L} t_{kl}$. Analogously, the association of the column clustering $\mathcal{C}$ to the row clustering $\mathcal{R}$ can be evaluated through the function $\hat{\tau}_{C|R}(\mathcal{R}, \mathcal{C})$. Since $\hat{\tau}$ is not symmetric, the best co-clustering solutions are those that simultaneously maximize $\hat{\tau}_{R|C}$ and $\hat{\tau}_{C|R}$. In [5] an iterative optimization strategy is introduced. It alternates the computation of $\hat{\tau}_{R|C}$ by fixing the column partition and the computation of $\hat{\tau}_{C|R}$ by keeping the row partition fixed.

3.3 Unfairness in co-clustering

Co-clustering can lead to unfair outcomes when applied without proper consideration of sensitive group attributes. To show this behavior, a simplified example is provided. The scenario under consideration is the one of movie ratings. Let's $\mathbf{A}$ be a data matrix whose rows represent users, columns represent movies, and the entries represent ratings from 0 to 5. Users are identified by two protected groups (u_0, u_1, u_4 users from group g_0 and u_2, u_3, u_5 from group g_1) and movies are categorized by genres (e.g., m_0 movie is Action, m_1 is Comedy, m_2 is Dramatic, m_3 is Horror, m_4 is Romantic).

A co-clustering algorithm will likely find the following clustering of users and movies: $\mathcal{R} = \{\{u_0, u_1\}, \{u_3, u_4\}, \{u_2, u_5\}\}$ and $\mathcal{C} = \{\{m_0, m_4\}, \{m_1, m_2\}, \{m_3\}\}$.

$$\mathbf{A} = \begin{array}{c|cc|cc|cc} & m_0 & m_4 & m_1 & m_2 & m_3 & & \textbf{group} \\ \hline u_0 & 5 & 4 & 0 & 0 & 0 & u_0 & g_0 \\ u_1 & 4 & 0 & 0 & 0 & 0 & u_1 & g_0 \\ \hline u_3 & 0 & 0 & 5 & 4 & 0 & u_3 & g_1 \\ u_4 & 0 & 0 & 0 & 5 & 0 & u_4 & g_0 \\ \hline u_2 & 0 & 0 & 2 & 0 & 5 & u_2 & g_1 \\ u_5 & 0 & 5 & 0 & 0 & 4 & u_5 & g_1 \end{array}$$

(a) Unfair co-clustering

$$\mathbf{A} = \begin{array}{c|cc|cc|cc} & m_0 & m_3 & m_1 & m_2 & m_4 & & \textbf{group} \\ \hline u_0 & 5 & 0 & 0 & 0 & 4 & u_0 & g_0 \\ u_5 & 0 & 4 & 0 & 0 & 5 & u_5 & g_1 \\ \hline u_1 & 4 & 0 & 0 & 0 & 0 & u_1 & g_0 \\ u_2 & 0 & 5 & 2 & 0 & 0 & u_2 & g_1 \\ \hline u_3 & 0 & 0 & 5 & 4 & 0 & u_3 & g_1 \\ u_4 & 0 & 0 & 0 & 5 & 0 & u_4 & g_0 \end{array}$$

(b) Perfectly balanced co-clustering

Fig. 1: A toy example with an unfair optimal solution (left) and its fair solution (right) with respect to row clustering.

(see Fig. 1a). This solution, while reflecting the rating patterns in the data, reinforces existing societal biases by grouping users based on demographic characteristics. In fact, the user clustering exhibits a segmentation into three distinct clusters, one of these consisting of all users from protected group g_0, another one encompassing all users in demographic group g_1. If this unfair solution is used in a recommender system, it can lead to unfair and limited recommendations, as users from g_0 will be primarily recommended action and romantic movies, while users from g_1 will receive horror movie suggestions. This reduces the likelihood that individuals will discover movies outside their stereotypical preferences, potentially missing out on content they would enjoy.

In order to identify a fair clustering of users, where each cluster is equally represented by each of the protected groups, according to Definition 1, it is necessary to ensure that each cluster contains a proportion of data points for each protected group equal to the proportion of data points for each group in the entire dataset. A potential fair solution could be the row clustering $\mathcal{R} = \{\{u_0, u_5\}, \{u_1, u_2\}, \{u_3, u_4\}\}$ shown in Fig. 1b. This is perfectly balanced, as the proportion of both groups in each cluster is exactly equal to their dataset ratios, thus resulting in balance that is equal to 1. In this case, ensuring a perfect balance leads to a limited loss of information. In fact, the objective function on rows $\tau_{R|C}$ only degrades from 0.62 to 0.55.

4 Fair Co-Clustering

In this section, we present Fair-τCC, a fair co-clustering method based on the de-normalized Goodman-Kruskal's τ (see Eq. 3). We first define the problem of fairness in co-clustering, then describe the algorithm for computing the co-clustering results in a fair manner.

Definition 3 (Fair Co-clustering). *Given a data matrix* $\mathbf{A}$ *and protected groups* $\mathcal{G}_{rows} = \{g_0, \ldots, g_w\}$, $\mathcal{G}_{cols} = \{g_0, \ldots, g_z\}$ *referring to the row and col-*

umn objects respectively, a co-clustering $(\mathcal{R}, \mathcal{C})$ *is fair if both row and column clustering* $\mathcal{R}, \mathcal{C}$ *are fair.*

Drawing inspiration from the definition of balance for clustering [14,7], we define it for co-clustering tasks. Ideally, a co-clustering is balanced if, for each protected group associated with the row (column) objects, the ratio of its points in every row (column) clusters is the same as the ratio of its points over the whole dataset.

