

Fair Model-Based Clustering

Jinwon Park¹, Kunwoong Kim², Jihu Lee², Yongdai Kim²

¹Graduate School of Data Science, Seoul National University

²Department of Statistics, Seoul National University
{jwpark127, kwkim.online, superstring1153, ydkim0903}@gmail.com

Abstract

The goal of fair clustering is to find clusters such that the proportion of sensitive attributes (e.g., gender, race, etc.) in each cluster is similar to that of the entire dataset. Various fair clustering algorithms have been proposed that modify standard K-means clustering to satisfy a given fairness constraint. A critical limitation of several existing fair clustering algorithms is that the number of parameters to be learned is proportional to the sample size because the cluster assignment of each datum should be optimized simultaneously with the cluster center, and thus scaling up the algorithms is difficult. In this paper, we propose a new fair clustering algorithm based on a finite mixture model, called Fair Model-based Clustering (FMC). A main advantage of FMC is that the number of learnable parameters is independent of the sample size and thus can be scaled up easily. In particular, mini-batch learning is possible to obtain clusters that are approximately fair. Moreover, FMC can be applied to non-metric data (e.g., categorical data) as long as the likelihood is well-defined. Theoretical and empirical justifications for the superiority of the proposed algorithm are provided.

Introduction

Artificial intelligence (AI) systems have been increasingly deployed as decision support tools in socially sensitive domains such as credit scoring, criminal risk assessment, and college admissions. AI models trained on real-world data may inherently encode biases present in the training data. As a result, these models can exhibit discriminatory behavior or amplify historical biases (Feldman et al. 2015; Barocas and Selbst 2016; Chouldechova 2017; Kleinberg et al. 2018; Mehrabi et al. 2021; Zhou et al. 2021). Numerous studies have shown that, in the presence of such unfairness, these systems tend to favor majority groups, such as white men, while disadvantaging minority groups, such as black women. This can lead to unfair treatment of socially sensitive groups (Dua, Graff et al. 2017; Angwin et al. 2022). These findings underscore the need for fairness-aware learning algorithms to mitigate bias in automated decision-making systems.

Given these circumstances, it is crucial to prioritize fairness in automated decision-making systems to ensure that their development is aligned with the principles of social

responsibility. Consequently, a variety of algorithmic approaches have been introduced to reduce bias by promoting equitable treatment of individuals across demographic groups. For instance, one common approach enforces fairness by requiring that the rates of favorable outcomes be approximately the same across groups defined by sensitive attributes such as race or gender (Calders, Kamiran, and Pechenizkiy 2009; Gitiaux and Rangwala 2021; Zafar et al. 2017; Zeng et al. 2021; Agarwal et al. 2018; Donini et al. 2018; Zemel et al. 2013; Xu et al. 2018; Quadrianto, Sharmanska, and Thomas 2019).

In parallel with advances in supervised learning, the study of algorithmic fairness in unsupervised learning, especially clustering, has garnered increasing attention. Clustering methods have long been used for various machine learning applications, including time series analysis (Lee, Choi, and Son 2024; Paparrizos and Gravano 2016), audio and language modeling (Brown et al. 1992; Diez et al. 2019; Butnaru and Ionescu 2017; Zhang, Wang, and Shang 2023), recommendation systems (Muhammad 2021; Widiyaningtyas, Hidayah, and Adji 2021), and image clustering (Le 2013; Guo et al. 2020; Mittal et al. 2022).

Fair clustering (Chierichetti et al. 2017) aims to ensure that the proportion of each protected group within every cluster is similar to that of the overall population. To this end, a variety of algorithms have been proposed to minimize a given clustering objective (e.g., K -means clustering cost) while satisfying predefined fairness constraints (Bera et al. 2019; Backurs et al. 2019; Li, Zhao, and Liu 2020; Esmaili et al. 2021; Ziko et al. 2021; Zeng et al. 2023).

Most fair clustering algorithms are based on the K -means (center) algorithm which searches for good cluster centers and a good assignment map (mapping each instance to one of the given cluster centers) under fairness constraints. A critical limitation of such existing fair clustering algorithms is that the fairness of a given assignment map depends on the entire training data and thus learning the optimal fair assignment map is computationally demanding. In fact, the assignment of each data point must be learned along with the cluster centers under a fairness constraint, so the number of learnable parameters is proportional to the sample size. Moreover, a mini-batch learning algorithm, a typical tool to scale-up a given learning algorithm, would not be possible since the fairness of a given assignment map cannot be evaluated on a

mini-batch.

