An Associative Approach to Fair Co-clustering

(Discussion Paper)

Federico Peiretti[1], Ruggero G. Pensa[1,*]

[1]University of Turin, Italy

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. Despite the extensive research on fairness considerations in clustering, this issue has not been addressed in the context of co-clustering algorithms. This paper proposes a novel fair co-clustering algorithm 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.

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 [1]. Some examples of fair methods for clustering include the balanced representation [2, 3, 4], the proportionally fair clustering [5] and the equitable distance fairness [6]. 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, 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 [7]. 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 is 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 [8], 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 [9], image segmentation [10], transfer learning [11], object detection and scene

SEBD 2025: 33rd Symposium on Advanced Database System, June 16-19, 2025 - Ischia, Italy

*Corresponding author.

federico.peiretti@unito.it (F. Peiretti); ruggero.pensa@unito.it (R. G. Pensa)

0000-0001-7648-162X (F. Peiretti); 0000-0001-5145-3438 (R. G. Pensa)

categorization [10]. 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 [12]. 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*[1]. 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. 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. 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.

2.1. 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) [2]. 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 [3]. 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., $r_g = |X_g|/|X|$ and $r_g(C_j) = |X_g(C_j)|/|X(C_j)|$.

In this paper, we use the definition given by Gupta *et al.* [13]. They introduce the notion of τ-ratio fairness, which ensures that each cluster contains a predefined fraction of points for each protected attribute value.

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, i.e., $|X_g(C_j)| \geq \tau_g n_g$ with $\tau_g \in [0, 1/k]$.*

We denote the number of clusters with k. Specifically, when τ is set to $1/k$, the definition is equivalent to Definition 1.

[1]The majority of co-clustering algorithms require the final number of row and column clusters to be found as an input. In contrast, our algorithm does not require such prior knowledge. Instead, it relies on the initial number of row and column clusters, which are merely a starting point. The algorithm is designed to automatically reduce the number of row and column clusters during execution, where possible, adapting to the data and determining the final number autonomously.

2.2. Fast Co-clustering

Fast-τCC [14] 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 [15], and can be defined as follows:

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

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 [14] 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. 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. 2). 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 column 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 [2, 3], 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:*

$$balance\left(\{\mathcal{R}, \mathcal{C}\}\right) = min\left(balance(\mathcal{R}), balance(\mathcal{C})\right) \tag{3}$$

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.

3.1. 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.* [14]. 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.

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

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 and l, 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

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 $\mathbf{R} = r_{ik}$ 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,

$$\mathbf{Q} = \mathbf{R}^\top \mathbf{P} \mathbf{C} \tag{4}$$

Equation 2 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

$$\hat{\tau}_{R|C}(\mathcal{R}^{(t)}, \mathcal{C}) = \sum_{i=1}^{n} \left(\sum_{l=1}^{L} \frac{p_{il}}{p_{\cdot l}} q_{kl}^{(t)} - p_{i\cdot} q_{k\cdot}^{(t)} \right) \tag{5}$$

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:

$$\sigma\left(\mathbf{p}_i, \mathbf{q}_k^{(t)}\right) = \sum_{l=1}^{L} \frac{p_{il}}{p_{\cdot l}} q_{kl}^{(t)} - p_{i\cdot} q_{k\cdot}^{(t)} \tag{6}$$

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

$$\hat{\tau}_{R|C}(\mathcal{R}^{(t)}, \mathcal{C}) = \sum_{i=1}^{n} \sigma\left(\mathbf{p}_i, \mathbf{q}_{k^\star}^{(t)}\right) \tag{7}$$

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

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. 9;

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

  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

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:

$$\boldsymbol{\Sigma} = \mathbf{P}\mathbf{C}(\mathbf{Q}^{(t-1)} \odot \mathbf{U}^{(t-1)} - \mathbf{V}^{(t-1)})^\top \tag{8}$$

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

$$\mathbf{U}^{(t)} = \begin{bmatrix} \frac{1}{\sum_l q_{1l}^{(t)}} & \cdots & \frac{1}{\sum_l q_{Kl}^{(t)}} \\ \vdots & \ddots & \vdots \\ \frac{1}{\sum_l q_{1l}^{(t)}} & \cdots & \frac{1}{\sum_l q_{Kl}^{(t)}} \end{bmatrix}, \qquad \mathbf{V}^{(t)} = \begin{bmatrix} \frac{1}{\sum_l q_{1l}^{(t)}} & \cdots & \frac{1}{\sum_l q_{1l}^{(t)}} \\ \vdots & \ddots & \vdots \\ \frac{1}{\sum_l q_{Kl}^{(t)}} & \cdots & \frac{1}{\sum_l q_{Kl}^{(t)}} \end{bmatrix} \tag{9}$$

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 and L.