Definition 4 (Co-clustering Balance). *Let* S_{rows} *and* S_{cols} *be sensitive features associated with the row and column items, such that* $s_i^{rows} \in \mathcal{G}_{rows}$ *and* $s_j^{cols} \in \mathcal{G}_{cols}$*, where* $\mathcal{G}_{rows}$ *and* $\mathcal{G}_{cols}$ *are the protected groups the* i*-th row and* j*-th column items belong to, respectively. The balance of a co-clustering* $(\mathcal{R}, \mathcal{C})$ *is defined as:*

$$balance\left(\{\mathcal{R}, \mathcal{C}\}\right) = min\left(balance(\mathcal{R}), balance(\mathcal{C})\right) \tag{4}$$

with

$$balance(\mathcal{R}) = \min_{R_i \in \mathcal{R}} \min_{w \in \mathcal{G}_{rows}} \left(\frac{r_w}{r_w(R_i)}, \frac{r_w(R_i)}{r_g} \right), \tag{5}$$

$$balance(\mathcal{C}) = \min_{C_i \in \mathcal{C}} \min_{z \in \mathcal{G}_{cols}} \left(\frac{r_z}{r_z(C_j)}, \frac{r_z(C_j)}{r_z} \right), \tag{6}$$

where r_w *and* r_z *are the ratios of the protected groups* w, z *in the dataset;* $r_w(R_i)$ *and* $r_z(C_j)$ *are the ratios of the protected groups* w, z *in each row cluster* R_i *and column cluster* C_j*.*

The protected groups for both row and column objects are not always known. Therefore, if only $\mathcal{G}_{rows}$ ($\mathcal{G}_{cols}$) is known, co-clustering is considered fair if the row (column) clustering is fair (i.e., $balance(\mathcal{R}) \approx 1$). For simplicity, in this work we ensure the fairness for only the protected groups of row objects.

4.1 Fair-τCC algorithm

We now introduce Fair-τCC, the fair adaptation of the current state-of-the-art co-clustering method proposed by Battaglia *et al.* [5]. The primary objective of this algorithm is to ensure balanced representation of each protected group in every row cluster. Specifically, it guarantees a minimum fraction of points from each protected group in every cluster, adhering to the concept of τ-ratio fairness (refer to Eq. 2), hereinafter referred as γ to avoid any ambiguity. The pseudocode for this algorithm is detailed in Algorithm 1, while the procedure for updating the row clustering is illustrated in Algorithm 2.

First, we must introduce two matrices $\mathbf{P} = (p_{ij})$ and $\mathbf{Q} = (q_{kl})$, with $p_{ij} = \frac{a_{ij}}{A}$ and $q_{kl} = \frac{t_{kl}}{A} = \sum_{i \in \mathcal{R}_k} \sum_{j \in \mathcal{C}_l} p_{ij}$, where A denotes the sum of all the entries of $\mathbf{A}$ (hence, $A = T$). We also introduce the row cluster incidence matrix $\mathbf{R} = r_{ik}$

Algorithm 1 Fair τCC($\mathbf{A}, \mathbf{s}, \mathcal{G}, K_0, L_0, t_{max}, \boldsymbol{\alpha}$)

Input: A $n \times m$ data matrix $\mathbf{A}$, a sensitive feature $\mathbf{s} = [s_0, \ldots, s_n]$, protected groups $\mathcal{G} = \{g_0, \ldots, g_w\}$, initial number of row and column clusters K_0 and L_0, max number of iterations t_{max}, a vector $\boldsymbol{\alpha} = [\alpha_0, \ldots, \alpha_w]$ with $\alpha_g \in [0, 1]$.
Result: $\mathbf{R}$, $\mathbf{C}$ row and column clustering such that $\mathbf{R}$ satifies γ-ratio fairness (Eq. 2).
Initialize $\mathbf{R}^{(0)}$ and $\mathbf{C}^{(0)}$;
$t \leftarrow 1$; *changes* $\leftarrow$ True;
compute $\mathbf{P}$ from $\mathbf{A}$;
while *changes* **and** $t < t_{max}$ **do**
  $\mathbf{R}^{(t)} \leftarrow$ FairUpdateRowClusters($\mathbf{P}, \mathbf{C}^{(t-1)}, \mathbf{R}^{(t-1)}, \mathbf{s}, \mathcal{G}, \boldsymbol{\alpha}$);
  $\mathbf{C}^{(t)} \leftarrow$ UpdateColumnClusters($\mathbf{P}, \mathbf{R}^{(t)}, \mathbf{C}^{(t-1)}$);
  if $\mathbf{R}^{(t)} = \mathbf{R}^{(t-1)}$ **and** $\mathbf{C}^{(t)} = \mathbf{C}^{(t-1)}$ **then**
    changes $\leftarrow$ False;
  end
  $t \leftarrow t + 1$;
end

and $\mathbf{C} = c_{jl}$, with $r_{ik} = 1$ if row i is in row cluster $\mathcal{R}_k$ ($r_{ik} = 0$ otherwise) and $c_{jl} = 1$ if column j is in column cluster $\mathcal{C}_l$. According to this notation,

$$\mathbf{Q} = \mathbf{R}^\top \mathbf{P} \mathbf{C} \tag{7}$$

Equation 3 can be then rewritten as:

$$\hat{\tau}_{R|C}(\mathcal{R}, \mathcal{C}) = \sum_{k=1}^{K} \sum_{l=1}^{L} \left(\sum_{i \in \mathcal{R}_k} \frac{p_{il}}{p_{\cdot l}} \right) q_{kl} - \sum_{k=1}^{K} \left(\sum_{i \in \mathcal{R}_k} p_{i\cdot} \right) q_{k\cdot}$$

where $q_{k\cdot} = \sum_{i \in \mathcal{R}_k} p_{i\cdot} = \sum_{l=1}^{L} \frac{t_{kl}}{A} = \sum_{l=1}^{L} \sum_{i \in \mathcal{R}_k} \sum_{j \in \mathcal{C}_l} \frac{a_{ij}}{A}$, and $q_{\cdot l} = p_{\cdot l} = \sum_{j \in \mathcal{C}_l} p_{\cdot j} = \sum_{k=1}^{K} \frac{t_{kl}}{A} = \sum_{k=1}^{K} \sum_{i \in \mathcal{R}_k} \sum_{j \in \mathcal{C}_l} \frac{a_{ij}}{A}$, $p_{i\cdot} = \sum_{j=1}^{m} \frac{a_{ij}}{A}$, $p_{\cdot j} = \sum_{i=1}^{n} \frac{a_{ij}}{A}$, $p_{il} = \sum_{j \in \mathcal{C}_l} p_{ij}$, and $p_{\cdot l} = \sum_{j \in \mathcal{C}_l} p_{\cdot j} = q_{\cdot l}$.