Another limitation of existing fair clustering algorithms is that they are based on the K -means algorithm, which requires data to lie in a metric space, thus making it difficult to process non-metric data such as categorical data.

The aim of this study is to develop a new fair clustering algorithm that resolves the aforementioned limitations. The proposed algorithm is based on model-based clustering, and we estimate the parameters under a fairness constraint.

Note that the existing methods are geometric and distance metric-based, yielding deterministic assignments, whereas model-based fair clustering assumes a probabilistic model for the data and uses likelihood-based estimation, providing probabilistic assignments.

An important advantage of the proposed fair clustering algorithm is that the number of learnable parameters is independent of the size of training data and thus can be applied to large-scale datasets easily. Moreover, the development of a mini-batch algorithm to provide approximately fair clusters is possible with theoretical guarantees. In addition, non-metric data can be processed without much difficulty as long as the likelihood can be well-specified.

The main contributions of this paper are summarized as follows.

- We propose a novel fair clustering algorithm based on a probabilistic model that facilitates scalable mini-batch learning and dynamically recalculates fair assignments for large test datasets.
- Unlike most existing fair clustering methods, our proposed algorithms are built upon a probabilistic mixture model and naturally extend to categorical data.
- We establish a theoretical guarantee that the proposed model generalizes well to unseen data, and validate it through comprehensive empirical evaluations.

Review of Existing Fair Clustering Algorithms

There are three categories of existing methods for fair clustering. (1) Pre-processing methods (Chierichetti et al. 2017; Backurs et al. 2019) use the concept of fairlets—small subsets of the data satisfying fairness—and then perform clustering on the space of fairlets. (2) In-processing methods (Kleindessner et al. 2019; Li, Zhao, and Liu 2020; Ziko et al. 2021; Zeng et al. 2023; Kim et al. 2025) optimize both the assignments and cluster centers to find a good clustering among those satisfying a given fairness level. (3) Post-processing methods (Bera et al. 2019; Harb and Lam 2020) only build fair assignments for fixed cluster centers. Here, we briefly review in-processing methods to explain the limitations of existing fair clustering algorithms.

Let $\mathcal{D} = \{x_1, \dots, x_N\}$ be the given training data to be clustered, and let s_1, \dots, s_N be sensitive attributes corresponding to x_1, \dots, x_N , where $s_i \in [M]$ for all $i \in [N]$. Here, M is the number of sensitive groups. Let $\mathcal{D}^{(s)} = \{x_i \in \mathcal{D} : s_i = s\}$ for $s \in [M]$. We first focus on the case $M = 2$ for the following discussion, and propose a way of extending the algorithm in the *Modification of FMC to multinary sensitive attributes* section.

A typical clustering algorithm (without a fairness constraint) receives \mathcal{D} and the number of clusters K as inputs and yields the cluster centers $\mu_k, k \in [K]$, and the assignment map $\mathcal{A} : \mathcal{D} \rightarrow [K]$ such that $\mathcal{A}(x_i)$ is the cluster to which x_i belongs. The algorithm searches $\{\mu_k, k \in [K]\}$ and \mathcal{A} that minimizes the clustering cost $L(\mathcal{D}; K) := \sum_{i=1}^N \rho(x_i, \mu_{\mathcal{A}(x_i)})$ for a given metric ρ .

A fair clustering algorithm receives $\mathcal{D}^{(s)}$, $s \in [2]$ and K as inputs and searches $\mu_k, k \in [K]$ and \mathcal{A} that minimize $L(\mathcal{D}; K)$ subject to the constraint that \mathcal{A} is fair, i.e., $|\{i : \mathcal{A}(x_i) = k, s_i = 1\}|/N_1 \approx |\{i : \mathcal{A}(x_i) = k, s_i = 2\}|/N_2$ for all $k \in [K]$, where $N_s = |\mathcal{D}^{(s)}|$. In certain cases, fair assignment maps do not exist. To ensure existence, we consider a soft assignment map $\mathcal{A}_{soft} : \mathcal{D} \rightarrow \mathcal{S}^K$, where \mathcal{S}^K is the $(K - 1)$ -dimensional simplex (Kim et al. 2025). Here, $\mathcal{A}_{soft}(x_i)_k$ is interpreted as the probability of x_i belonging to cluster k .

A typical procedure for in-processing fair clustering algorithms is summarized as follows. A fair clustering algorithm first finds the cluster centers using a fairness-agnostic clustering algorithm. It then updates the (soft) assignment map while the cluster centers are fixed. In turn, it updates the cluster centers while the assignment map is fixed, and it repeats these two updates iteratively until convergence.