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 $\boldsymbol{\Sigma}$, we derive a $n \times K$ matrix $\mathbf{D} = (d_{ik})$, defined as follows:

$$d_{ik} = \sigma(\mathbf{p}_i, \mathbf{q}_k^*) - \sigma(\mathbf{p}_i, \mathbf{q}_k) \quad \forall k = 1, \ldots, K \tag{10}$$

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

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

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.10;
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 ($balance \approx 1.0$), 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.

4. Experiments

In this section, we present the findings from our experiments conducted on four real-world datasets to evaluate the effectiveness of Fair-τCC. In our experiments, we use four high-dimensional datasets: two ratings dataset (MovieLens-1M [16] and Yelp [17, 18]), a product reviews dataset (Amazon reviews [17, 18]), and an image collection for facial recognition (Labeled Faces in the Wild [19]). Table 1 summarizes the characteristics of the data matrix for each dataset utilized in our experiments. We compared our algorithm against the standard version of Fast-τCC, which does not incorporate fairness constraints and the only closely related competitor, Parity LBM [12], a latent block model designed for fair recommendations independent of protected attributes. To assess the performance of each algorithm regarding co-clustering quality and fairness, we employed several evaluation metrics:

- $\tau_{R|C}$ and $\tau_{C|R}$: the Goodman-Kruskal's τs measuring the quality of row and column clustering predicted by both versions of τCC algorithm.
- **ARI**: 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 2). Additionally, it is used to compute the aggrement between the clustering and the given ground-truth labels detailed in Table 1 (ARI in Table 2).
- **Balance**: This metric quantifies the balanced representation of protected groups within each cluster according to Definition 1.
- **Kullback-Leibler fairness error**: Based on Kullback-Leibler divergence as proposed in [20], it quantifies the fairness error in clustering. Lower KL error values indicate better adherence to fairness constraints.

To evaluate the effectiveness of our algorithm, we set the number of initial clusters for both rows and columns to $k = 10$ and $l = 10$, 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, and a learning rate of 2e-2.

4.1. Results

In Table 2, 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). 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-$\tau\text{CC}_{\text{weak}}$). For MovieLens (ML) dataset with age as sensitive attribute, having three protected groups, the identification of a fair row clustering is more challenging, as it engenders greater problem complexity and necessitates the allocation of additional computational resources for its resolution. Consequently, in such instance, 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 $\tau_{R|C}$.

Overall, the results demonstrate that Fair-τCC consistently delivers superior fairness performance across all datasets while maintaining reasonable clustering quality relative to its vanilla counterpart, Fast-τCC, and the other competitors (Parity LBM and standard LBM). Allowing slight violations of fairness constraints, even for a single protected group, can lead to and 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.

Table 2
Summary of the results for all co-clustering algorithms.