Let $\mathbf{R}^{(t)}$ be the row cluster incidence matrix at iteration t, and $\mathbf{Q}^{(t)} = \mathbf{R}^{(t)\top} \mathbf{P} \mathbf{C}$ its associated distribution. The objective function $\hat{\tau}_{R|C}(\mathcal{R}^{(t)}, \mathcal{C})$ is

$$\hat{\tau}_{R|C}(\mathcal{R}^{(t)}, \mathcal{C}) = \sum_{i=1}^{n} \left(\sum_{l=1}^{L} \frac{p_{il}}{p_{\cdot l}} q_{kl}^{(t)} - p_{i\cdot} q_{k\cdot}^{(t)} \right) \tag{8}$$

Each row $\mathbf{q}_k^{(t)}$ of $\mathbf{Q}^{(t)}$ can be interpreted as a prototype of the k-th cluster of $\mathcal{R}^{(t)}$, and the following similarity function between any row $\mathbf{p}_i$ of $\mathbf{P}$ and $\mathbf{q}_k^{(t)}$ is defined:

$$\sigma\left(\mathbf{p}_i, \mathbf{q}_k^{(t)}\right) = \sum_{l=1}^{L} \frac{p_{il}}{p_{\cdot l}} q_{kl}^{(t)} - p_{i\cdot} q_{k\cdot}^{(t)} \tag{9}$$

It measures the similarity between a "point" p_i and a cluster prototype $q_r^{(t)}$. The objective function becomes

$$\hat{\tau}_{R|C}(\mathcal{R}^{(t)}, \mathcal{C}) = \sum_{i=1}^{n} \sigma\left(\mathbf{p}_i, \mathbf{q}_{k^\star}^{(t)}\right) \tag{10}$$

Algorithm 2 FairUpdateRowClusters($\mathbf{P}, \mathbf{C}, \mathbf{R}^{(0)}, \mathbf{s}, \mathcal{G}, \boldsymbol{\alpha}$)

h[!ht] **Input:** A $n \times m$ matrix $\mathbf{P}$, column clustering $\mathbf{C}$, initial row clustering $\mathbf{R}^{(0)}$, sensitive feature $\mathbf{s} = [s_0, \ldots, s_n]$, protected groups $\mathcal{G} = \{g_0, \ldots, g_w\}$, fairness parameters $\boldsymbol{\alpha} = [\alpha_0, \ldots, \alpha_w]$ with $\alpha_g \in [0, 1]$.
Result: row clustering $\mathbf{R}$ that satisfies γ-ratio fairness (Eq. 2)
$t \leftarrow 1$; *changes* $\leftarrow$ True;
while *changes* **do**
  $\mathbf{Q}^{(t-1)} = \mathbf{R}^{(t-1)\top}\mathbf{P}\mathbf{C}$;
  compute $\mathbf{U}^{(t-1)}$ and $\mathbf{V}^{(t-1)}$ as in Eq. 12;
  $\boldsymbol{\Sigma} = \mathbf{P}\mathbf{C}(\mathbf{Q}^{(t-1)} \odot \mathbf{U}^{(t-1)} - \mathbf{V}^{(t-1)})^\top$;
  for $i = 1, \ldots, n$ **do**
    $k^\star(i) \leftarrow \arg\max_k(\sigma_{ik})$;
  end
  compute $\mathbf{R}^{(t)}$ using $k^\star$;
  remove empty clusters and update $\mathbf{R}^{(t)}$;
  if $\mathbf{R}^{(t)}$ *violates* γ*-ratio fairness* **then**
    $\mathbf{R}^{(t)}$ = FairRowAssignments($\mathbf{R}^{(t)}$, $\boldsymbol{\Sigma}$, $\mathbf{s}$, $\mathcal{G}$, $\boldsymbol{\alpha}$);
  end
  if $\mathbf{R}^{(t)} = \mathbf{R}^{(t-1)}$ **then**
    changes $\leftarrow$ False;
  end
  $t \leftarrow t + 1$;
end

where $k^\star = \arg\max_k\left(\sigma\left(\mathbf{p}_i, \mathbf{q}_k^{(t)}\right)\right)$ is the cluster assignment maximizing function σ.

Algorithm 2 uses two $K \times L$ matrices $\mathbf{U}$ and $\mathbf{V}$ to compute all σ values in a $n \times K$ matrix $\boldsymbol{\Sigma} = (\sigma_{ik})$, where $\sigma_{ik} = \sigma\left(\mathbf{p}_i, \mathbf{q}_k^{(t)}\right)$. More precisely:

$$\boldsymbol{\Sigma} = \mathbf{P}\mathbf{C}(\mathbf{Q}^{(t-1)} \odot \mathbf{U}^{(t-1)} - \mathbf{V}^{(t-1)})^\top \tag{11}$$

where $\odot$ indicates the Hadamard matrix product, and

$$\mathbf{U}^{(t)} = \begin{bmatrix} \frac{1}{\sum_l q_{1l}^{(t)}} & \cdots & \frac{1}{\sum_l q_{Kl}^{(t)}} \\ \vdots & \ddots & \vdots \\ \frac{1}{\sum_l q_{1l}^{(t)}} & \cdots & \frac{1}{\sum_l q_{Kl}^{(t)}} \end{bmatrix}, \qquad \mathbf{V}^{(t)} = \begin{bmatrix} \frac{1}{\sum_l q_{1l}^{(t)}} & \cdots & \frac{1}{\sum_l q_{1l}^{(t)}} \\ \vdots & \ddots & \vdots \\ \frac{1}{\sum_l q_{Kl}^{(t)}} & \cdots & \frac{1}{\sum_l q_{Kl}^{(t)}} \end{bmatrix} \tag{12}$$

Then, the algorithm also removes all empty clusters. Hence, from one iteration to another, the number of clusters may decrease and $\mathbf{R}^{(0)}$ and $\mathbf{C}^{(0)}$ can be initialized with random partitions using safely high values of K_0 and L_0.

Given this initial assignment, we evaluate whether the optimal solution $\mathbf{R}^*$ satisfies the γ-ratio fairness property. If it does not, a fair assignment $\mathbf{R}^{fair}$ is determined (see Algorithm 3). The trade-off between fairness and clustering quality is managed through the utilization of the similarity matrix $\boldsymbol{\Sigma}$. Let $\mathbf{s} =$

$[s_0, \ldots, s_n]$ denote the sensitive feature associated with the rows of the data matrix, where $s_i \in \mathcal{G}$ and $\mathcal{G} = \{g_0, \ldots, g_w\}$ represents the set of protected groups. From the similarity matrix $\mathbf{\Sigma}$, we derive a $n \times K$ matrix $\mathbf{D} = (d_{ik})$, defined as follows:

$$d_{ik} = \sigma(\mathbf{p}_i, \mathbf{q}_k^*) - \sigma(\mathbf{p}_i, \mathbf{q}_k) \quad \forall k = 1, \ldots, K \tag{13}$$

Here, $\sigma(\mathbf{p}_i, \mathbf{q}_k^*)$ indicates the similarity value between point p_i and its optimal cluster prototype q_k^*, while $\sigma(\mathbf{p}_i, \mathbf{q}_k)$ represents the similarity value between point p_i and an alternative cluster prototype q_k. Consequently, d_{ik} quantifies the loss in clustering quality when point p_i is allocated to cluster k instead of its optimal cluster k^*. To ensure optimal preservation of quality, it is important to determine the sequence in which cluster prototypes for each point should be evaluated and the sequence in which the points from the same protected group should be chosen. To do this, we sort the indices of the row vector $\mathbf{d}_i$, corresponding to the cluster prototypes of the point p_i, by value in ascending order. Then, for each protected group, we sort points by $\mathbf{d}_i$ in ascending order.