A problem in such fair clustering algorithms is that updating the assignment map is computationally demanding when the sample size is large, as the full optimization has cubic complexity: the number of learnable parameters is N , since $\mathcal{A}(x_i), i \in [N]$ must be updated simultaneously. Moreover, a mini-batch algorithm—a typical way to scale-up a learning procedure—is not possible, since the fairness of a given assignment map depends on the entire dataset.

Another problem arises when assigning test data to clusters. Assigning newly arrived data fairly after learning fair clusters is challenging, because the learned assignment map is defined only on the training data. A practical solution would be to obtain the optimal assignment map for test data while the cluster centers are fixed. This naive solution, however, is not applicable when test data arrive sequentially.

The aim of this paper is to propose a new fair clustering algorithm whose number of parameters is independent of the size of the training data and is thus easy to scale-up. Moreover, it is possible to develop a mini-batch algorithm to provide an approximately fair clustering with theoretical guarantees. In addition, the algorithm yields a parameterized assignment map, and thus assigning newly arrived unseen data is easy.

Finite Mixture Models

One renowned model-based clustering method is the finite mixture model, which assumes that $x_1, \dots, x_N \in \mathbb{R}^d$ are independent realizations of a d -dimensional random variable X with density given by

$$X \sim \mathbf{f}(\cdot; \Theta) := \sum_{k=1}^K \pi_k f(\cdot; \theta_k), \quad (1)$$

where K is the number of mixture components and $f(\cdot; \theta_k)$ is the density of a parametric distribution with parameter θ_k

for $k \in [K]$. The Gaussian distribution is commonly used for $f(\cdot; \theta)$ but other parametric distributions can also be used (e.g., the categorical (multinoulli) distribution for categorical data). The mixture weight vector $\pi = (\pi_1, \dots, \pi_K)$ lies on the $(K - 1)$ -dimensional simplex. From a clustering perspective, K is the number of clusters; $f(\cdot; \theta_k)$ is the density of the k th component (cluster); and π_k is the probability of belonging to the k th cluster. The parameter $\Theta = (\pi, (\theta_1, \dots, \theta_K))$ can be estimated by maximum likelihood (MLE), which maximizes the log-likelihood given by

$$\ell(\Theta | \mathcal{D}) = \sum_{i=1}^N \log \left(\sum_{k=1}^K \pi_k f(x_i; \theta_k) \right).$$

Various algorithms have been proposed to compute the MLE in the finite mixture model. Among these, the Expectation-Maximization (EM) algorithm (Titterton, Smith, and Makov 1985; Xu and Jordan 1996a; Wu 1983a) and gradient-descent-based (GD) optimization algorithm (Gepperth and Pfülb 2021; Hosseini and Sra 2020) are most widely used. Xu and Jordan (1996b) showed that EM and GD for the finite mixture model are linked via a projection matrix. An advantage of EM is that the objective function is monotonically increasing without the need to set a learning rate. In contrast, GD is more flexible and applicable to a broader range of differentiable objectives.

Assignment Map for the Finite Mixture Model

An equivalent formulation of the finite mixture model in Eq. (1) is given by:

$$\begin{aligned} Z_1, \dots, Z_N &\stackrel{\text{i.i.d.}}{\sim} \text{Categorical}(\pi) \\ X_i | Z_i &\sim f(\cdot; \theta_{Z_i}), i \in [N]. \end{aligned}$$

Here, from a clustering perspective, Z_i is interpreted as the cluster index to which x_i belongs. Note that

$$p(Z_i = k | X_i = x_i; \Theta) = \frac{\pi_k f(x_i; \theta_k)}{\sum_{l=1}^K \pi_l f(x_i; \theta_l)}. \quad (2)$$

In this paper, we use (2) as the (soft) assignment map corresponding to the finite mixture model with parameter Θ and denote it by $\psi_k(x_i; \Theta) = p(Z_i = k | X_i = x_i; \Theta)$ for $k \in [K]$.

In practice, the hard assignment map $\psi_{hard}(x_i; \Theta)$ for any given x_i can be determined from the soft assignment map via $\psi_{hard}(x_i; \Theta) = \operatorname{argmax}_{k \in [K]} \{\psi_k(x_i; \Theta)\}$. In the case of the Gaussian mixture model, this hard assignment map is close to the optimal assignment map of the K -center algorithm under regularity conditions. See Theorem 6 in the Appendix for a detailed discussion.