Algorithm	$\tau_{\mathrm{R}\|\mathrm{C}}$	$\tau_{\mathrm{C}\|\mathrm{R}}$	ARI	$\mathrm{ARI}_{\mathrm{rows}}$	$\mathrm{ARI}_{\mathrm{cols}}$	Balance	KL fairness
ML (gender)							
Fast-τCC	**0.11 ± 0.01**	**0.11 ± 0.01**	**0.10** ± 0.02	-	-	0.79 ± 0.05	0.02 ± 0.01
Fair-τCC	0.02 ± 4e-3	0.09 ± 0.01	0.07 ± 0.02	0.12 ± 0.02	0.59 ± 0.14	**0.97 ± 3e-3**	**2e-4 ± 4e-5**
Fair-$\tau\mathrm{CC}_{\mathrm{weak}}$	0.09 ± 0.02	0.09 ± 0.02	0.08 ± 0.03	**0.54 ± 0.28**	**0.72 ± 0.19**	0.92 ± 0.02	3e-3 ± 10e-4
LBM	-	-	0.03 ± 3e-3	-	-	0.54 ± 0.15	0.39 ± 0.17
Parity LBM	-	-	0.03 ± 4e-3	0.26 ± 0.02	0.51 ± 0.04	0.60 ± 0.09	0.17 ± 0.03
ML (age)							
Fast-τCC	**0.11 ± 0.01**	**0.11 ± 0.01**	0.10 ± 0.02	-	-	0.57 ± 0.04	0.08 ± 0.03
Fair-τCC	0.02 ± 10e-4	0.07 ± 0.01	0.09 ± 0.02	0.09 ± 0.01	0.44 ± 0.03	**0.95 ± 0.00**	**2e-4 ± 0.00**
Fair-$\tau\mathrm{CC}_{\mathrm{weak}}$	0.06 ± 0.04	0.10 ± 0.02	**0.11 ± 0.03**	**0.33 ± 0.20**	**0.64 ± 0.23**	0.80 ± 0.11	0.03 ± 0.04
LBM	-	-	-5e-3 ± 10e-4	-	-	0.29 ± 0.11	0.72 ± 0.14
Parity LBM	-	-	-0.01 ± 6e-3	0.04 ± 0.01	0.19 ± 0.05	0.00 ± 0.00	$+\infty$
Amazon							
Fast-τCC	**0.08 ± 0.01**	**0.08 ± 0.01**	**0.11 ± 0.02**	-	-	0.37 ± 0.07	$+\infty$
Fair-τCC	0.03 ± 2e-3	0.04 ± 10e-4	0.01 ± 2e-3	7e-3 ± 2e-3	0.02 ± 0.01	**0.96 ± 0.02**	**1e-3 ± 6e-4**
Fair-$\tau\mathrm{CC}_{\mathrm{weak}}$	0.03 ± 4e-3	0.04 ± 3e-3	0.01 ± 6e-3	0.02 ± 0.01	0.03 ± 0.02	0.77 ± 0.09	0.04 ± 0.02
LBM	-	-	0.10 ± 2e-3	-	-	0.00	$+\infty$
Parity LBM	-	-	0.10 ± 4e-3	**0.35 ± 0.03**	**0.83 ± 0.03**	0.02 ± 0.07	$+\infty$
Yelp							
Fast-τCC	**0.56 ± 6e-3**	**0.56 ± 6e-3**	**10e-4 ± 3e-3**	-	-	0.66 ± 0.10	0.17 ± 0.09
Fair-τCC	0.45 ± 0.03	0.50 ± 0.02	8e-4 ± 2e-3	0.02 ± 4e-3	2e-3 ± 4e-3	**0.98 ± 7e-3**	**4e-4 ± 2e-4**
Fair-$\tau\mathrm{CC}_{\mathrm{weak}}$	0.54 ± 0.04	0.54 ± 0.03	9e-4 ± 3e-3	0.03 ± 8e-3	2e-3 ± 2e-3	0.87 ± 0.02	0.02 ± 0.01
LBM	-	-	-0.01 ± 5e-3	-	-	0.55 ± 0.06	0.22 ± 0.04
Parity LBM	-	-	-0.01 ± 7e-3	**0.22 ± 0.05**	**0.12 ± 0.05**	0.62 ± 0.15	0.16 ± 0.11
LFW							
Fast-τCC	**5e-3 ± 1e-5**	5e-3 ± 1e-5	3e-5 ± 3e-5	-	-	0.93 ± 0.01	10e-4 ± 5e-4
Fair-τCC	9e-4 ± 1e-4	**6e-3 ± 2e-3**	5e-4 ± 4e-4	0.16 ± 0.03	0.77 ± 0.24	**0.99 ± 3e-3**	**2e-5 ± 0.00**
Fair-$\tau\mathrm{CC}_{\mathrm{weak}}$	4e-3 ± 2e-3	5e-3 ± 9e-3	4e-4 ± 5e-4	**0.59 ± 0.40**	**0.78 ± 0.28**	0.86 ± 0.17	0.01 ± 0.02
LBM	-	-	**2e-3 ± 2e-4**	-	-	0.39 ± 0.03	0.38 ± 0.03
Parity LBM	-	-	10e-4 ± 8e-5	0.35 ± 0.03	0.77 ± 0.05	0.80 ± 0.02	0.03 ± 5e-3

5. 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 accounting for the protected groups 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 fair recommendation approach based on co-clustering. As future work, we will study the co-clustering problem under the individual fairness setting and we will consider multiobjective optimization as a way to automatically select optimal quality-fairness tradeoffs.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] A. Chhabra, K. Masalkovaite, P. Mohapatra, An overview of fairness in clustering, IEEE Access 9 (2021) 130698–130720.