Algorithm 3 FairRowAssignments($\mathbf{R}^*, \mathbf{\Sigma}, \mathbf{s}, \mathcal{G}, \alpha$)

Input: The optimal row clustering $\mathbf{R}^*$, $n \times K$ similarity matrix $\mathbf{\Sigma}$, sensitive feature $\mathbf{s} = [s_0, \ldots, s_n]$, protected groups $\mathcal{G} = \{g_0, \ldots, g_w\}$, fairness parameters $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_w)$ with $\alpha_g \in [0, 1]$, $\forall g \in \mathcal{G}$.
Result: row clustering $\mathbf{R}^{fair}$ that satisfies γ-ratio fairness
Initialize $\mathbf{R}^{fair} = 0^{(n \times K)}$;
Compute $\mathbf{D}$ as in Eq. 13;
Sort cluster prototypes by d_{ij} values in ascending order, $\forall i = 1, \ldots, n$;
Sort row objects by protected group and then by d_{ij} value in ascending order;
for g *in* $\mathcal{G}$ **do**
  $\mathbf{A}_g = \{\mathbf{p}_i \in \mathbf{A}$ s.t. $\mathbf{s}_i = g\}$; $n_g = |\mathbf{A}_g|$; $\gamma_g = \frac{1}{K}\alpha_g$;
  for $iter = 1 \ldots \lfloor \gamma_g n_g \rfloor$ **do**
    for $j = 1 \ldots K$ **do**
      $\mathbf{p}_{min} = \arg\min_{\mathbf{p}_i \in \mathbf{A}_g : \sum_{j=1}^{K} r_{i,j}^{fair} = 0} (\sigma(\mathbf{p}_i, \mathbf{q}_{k*}) - \sigma(\mathbf{p}_i, \mathbf{q}_j))$;
      $r_{min,j}^{fair} = 1$;
    end
  end
  $\forall \mathbf{p}_i \in \mathbf{A}_g : \sum_{j=1}^{K} r_{i,j}^{fair} = 0, \quad r_{i,k*}^{fair} = r_{i,k*}^{*}$;
end

For each protected group g, a fraction of unassigned row items equivalent to $\gamma_g n_g$ is chosen for allocation to a non-optimal cluster with the aim of minimizing loss value and ensuring fairness. The parameter n_g denotes the number of points belonging to the protected group g. The fairness parameter $\gamma_g \in \left[0, \frac{1}{K}\right]$ is the fraction of n_g points to be allocated in each cluster. Specifically, it is defined as $\gamma_g = \frac{1}{K}\alpha_g$ where K represents the number of row clusters identified by the vanilla approach and $\alpha_g \in [0, 1]$ is a user-defined parameter that quantifies the desired

level of fairness. If $\alpha_g = 1.0$ for a group g, then the n_g points will be equally distributed across K clusters ($\frac{n_g}{K}$ points in each cluster) and the group's ratio in each cluster matches its ratio in the overall dataset. Conversely, if $\alpha_g = 0.0$ for a group g, fairness violation is permitted for that group. If all groups have their parameters set to zero ($\alpha_g = 0.0, \forall g \in \mathcal{G}$), any solution is acceptable, allowing for selection of the optimal row clustering. Notably, if $\alpha_g = 1.0$ for all groups, row clustering achieves perfect balance ($balance(\mathcal{R}) \approx 1.0$), otherwise with $\alpha_g = 0.8$ the 80% rule of disparate impact doctrine is guaranteed. Finally, any points that remain unallocated at the end of this procedure are assigned to their optimal cluster.

5 Experiments

In this section, we present the findings from our experiments conducted on four real-world datasets to evaluate the effectiveness of Fair-τCC. We first present the dataset used, the competing methods and the experimental protocol. Then, we delve into the results and discuss them.

5.1 Experiment design

The datasets most commonly employed for fairness assessment (e.g. Adults, German credit, Banks, etc.) are low-dimensional and, consequently, are not well-suited to our experiments: co-clustering, in fact, identifies subsets of rows and columns in a high-dimensional data matrix that exhibit meaningful patterns. Furthermore, the majority of the benchmark datasets utilized for assessing co-clustering results lack any sensitive information. Following a thorough examination of available datasets, we identified four of them as being suitable for our purposes. These include two ratings dataset (MovieLens-1M [23] and Yelp [17,18]), a product reviews dataset (Amazon reviews [17,18]), and an image collection for facial recognition (Labeled Faces in the Wild [24]). Table 1 summarizes the characteristics of the data matrix for each dataset utilized in our experiments. Additionally, Table 2 provides details on the sensitive features selected for each dataset, including the protected groups and their respective proportions within the datasets.

Table 1: Datasets used for the experiments

Dataset	Size	Rows	Columns	Values	Labels	Sens. Att.
MovieLens-1M	6040×3645	users	movies	ratings	genres	gender; age
Yelp	1441×333	users	restaurants	ratings	category	gender
Amazon	705×10152	reviews	words	frequencies	prod. categories	gender
LFW	13233×1850	face images	features	RGB value	people IDs	gender

Table 2: Information about protected groups and their ratio in the whole dataset.