Learning a Fair Finite Mixture Model

Fairness Constraint for the Finite Mixture Model

Let $\mathcal{C} = \{C_1, \dots, C_K\}$ denote a given set of clusters of \mathcal{D} where $C_k = \{x_i \in \mathcal{D} : Z_i = k\}$ is the k th cluster of \mathcal{D} . Let $\mathcal{C}^1 = \{C_1^1, \dots, C_K^1\}$ and $\mathcal{C}^2 = \{C_1^2, \dots, C_K^2\}$ denote the sets of clusters of $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$, respectively, where $C_k = C_k^1 \cup C_k^2$. The most commonly used fairness measure

in clustering is *Balance* (Chierichetti et al. 2017), which is the minimum of the ratios of each sensitive attribute in every cluster. The Balance of $\mathcal{C} = \{C_1, \dots, C_K\}$ is defined as

$$\min_{k \in [K]} \min \{ |C_k^1|/|C_k^2|, |C_k^2|/|C_k^1| \}. \quad (3)$$

Smaller Balance indicates less fair clusters, and vice versa. Bera et al. (2019) studied a clustering algorithm that finds optimal clusters under a Balance constraint by constructing fair assignments.

A similar but numerically easier measure for fairness than Balance is a difference-based rather than ratio-based measure. That is, we define *Gap* as

$$\max_{k \in [K]} \left| \frac{|C_k^1|}{N_1} - \frac{|C_k^2|}{N_2} \right|. \quad (4)$$

A larger Gap means less fairness of clusters and vice versa. Kim et al. (2025) considered the additive version of Gap defined by

$$\sum_{k \in [K]} \left| \frac{|C_k^1|}{N_1} - \frac{|C_k^2|}{N_2} \right|, \quad (5)$$

and developed a fair clustering algorithm.

A natural extension of Gap for the assignment map $\psi_k(\cdot; \Theta)$, $k \in [K]$, is

$$\Delta(\Theta) := \max_{k \in [K]} \left| \frac{\sum_{x_i \in \mathcal{D}^{(1)}} \psi_k(x_i; \Theta)}{N_1} - \frac{\sum_{x_j \in \mathcal{D}^{(2)}} \psi_k(x_j; \Theta)}{N_2} \right|. \quad (6)$$

In this paper, we propose to estimate Θ under the constraint $\Delta(\Theta) \leq \varepsilon$ for a prespecified fairness level $\varepsilon \geq 0$ (the smaller ε , the fairer). That is, the objective of our proposed fair clustering is:

$$\max_{\Theta} \ell(\Theta | \mathcal{D}) \text{ subject to } \Delta(\Theta) \leq \varepsilon. \quad (7)$$

Given a Lagrange multiplier $\lambda \geq 0$, we can alternatively maximize $\ell(\Theta | \mathcal{D}; \lambda) := \ell(\Theta | \mathcal{D}) - \lambda \Delta(\Theta)$. We abbreviate $\Delta(\Theta)$ as Δ where appropriate.

For the algorithms presented below, we identify the maximizing index k in Eq. (6) and then evaluate Δ using that k .

We use Gap instead of Balance for numerical stability when applying gradient-descent optimization. Regarding the relationship between Balance and Gap, we note that: (i) (Kim et al. 2025) theoretically showed that $|\text{Balance} - \min(n_0/n_1, n_1/n_0)| \leq \text{Additive Gap}$ in Eq. (5) (where $\min(n_0/n_1, n_1/n_0)$ is the balance of perfectly fair clustering), and (ii) our numerical experiments show that smaller Δ corresponds to a larger Balance (Fig. 11 in the Appendix).

Learning Algorithms: FMC-GD and FMC-EM

To optimize the objective in Eq. (7), we employ gradient-descent (GD) and expectation-maximization (EM) algorithms, with pseudocode in Algorithms 1 and 2. We hereafter refer to these as FMC-GD (Fair Model-based Clustering with Gradient Descent) and FMC-EM (Fair Model-based Clustering with Expectation-Maximization), respectively.