[2] F. Chierichetti, R. Kumar, S. Lattanzi, S. Vassilvitskii, Fair clustering through fairlets, in: Proc. NIPS 2017, 2017, pp. 5029–5037.
[3] S. K. Bera, D. Chakrabarty, N. Flores, M. Negahbani, Fair algorithms for clustering, in: Proc. NeurIPS 2019, 2019, pp. 4955–4966.
[4] I. O. Bercea, M. Groß, S. Khuller, A. Kumar, C. Rösner, D. R. Schmidt, M. Schmidt, On the cost of essentially fair clusterings, in: Proc. APPROX/RANDOM 2019, volume 145, 2019, pp. 18:1–18:22.
[5] X. Chen, B. Fain, L. Lyu, K. Munagala, Proportionally fair clustering, in: Proc. ICML 2019, volume 97, 2019, pp. 1032–1041.
[6] D. Chakrabarti, J. P. Dickerson, S. A. Esmaeili, A. Srinivasan, L. Tsepenekas, A new notion of individually fair clustering: α-equitable k-center, in: Proc. AISTATS 2022, volume 151, 2022, pp. 6387–6408.
[7] E. Battaglia, F. Peiretti, R. G. Pensa, Co-clustering: A survey of the main methods, recent trends, and open problems, ACM Comput. Surv. 57 (2025) 48:1–48:33.
[8] Y. Zhao, Y. Wang, Y. Liu, X. Cheng, C. C. Aggarwal, T. Derr, Fairness and diversity in recommender systems: A survey, ACM Trans. Intell. Syst. Technol. 16 (2025) 2:1–2:28.
[9] Y. Chen, Z. Lei, Y. Rao, H. Xie, F. L. Wang, J. Yin, Q. Li, Parallel non-negative matrix tri-factorization for text data co-clustering, IEEE Trans. Knowl. Data Eng. 35 (2023) 5132–5146.
[10] M. Keuper, S. Tang, B. Andres, T. Brox, B. Schiele, Motion segmentation & multiple object tracking by correlation co-clustering, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2020) 140–153.
[11] P. Zeng, Z. Lin, couple coc+: An information-theoretic co-clustering-based transfer learning framework for the integrative analysis of single-cell genomic data, PLoS Computational Biology 17 (2021) e1009064.
[12] G. Frisch, J. Léger, Y. Grandvalet, Co-clustering for fair recommendation, in: Proc. of BIAS 2021, co-located with ECML PKDD 2021, volume 1524 of *CCIS*, Springer, 2021, pp. 607–630.
[13] S. Gupta, G. Ghalme, N. C. Krishnan, S. Jain, Efficient algorithms for fair clustering with a new notion of fairness, Data Min. Knowl. Discov. 37 (2023) 1959–1997.
[14] E. Battaglia, F. Peiretti, R. G. Pensa, Fast parameterless prototype-based co-clustering, Mach. Learn. 113 (2024) 2153–2181.
[15] L. A. Goodman, W. H. Kruskal, Measures of association for cross classification, Journal of the American Statistical Association 49 (1954) 732–764.
[16] F. M. Harper, J. A. Konstan, The movielens datasets: History and context, ACM Trans. Interact. Intell. Syst. 5 (2016) 19:1–19:19.
[17] O. Fan-Osuala, Gender-based differences in online reviews: An empirical investigation, in: H. Krcmar, J. Fedorowicz, W. F. Boh, J. M. Leimeister, S. Wattal (Eds.), Proc. ICIS 2019, 2019.
[18] O. Fan-Osuala, Gender Bias In Online Reviews, 2020. doi:10.6084/m9.figshare.12834617.v4.
[19] G. B. Huang, M. Mattar, T. Berg, E. Learned-Miller, 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.
[20] I. M. Ziko, J. Yuan, E. Granger, I. B. Ayed, Variational fair clustering, in: Proc. AAAI 2021, 2021, pp. 11202–11209.