Dataset	Sens. Att.	Protected groups	Group ratio
MovieLens-1M	gender	Male; Female	72%; 28%
MovieLens-1M	age	< 35; 35-50; > 50	57%; 29%; 14%
Yelp	gender	Male; Female	38%; 62%
Amazon	gender	Male; Female	39%; 61%
LFW	gender	Male; Female	78%; 22%

We compared our algorithm against the standard version of Fast-τCC, which does not incorporate fairness constraints[1] and the only closely related competitor, Parity LBM [19], a Gaussian latent block model with an ordinal regression designed for fair recommendations independent of protected attributes[2]. To assess the performance of each algorithm regarding co-clustering quality and fairness, we employed several evaluation metrics:

- $\boldsymbol{\tau_{R|C}}$ and $\boldsymbol{\tau_{C|R}}$ [21]: the Goodman-Kruskal's τs measuring the quality of co-clustering computed by every algorithms.
- **ARI** [25]: the Adjusted Rand Index. It is used to compare the agreement between row and column assignments predicted by the fair algorithms with those from the corresponding vanilla approach (ARI_{rows} and ARI_{cols} in Table 3). Additionally, it is used to compute the aggrement between the clustering and the given ground-truth labels detailed in Table 1 (ARI in Table 3).
- **Balance** [7]: This metric quantifies the balanced representation of protected groups within each cluster according to Definition 1.
- **Kullback-Leibler fairness error** [31]: It is based on Kullback-Leibler divergence and quantifies the distributional disparities between the predefined target demographic proportions and the marginal probabilities of the demographics within cluster. It attains its theoretical minimum of zero iff all clusters exhibit demographic proportions identical to the target distribution, thereby enforcing strict adherence to the specified fairness constraints.

We executed 10 iterations of each algorithm. The mean values and standard deviations for all metrics are reported.

To evaluate the effectiveness of our algorithm, we set the number of initial clusters for both rows and columns to $K_0 = 10$ and $L_0 = 10$, respectively. Furthermore, for adjusting the trade-off between the level of fairness and co-clustering efficiency, we launched the experiments varying all α values within the range $[0, 1]$ for all protected groups. Conversely, Parity LBM was executed with hyperparameters configured as follows: 25 row and column clusters to be found for the MovieLens dataset and 10 for all others; a maximum number of 300 epochs for the training; a batch size of (200,200) and a learning rate of 2e-2. After

[1] https://github.com/rupensa/tauCC

[2] https://github.com/jackmedda/C-Fairness-RecSys/tree/main/reproducibility_study/Frisch_et_al

Table 3: Summary of the results for all co-clustering algorithms.

Algorithm	$\tau_{R\|C}$	$\tau_{C\|R}$	ARI	ARI_{rows}	ARI_{cols}	Balance	KL fairness
ML (gender)							
Fast-τCC	**0.109 ± 0.011**	**0.104 ± 0.013**	**0.090 ± 0.024**	-	-	0.787 ± 0.046	0.018 ± 0.007
Fair-τCC	0.021 ± 0.004	0.088 ± 0.009	0.070 ± 0.022	0.115 ± 0.024	0.595 ± 0.138	**0.969 ± 0.003**	**0.000 ± 0.000**
Fair-τCC$_{weak}$	0.096 ± 0.022	0.099 ± 0.015	0.080 ± 0.026	**0.542 ± 0.278**	**0.717 ± 0.191**	0.919 ± 0.018	0.003 ± 0.001
LBM	0.004 ± 0.000	0.004 ± 0.000	0.025 ± 0.003	-	-	0.535 ± 0.152	0.392 ± 0.168
Parity LBM	0.004 ± 0.000	0.004 ± 0.000	0.026 ± 0.004	0.256 ± 0.022	0.510 ± 0.035	0.600 ± 0.087	0.169 ± 0.026
ML (age)							
Fast-τCC	**0.109 ± 0.011**	**0.104 ± 0.013**	0.090 ± 0.024	-	-	0.787 ± 0.046	0.018 ± 0.007
Fair-τCC	0.016 ± 0.001	0.070 ± 0.006	0.088 ± 0.024	0.096 ± 0.009	0.444 ± 0.029	**0.954 ± 0.000**	**0.000 ± 0.000**
Fair-τCC$_{weak}$	0.063 ± 0.036	0.096 ± 0.016	**0.108 ± 0.028**	**0.329 ± 0.204**	**0.643 ± 0.233**	0.806 ± 0.108	0.033 ± 0.043
LBM	0.004 ± 0.000	0.004 ± 0.000	-0.005 ± 0.001	-	-	0.288 ± 0.115	0.718 ± 0.143
Parity LBM	0.001 ± 0.001	0.001 ± 0.001	-0.013 ± 0.006	0.042 ± 0.009	0.195 ± 0.052	0.000 ± 0.000	$+\infty$
Amazon							
Fast-τCC	**0.076 ± 0.013**	**0.079 ± 0.012**	**0.111 ± 0.029**	-	-	0.385 ± 0.025	0.505 ± 0.046
Fair-τCC	0.027 ± 0.003	0.035 ± 0.001	0.011 ± 0.002	0.007 ± 0.002	0.022 ± 0.010	**0.957 ± 0.016**	**0.001 ± 0.001**
Fair-τCC$_{weak}$	0.032 ± 0.004	0.037 ± 0.003	0.012 ± 0.006	0.020 ± 0.012	0.029 ± 0.022	0.773 ± 0.094	0.037 ± 0.020
LBM	0.002 ± 0.000	0.002 ± 0.000	0.102 ± 0.002	-	-	0.000 ± 0.000	$+\infty$
Parity LBM	0.002 ± 0.000	0.002 ± 0.000	0.098 ± 0.004	**0.350 ± 0.031**	**0.829 ± 0.027**	0.023 ± 0.072	$+\infty$
Yelp							
Fast-τCC	**0.566 ± 0.005**	**0.564 ± 0.005**	0.000 ± 0.005	-	-	0.721 ± 0.045	0.142 ± 0.063
Fair-τCC	0.453 ± 0.035	0.499 ± 0.017	**0.001 ± 0.002**	0.025 ± 0.004	0.002 ± 0.004	**0.976 ± 0.007**	**0.000 ± 0.000**
Fair-τCC$_{weak}$	0.536 ± 0.041	0.543 ± 0.028	0.001 ± 0.003	0.030 ± 0.008	0.002 ± 0.002	0.870 ± 0.023	0.018 ± 0.013
LBM	0.042 ± 0.009	0.026 ± 0.005	-0.010 ± 0.005	-	-	0.553 ± 0.056	0.217 ± 0.041
Parity LBM	0.028 ± 0.015	0.017 ± 0.008	-0.011 ± 0.007	**0.225 ± 0.050**	**0.122 ± 0.046**	0.622 ± 0.152	0.161 ± 0.111
LFW							
Fast-τCC	**0.005 ± 0.000**	**0.005 ± 0.000**	0.000 ± 0.000	-	-	0.940 ± 0.013	0.001 ± 0.000
Fair-τCC	0.001 ± 0.000	0.006 ± 0.002	0.001 ± 0.000	0.159 ± 0.029	0.774 ± 0.241	**0.989 ± 0.003**	**0.000 ± 0.000**
Fair-τCC$_{weak}$	0.004 ± 0.002	0.005 ± 0.001	0.000 ± 0.001	**0.595 ± 0.401**	**0.782 ± 0.281**	0.864 ± 0.170	0.012 ± 0.016
LBM	0.002 ± 0.000	0.002 ± 0.000	**0.002 ± 0.000**	-	-	0.393 ± 0.028	0.381 ± 0.032
Parity LBM	0.002 ± 0.000	0.002 ± 0.000	0.001 ± 0.000	0.348 ± 0.031	0.768 ± 0.049	0.803 ± 0.019	0.025 ± 0.005