Algorithm 1 FMC-GD (Gradient-Descent)In practice, we set $(T, \gamma) = (10000, 10^{-3})$ **Input:** Dataset $\mathcal{D} = \{x_1, \dots, x_N\}$, Number of clusters K , Lagrange multiplier λ , Maximum number of iterations T and Learning rate γ .**Initialize:** $\Theta^{[0]} = (\boldsymbol{\eta}^{[0]}, \boldsymbol{\theta}^{[0]})$ **while** Θ has not converged and $t < T$ **do**

$$\boldsymbol{\theta}^{[t+1]} \leftarrow \boldsymbol{\theta}^{[t]} - \gamma \frac{\partial}{\partial \boldsymbol{\theta}} (-\ell(\Theta^{[t]} | X) + \lambda \Delta(\Theta^{[t]}))$$

$$\boldsymbol{\eta}^{[t+1]} \leftarrow \boldsymbol{\eta}^{[t]} - \gamma \frac{\partial}{\partial \boldsymbol{\eta}} (-\ell(\Theta^{[t]} | X) + \lambda \Delta(\Theta^{[t]}))$$

$$\boldsymbol{\pi}^{[t+1]} \leftarrow \text{softmax}(\boldsymbol{\eta}^{[t+1]})$$

$$t \leftarrow t + 1$$

end while

(1) **FMC-GD** FMC-GD maximizes $\ell(\Theta | \mathcal{D}; \lambda)$ using gradient-descent-based optimization. To improve numerical stability of the gradients, we reparameterize the mixture weights as $\boldsymbol{\pi} = \text{softmax}(\boldsymbol{\eta})$. The gradient formulas used in Algorithm 1 are provided in the Appendix, in the section *Derivation of the Gradients in Algorithm 1*.

(2) **FMC-EM** For FMC-EM, we consider the complete-data $Y = (\mathcal{D}, Z)$, where $Z = (Z_1, \dots, Z_N)^\top$, along with the complete-data log-likelihood $\ell_{comp}(\Theta | Y)$:

$$\ell_{comp}(\Theta | Y) = \sum_{i=1}^N \sum_{k=1}^K \mathbb{I}(Z_i = k) \{\log \pi_k + \log f(x_i; \theta_k)\}. \quad (8)$$

As in the standard EM algorithm, FMC-EM iterates the E-step and M-step. Let $\Theta^{[t]}$ be the updated parameter at the t th iteration. Given \mathcal{D} and $\Theta^{[t]}$, the Q function (computed in the E-step and maximized in the M-step) is

$$Q(\Theta | \Theta^{[t]}) = \mathbb{E}_{Z|\mathcal{D}, \Theta^{[t]}} \ell_{comp}(\Theta | Y). \quad (9)$$

See Eq. (42) in the Appendix for a detailed derivation of $Q(\Theta | \Theta^{[t]})$. For FMC-EM, the Q function is modified to

$$Q_{fair}(\Theta | \Theta^{[t]}; \lambda) = \mathbb{E}_{Z|\mathcal{D}, \Theta^{[t]}} (\ell_{comp}(\Theta | Y) - \lambda \Delta(\Theta)). \quad (10)$$

We compute Q_{fair} in the E-step and update Θ to maximize Q_{fair} in the M-step. Since maximizing Q_{fair} admits no closed-form solution, we instead apply gradient-based optimization to ensure an increase at each iteration. This procedure is a special case of the Generalized EM (GEM) algorithm (Dempster, Laird, and Rubin 1977), which requires only that each update of Θ increase the Q function. To sum up, given $\Theta^{[t]}$ at the t th iteration, we update the parameters $\Theta^{[t+1]}$ via gradient-descent-based optimization, choosing the learning rate to satisfy

$$Q_{fair}(\Theta^{[t+1]} | \Theta^{[t]}; \lambda) \geq Q_{fair}(\Theta^{[t]} | \Theta^{[t]}; \lambda). \quad (11)$$

See Fig. 10 in the Appendix for the empirical convergence of FMC-EM.

Mini-Batch Learning with Sub-Sampled Δ

When the size of the dataset is large, reducing computational cost becomes a critical concern, and the use of mini-batches

Algorithm 2 FMC-EM (Expectation-Maximization)In practice, we set $(T, R, \gamma) = (200, 10, 10^{-2})$ **Input:** Dataset $\mathcal{D} = \{x_1, \dots, x_N\}$, Number of clusters K , Lagrange multiplier λ , Maximum numbers of iterations T, R and Learning rate γ .**Initialize:** $\Theta^{[0]} = (\boldsymbol{\eta}^{[0]}, \boldsymbol{\theta}^{[0]})$ **while** Q_{fair} has not converged and $t < T$ **do**Compute $Q_{fair}(\Theta | \Theta^{[t]}; \lambda)$ in Eq. (10)**while** $r < R$ **do**

$$\boldsymbol{\theta}_{(r+1)}^{[t]} \leftarrow \boldsymbol{\theta}_{(r)}^{[t]} - \gamma \frac{\partial}{\partial \boldsymbol{\theta}} (-Q_{fair}(\Theta; \Theta^{[t]}, \lambda))$$