training, we applied the KMeans algorithm to the row and column probability matrices generated by the model to obtain the definitive cluster assignments. All experiments were conducted on a Linux server equipped with 32 Intel Xeon Skylakes cores running at 2.1 GHz, 256 GB RAM, and one Tesla T4 GPU. The source code of our algorithm and the datasets necessary to reproduce all experiments are available online[3].

5.2 Results

In Table 3, we report the performance of Fair-τCC in comparison with its vanilla version (Fast-τCC), the direct competitor (Parity LBM) and its non-fair counterpart (LBM). The running times are reported in Table 4. We present two versions of our algorithm: the first with a maximum fairness constraint (Fair-τCC), and the second with a more relaxed fairness constraint allowing a small violation for only one protected group (Fair-τCC$_{\text{weak}}$). For MovieLens (ML) with age as sensitive attribute, we allow a minor infringement on the constraint for two protected groups. The α values of the relaxed version are selected from two values, 0.9 and 1.0, by maximizing the row clustering quality $\tau_{R|C}$.

On the MovieLens-1M (ML) dataset with gender as the sensitive attribute, Fair-τCC achieves significant improvements in fairness with a Balance of 0.97

[3] https://github.com/federicopeiretti/fair_taucc

Table 4: Execution times in seconds

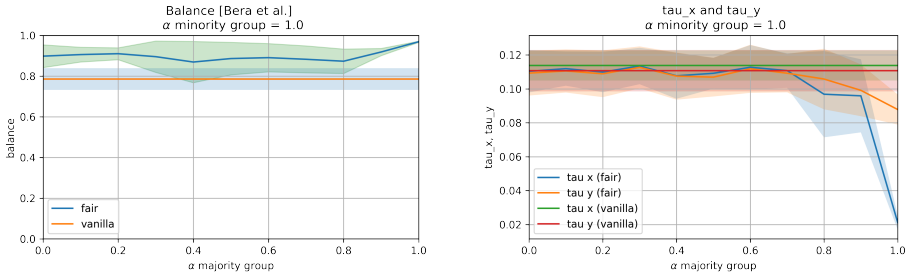
Dataset	Fast-τCC	Fair-τCC	Fair-τCC$_{weak}$	LBM	Parity LBM
ML (gender)	3.37 ± 1.10	490.04 ± 96.41	217.85 ± 158.04	5110.05 ± 248.72	5262.01 ± 257.77
ML (age)	3.37 ± 1.10	202.41 ± 2.15	85.16 ± 47.25	8983.49 ± 121.10	9629.89 ± 604.88
Yelp	0.052 ± 0.0125	35.31 ± 60.94	24.87 ± 61.45	3242.94 ± 508.38	3321.18 ± 558.38
Amazon	1.14 ± 0.49	117.83 ± 22.76	68.85 ± 43.66	12189.15 ± 42.35	13966.79 ± 741.62
LFW	3.51 ± 0.84	245.72 ± 155.32	404.47 ± 251.65	38414.79 ± 5950.14	40201.68 ± 9781.72

and a near-zero KL fairness error (0.0002), significantly outperforming both Fast-τCC (Balance 0.79, KL 0.018) and Parity LBM (Balance 0.60, KL 0.17). However, this comes at the cost of clustering quality, as $\tau_{R|C}$ and $\tau_{C|R}$ drop to 0.021 and 0.088, respectively, compared to Fast-τCC's 0.11 for both metrics. Fair-τCC$_{\text{weak}}$ strikes the trade-off between fairness and clustering quality by allowing a small fairness violation for the majority group ($\alpha_{male} = 0.9$). It achieves higher $\tau_{R|C}$ (0.096) and $\tau_{C|R}$ (0.099) than Fair-τCC while maintaining strong fairness metrics (Balance 0.92, KL 0.003). Interestingly, Fair-τCC$_{\text{weak}}$ also exhibits better alignment with its vanilla counterpart, as shown by ARI$_{\text{rows}}$ (0.54) and ARI$_{\text{cols}}$ (0.72), compared to Fair-τCC's lower values.

For the MovieLens-1M (ML) dataset with age as the sensitive attribute, Fair-τCC again demonstrates superior fairness metrics (Balance 0.954, KL 0.0002), outperforming all other algorithms. However, its ARI score (0.08) is slightly lower than that of its standard version (0.09). Fair-τCC$_{\text{weak}}$ improves on all ARI scores (ARI 0.108, ARI$_{\text{rows}}$ 0.329, ARI$_{\text{cols}}$ 0.643), while maintaining reasonable $\tau_{R|C}, \tau_{C|R}$ and fairness metrics.

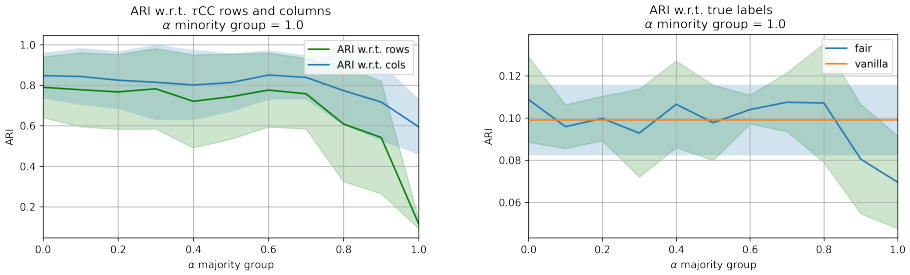
On the Amazon dataset, Fair-τCC achieves near-perfect fairness with a Balance of 0.96 and a KL error of 0.001, addressing the infinite KL errors observed in the other counterparts (Fast-τCC, Parity LBM and standard LBM) due to their lack of fairness constraints. However, its ARI score drops significantly to 0.01 from Fast-τCC's 0.11, reflecting the difficulty of maintaining clustering quality under strict fairness constraints in this dataset. Fair-τCC$_{\text{weak}}$, allowing a small fairness violation for the majority group ($\alpha_{female} = 0.9$), achieves performance very similar to that of its more rigorous version, but the Balance drops significantly to 0.77. Parity LBM achieves high column alignment with its vanilla counterpart (ARI$_{\text{cols}}$=0.83), but its Balance score remains low at 0.02.