$$\boldsymbol{\eta}_{(r+1)}^{[t]} \leftarrow \boldsymbol{\eta}_{(r)}^{[t]} - \gamma \frac{\partial}{\partial \boldsymbol{\eta}} (-Q_{fair}(\Theta; \Theta^{[t]}, \lambda))$$

$$\boldsymbol{\pi}_{(r+1)}^{[t]} \leftarrow \text{softmax}(\boldsymbol{\eta}_{(r+1)}^{[t]})$$

$$r \leftarrow r + 1$$

end while

$$\boldsymbol{\theta}^{[t+1]} \leftarrow \boldsymbol{\theta}_{(R)}^{[t]}$$

$$\boldsymbol{\pi}^{[t+1]} \leftarrow \boldsymbol{\pi}_{(R)}^{[t]}$$

$$t \leftarrow t + 1$$

end while

can be a compelling approach. The objective function of FMC consists of two terms: the log-likelihood and the fairness penalty term Δ . However, note that a mini-batch algorithm can be applied to the log-likelihood term but cannot be easily applied to Δ . Hence, we need a technique to reduce the computational cost of computing the gradient of Δ .

For this purpose, we use the fact that Δ can be closely approximated by its value on sub-sampled data. Let \mathcal{D}_n be a dataset of size n obtained by randomly sampling n data points from \mathcal{D} , and let $\Delta(\Theta; \mathcal{D}_n)$ be the Δ computed on \mathcal{D}_n . Under regularity conditions, it can be shown that fair clusters under the sub-sampled Δ constraint are approximately fair in terms of the (population) Δ constraint. For simplicity, we focus on the Gaussian mixture model: let $f(x; \theta)$ be a Gaussian distribution with mean vector μ and covariance matrix Σ , and thus $\Theta = (\boldsymbol{\pi}, ((\mu_1, \Sigma_1), \dots, (\mu_K, \Sigma_K)))$. For given positive constants ξ, ζ and ν , let $\Phi_{\xi, \zeta, \nu} = \{\Theta : \xi < \min_k \{\pi_k\} \leq \max_k \{\pi_k\} < 1 - \xi, \max_k \|\mu_k\| \leq \nu, \zeta < \min_k \lambda_{\min}(\Sigma_k) \leq \max_k \lambda_{\max}(\Sigma_k) < 1/\zeta\}$, where λ_{\min} and λ_{\max} are the smallest and largest eigenvalues, respectively. The following proposition gives a bound on the error between $\Delta(\Theta; \mathcal{D}_n)$ and $\Delta(\Theta)$. To avoid technical difficulties, we assume that \mathcal{D}_n is obtained by sampling with replacement from \mathcal{D} . The proof is given in the Appendix.

Proposition 1. Let $n_s = |\{x_i \in \mathcal{D}_n : s_i = s\}|$ for $s \in \{1, 2\}$. Then, with probability at least $1 - \delta$, we have

$$\sup_{\Theta \in \Phi_{\xi, \zeta, \nu}} |\Delta(\Theta) - \Delta(\Theta; \mathcal{D}_n)| \leq 4C \sqrt{\frac{d}{n'}} + 8 \sqrt{\frac{2 \log(8/\delta)}{n'}} \quad (12)$$

for some constant $C = C(\xi, \zeta, \nu, x_{\max})$, where $x_{\max} = \max_{x \in \mathcal{D}} \|x\|_2$ and $n' = \min\{n_1, n_2\}$.

Motivated by Proposition 1, we propose to estimate Θ by maximizing

$$\ell(\Theta | \mathcal{D}) - \lambda \Delta(\Theta; \mathcal{D}_n),$$

where we apply mini-batch learning to the term $\ell(\Theta \mid \mathcal{D})$ while \mathcal{D}_n is fixed. In FMC-GD, we take gradient steps for $\ell(\Theta \mid \mathcal{D})$ on a given mini-batch, whereas in FMC-EM, we compute $\ell_{comp}(\Theta \mid Y)$ in the E-step only on a given mini-batch.

See Algorithm 3 in the Appendix for details of the mini-batch learning algorithm with the sub-sampled Δ constraint. Moreover, Karimi, Lavielle, and Moulines (2019) studied convergence properties of the mini-batch EM algorithm, and we show a similar result for the mini-batch GEM in Theorem 7 in the Appendix.