The Yelp dataset reveals an interesting trade-off between clustering quality and fairness across algorithms. While Fast-τCC achieves the highest $\tau_{R|C}$ and $\tau_{C|R}$ values of 0.56, it exhibits poor fairness metrics, with a Balance of 0.72 and KL error of 0.14. In contrast, Fair-τCC achieves near-perfect fairness with a Balance of 0.98 and a KL error of 0.0004, while maintaining reasonable clustering quality ($\tau_{R|C} = 0.45$, $\tau_{C|R} = 0.50$). This outcome may be attributable to excessive sparsity in the data matrix. Fair-τCC$_{\text{weak}}$ offers a compromise, exhibiting enhanced $\tau_{R|C}$ and $\tau_{C|R}$ values (0.54) compared to Fair-τCC, while preserving good fairness metrics (Balance 0.87). Parity LBM demonstrates moderate alignment with its baseline counterpart in terms of row and column assignments



(a) Balance

(b) τ_x and τ_y coefficients



(c) ARI_{rows}

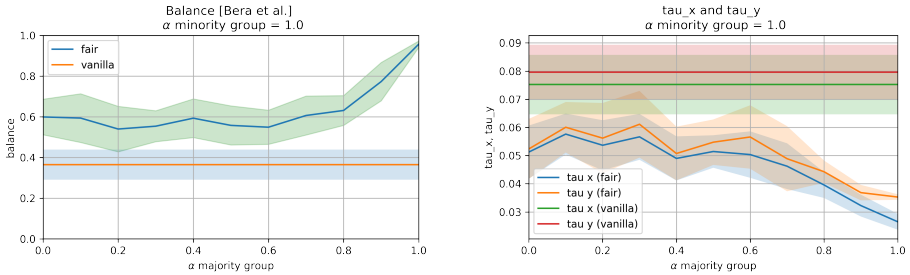
(d) ARI_{cols}

Fig. 2: Fair-τCC vs. Fast-τCC on MovieLens for increasing values of α.

(ARI_{rows}=0.22, ARI_{rows}=0.12), but its overall performance is weaker in both clustering quality and fairness.

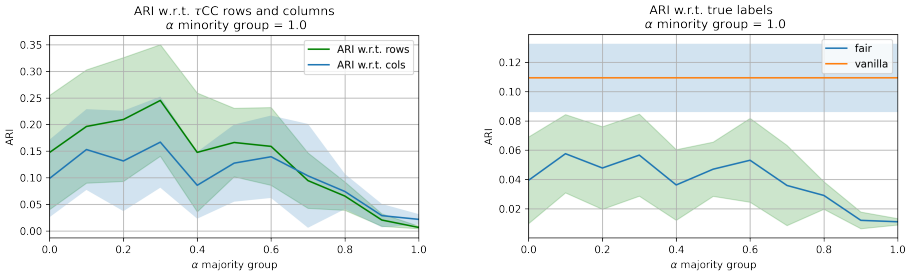
A particularly noteworthy case arises in the Labeled Faces in the Wild (LFW) dataset, where Fair-$\tau\text{CC}_{\text{weak}}$ exhibits a Balance score of 0.86 — lower than that of Fair-τCC (0.94) — despite achieving significantly better row assignment alignment with an ARI_{rows} score of 0.59 compared to Fair-τCC's much lower score of 0.16. This observation underscores an essential aspect of our findings: while strict adherence to fairness constraints may lead to diminished performance in terms of balance, allowing for slight violations can enhance clustering effectiveness without severely compromising overall fairness.

Overall, these results demonstrate that Fair-τCC consistently delivers superior fairness performance across all datasets while maintaining reasonable clustering quality with respect to Fast-τCC, and the other competitors (Parity LBM and standard LBM) and reasonable computational time, as showed in Table 4. Allowing slight violations of fairness constraints, even for a single protected group, can lead to an improvement in terms of clustering quality, while achieving substantial gains in fairness compared to non-fair method. This trade-off makes Fair-$\tau\text{CC}_{\text{weak}}$ particularly suitable for applications where both fairness and clustering effectiveness are critical considerations.



(a) Balance

(b) τ_x and τ_y coefficients



(c) ARI_{rows}

(d) ARI_{cols}

Fig. 3: Fair-τCC compared to Fast-τCC on Amazon for increasing values of α.

5.3 Impact of α on the trade-off between fairness and quality

In this section, the impact of α values on the trade-off between clustering quality and fairness is assessed. To do that, a comparative analysis is conducted between the performance of the proposed algorithm and that of Fast-τCC. This is achieved by systematically varying the $\alpha_{majority}$ value assigned to the majority group in the range of $[0.0, 1.0]$ while maintaining a constant value $\alpha_{minority} = 1.0$ for the group least represented in the dataset. Fig. 2 and Fig. 3 show the trend of evaluation metrics of both algorithms as $\alpha_{majority}$ increases on MovieLens and Amazon datasets, considering gender as sensitive feature. Fair-τCC maintains a relatively high balance with a positive trend as the majority group $\alpha_{majority}$ increases, indicating its superiority in maintaining a well-balanced representations than the non-fair approach. This phenomenon, which persists even with $\alpha_{majority} = 0.0$, is likely attributable to the fact that, prior to implementing a fair reassignment, the row clustering obtained via $\tau_{R|C}$ maximization is evaluated. This observation is more evident in the MovieLens dataset, when $\alpha_{majority}$ is set to a low value. In this case, the $\tau_{R|C}$, $\tau_{C|R}$ and ARI scores are close to those of the non-fair version (Fig. 2b and 2d) and the row and column clustering agreements with it are very high (Fig. 2c). A good trade-off between fairness and clustering quality can be seen for $\alpha_{majority} = 0.9$, as the balance remains high and τ_x, τ_y do not decrease too much.