We emphasize that neither mini-batch learning nor a sub-sampled fairness constraint is possible in existing fair clustering algorithms, since the assignment map for the entire dataset must be learned simultaneously. The key innovation of the proposed fair finite mixture model is to use the parametric model in Eq. (2) as the (soft) assignment map. Note that learning the parameters is computationally easier than directly learning the assignment map for the entire dataset.

Sub-Sample Learning

Another advantage of FMC is that the functional form of the assignment map is available after training, which allows us to fairly assign newly arrived data (which are not available during the learning of Θ) to clusters. That is, we assign an unseen datum x to cluster k with probability $\psi_k(x; \hat{\Theta})$, where $\hat{\Theta}$ is the estimated parameter vector. However, this fair post-assignment is not possible for existing fair clustering algorithms, since the assignment map is defined only for the training data.

The fair post-assignment also enables learning the fair mixture model on randomly selected sub-samples when the entire dataset is too large. Specifically, we estimate Θ on randomly selected sub-samples \mathcal{D}_n of size n from \mathcal{D} , and then assign the data in $\mathcal{D} \setminus \mathcal{D}_n$ using the estimated assignment map $\{\psi_k(x; \hat{\Theta})\}_{k \in [K]}$. Proposition 1 guarantees that the clusters obtained in this way are approximately fair. In the *Numerical experiments* section, we empirically show that the reduction in performance, as measured by clustering cost and Δ , due to sub-sampling is minimal, unless n is too small.

Numerical Experiments

In our numerical experiments, we evaluate FMC by analyzing several real-world datasets: (i) on three moderate-scale real datasets, we show that FMC is competitive with baseline methods; and (ii) on a large-scale dataset consisting of millions of instances, FMC outperforms the baselines while requiring significantly less computational time. In addition, we investigate mini-batch learning and sub-sample learning of FMC, and show that the two algorithms perform similarly when the sub-sample size is not too small (e.g., 5%).

We additionally conduct parameter impact studies on (i) the number of clusters K and (ii) the choice of covariance structure when \mathbf{f} is a Gaussian mixture.

Experimental Settings

Datasets We use four real-world datasets: Adult, Bank, Credit, and Census datasets, which are all available in the

UCI Machine Learning Repository¹. Brief descriptions of the datasets are given below, with details in the Appendix.

- Adult dataset: UCI Adult (1994 US Census), 32,561 samples; 5 continuous and 8 categorical features; sensitive attribute: gender (male 66.9%, female 33.1%).
- Bank dataset: UCI Bank Marketing, 41,008 samples; 6 continuous and 10 categorical features; sensitive attribute: marital status (married 60.6%, unmarried 39.4%).
- Credit dataset: UCI Default of Credit Card Clients, 30,000 samples; 4 continuous and 8 categorical features; sensitive attribute: education, aggregated into two groups (46.8% vs. 53.2%) or three groups (46.8%, 35.3%, 17.9%).
- Census dataset: UCI Census (1990 US Census), 2,458,285 samples; 25 features; sensitive attribute: gender (male 51.53%, female 48.47%).

We analyze only the continuous variables in the main analysis, since we focus on the Gaussian mixture model. Subsequently, we include categorical data to validate the applicability of FMC to categorical variables. We standardize the continuous variables to have zero mean and unit variance. Then, we apply L_2 normalization to the standardized data (i.e., so that each datum has unit L_2 norm), because VFC (Ziko et al. 2021), one of the baseline methods, often fails to converge without L_2 normalization. Numerical results without L_2 normalization are given in the Appendix.

Algorithms For continuous data, we use the Gaussian mixture model (GMM) for \mathbf{f} with an isotropic covariance matrix $\sigma^2 \mathbb{I}_d$. That is, the complete-data log-likelihood of the GMM is derived as

$$\ell_{comp}(\Theta \mid Y) = \sum_{i=1}^N \sum_{k=1}^K \mathbb{I}(Z_i = k) \left(\log \pi_k - \frac{d \log(2\pi\sigma^2)}{2} - \frac{\|x_i - \mu_k\|_2^2}{2\sigma^2} \right), \quad (13)$$

where $\Theta = (\boldsymbol{\pi}, (\boldsymbol{\mu}, \sigma))$. For categorical data, we use the categorical (multinoulli) mixture model; details are given in Eqs. (84) and (85) in the Appendix.

For fair clustering baseline methods, we consider three existing methods: SFC, VFC, and FCA. SFC (Backurs et al. 2019) is a scalable fair clustering algorithm based on fairlet decomposition, yielding near-perfectly fair clustering. VFC (Ziko et al. 2021) performs fair clustering by adding a variational fairness constraint based on the Kullback–Leibler (KL) divergence. FCA (Kim et al. 2025) is a recently developed state-of-the-art method that performs fair clustering by matching distributions across different groups. In addition, for fairness-unaware clustering methods, we consider K -means and Gaussian mixture models learned with EM or GD (denoted MM-EM and MM-GD).