6 Conclusion

We have introduced an algorithm that computes co-clustering with fairness constraints. It seeks a tradeoff between cluster quality and balance by adopting an optimization strategy that accounts for the protected groups the data instances belong to, by exploiting the properties of a co-clustering approach based on an associative statistical measure that has some desirable properties: it leads to fast convergence and to the identification of a congruent number of clusters on both rows and columns starting from an initial overestimation. The experiments have shown that our algorithm is effective also when compared with the only existing competitor, a co-clustering approach for fair recommendation based on the latent block model.

As future work, we intend to extend the current framework to guarantee fairness balance not only in the row clustering but also in the column clustering, thus addressing fairness constraints bidirectionally. Moreover, we plan to investigate the co-clustering problem under the individual fairness setting. Finally, we will explore multiobjective optimization as a way to automatically select optimal quality-fairness tradeoffs.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ahmadian, S., Epasto, A., Knittel, M., Kumar, R., Mahdian, M., Moseley, B., Pham, P., Vassilvitskii, S., Wang, Y.: Fair hierarchical clustering. In: Proc. NeurIPS 2020 (2020)
2. Ahmadian, S., Epasto, A., Kumar, R., Mahdian, M.: Clustering without over-representation. In: Proc. SIGKDD KDD 2019. pp. 267–275 (2019)
3. Ahmadian, S., Epasto, A., Kumar, R., Mahdian, M.: Fair correlation clustering. In: Proc. AISTATS 2020. vol. 108, pp. 4195–4205 (2020)
4. Backurs, A., Indyk, P., Onak, K., Schieber, B., Vakilian, A., Wagner, T.: Scalable fair clustering. In: Proc. ICML 2019. vol. 97, pp. 405–413 (2019)
5. Battaglia, E., Peiretti, F., Pensa, R.G.: Fast parameterless prototype-based co-clustering. Mach. Learn. **113**(4), 2153–2181 (2024)
6. Battaglia, E., Peiretti, F., Pensa, R.G.: Co-clustering: A survey of the main methods, recent trends, and open problems. ACM Comput. Surv. **57**(2), 48:1–48:33 (2025)
7. Bera, S.K., Chakrabarty, D., Flores, N., Negahbani, M.: Fair algorithms for clustering. In: Proc. NeurIPS 2019. pp. 4955–4966 (2019)
8. 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 2019. vol. 145, pp. 18:1–18:22 (2019)
9. Brubach, B., Chakrabarti, D., Dickerson, J., Khuller, S., Srinivasan, A., Tsepenekas, L.: A pairwise fair and community-preserving approach to k-center clustering. In: Proc. ICML 2020. pp. 1178–1189 (2020)

10. Chakrabarti, D., Dickerson, J.P., Esmaeili, S.A., Srinivasan, A., Tsepenekas, L.: A new notion of individually fair clustering: α-equitable k-center. In: Proc. AISTATS 2022. vol. 151, pp. 6387–6408 (2022)
11. Chen, X., Fain, B., Lyu, L., Munagala, K.: Proportionally fair clustering. In: Proc. ICML 2019. vol. 97, pp. 1032–1041 (2019)
12. Chen, Y., Lei, Z., Rao, Y., Xie, H., Wang, F.L., Yin, J., Li, Q.: Parallel non-negative matrix tri-factorization for text data co-clustering. IEEE Trans. Knowl. Data Eng. **35**(5), 5132–5146 (2023)
13. Chhabra, A., Masalkovaite, 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 2017. pp. 5029–5037 (2017)
15. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.S.: Fairness through awareness. In: Proc. ITCS 2012. pp. 214–226 (2012)
16. Esmaeili, S.A., Brubach, B., Srinivasan, A., Dickerson, J.: Fair clustering under a bounded cost. In: Proc. NeurIPS 2021. pp. 14345–14357 (2021)
17. Fan-Osuala, O.: Gender-based differences in online reviews: An empirical investigation. In: Proc. ICIS 2019 (2019)
18. Fan-Osuala, O.: Gender Bias In Online Reviews (2020), Dataset, available online
19. Frisch, G., Léger, J., Grandvalet, Y.: Co-clustering for fair recommendation. In: Proc. of BIAS 2021, co-located with ECML PKDD 2021 (2021)
20. Ghodsi, S., Seyedi, S.A., Ntoutsi, E.: Towards cohesion-fairness harmony: Contrastive regularization in individual fair graph clustering. In: Proc. PAKDD 2024. pp. 284–296. Springer (2024)
21. Goodman, L.A., Kruskal, W.H.: Measures of association for cross classification. Journal of the American Statistical Association **49**, 732–764 (1954)
22. Gupta, S., Ghalme, G., Krishnan, N.C., Jain, S.: Efficient algorithms for fair clustering with a new notion of fairness. Data Min. Knowl. Discov. **37**(5), 1959–1997 (2023)
23. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. **5**(4), 19:1–19:19 (2016)
24. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In: Workshop on faces in'Real-Life'Images: detection, alignment, and recognition (2008)
25. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)
26. Keuper, M., Tang, S., Andres, B., Brox, T., Schiele, B.: Motion segmentation & multiple object tracking by correlation co-clustering. IEEE Trans. Pattern Anal. Mach. Intell. **42**(1), 140–153 (2020)
27. Kleindessner, M., Samadi, S., Awasthi, P., Morgenstern, J.: Guarantees for spectral clustering with fairness constraints. In: Proc. ICML 2019. pp. 3458–3467 (2019)
28. Li, P., Zhao, H., Liu, H.: Deep fair clustering for visual learning. In: Proc. CVPR 2020. pp. 9067–9076 (2020)
29. Zeng, P., Lin, Z.: couple coc+: An information-theoretic co-clustering-based transfer learning framework for the integrative analysis of single-cell genomic data. PLoS Computational Biology **17**(6), e1009064 (2021)
30. Zhao, Y., Wang, Y., Liu, Y., Cheng, X., Aggarwal, C.C., Derr, T.: Fairness and diversity in recommender systems: A survey. ACM Trans. Intell. Syst. Technol. **16**(1), 2:1–2:28 (2025)
31. Ziko, I.M., Yuan, J., Granger, E., Ayed, I.B.: Variational fair clustering. In: Proc. AAAI 2021. pp. 11202–11209 (2021)