Evaluation For fairness metrics, we consider Δ and Balance, and for clustering utility metrics, we consider Cost (the sum of distances from each data point to its assigned cluster center). For FMC, we use the hard assignment map when computing the fairness metrics to ensure comparability, since SFC and VFC only yield hard assignments.

¹<https://archive.ics.uci.edu/datasets>

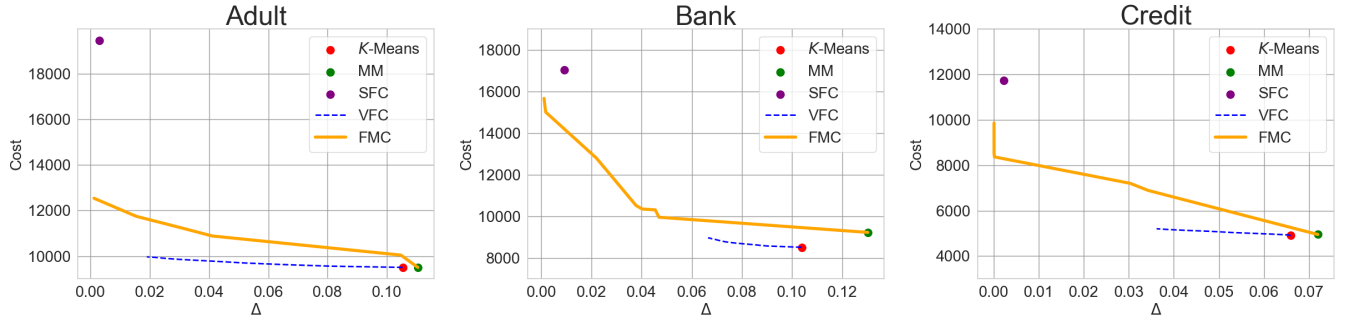


Figure 1: Pareto front lines between Δ and Cost on Adult, Bank, and Credit datasets. See Fig. 5 for the lines between Balance and Cost. See Fig. 6 for the similar results without L_2 normalization.

Initial parameters We set the initial value of mixture weights to $\pi = (1/K, \dots, 1/K)^T$. The initial means μ are set to the cluster centers obtained by the K -means algorithm with random initialization. For the variance, we set $\sigma = 1.0$. See the Appendix for additional implementation details.

Performance Comparison

Comparing FMC-EM and FMC-GD Fig. 2 presents Pareto front plots of Δ versus Cost (with standard deviation bands from five random initializations) for FMC-EM and FMC-GD on the Adult dataset. It shows that FMC-EM achieves a better cost-fairness trade-off with smaller variance. Accordingly, we use FMC-EM for subsequent experiments and, where appropriate, refer to it simply as FMC.

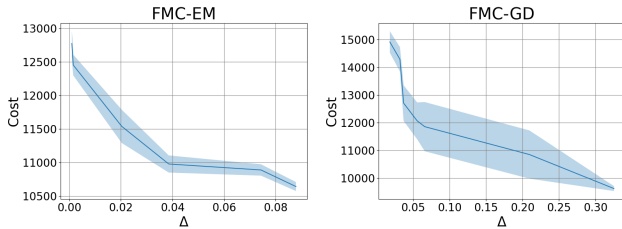


Figure 2: Pareto front lines between Δ and Cost (with standard deviation bands obtained by five random initializations) of FMC-EM (left) and FMC-GD (right) on Adult dataset. See Fig. 4 in Appendix for similar results with respect to Balance and on other datasets.

FMC vs. Baseline algorithms Fig. 1 shows the Pareto front plots of Δ versus Cost. Note that SFC operates only at (near-perfect) fairness and therefore appears as a single point. These results lead to two observations.

(i) FMC is better than SFC in terms of Cost at a near-perfect fairness level. This is not surprising, since FMC learns the cluster centers and the fair assignment map simultaneously, whereas SFC learns them sequentially. An additional advantage of FMC over SFC is the ability to control the fairness level, whereas SFC cannot.

(ii) VFC is slightly superior to FMC with respect to the cost-fairness trade-off. This is expected because VFC min-

imizes Cost explicitly whereas FMC maximizes the log-likelihood, which is only indirectly related to Cost. On the other hand, VFC fails to control the fairness level across the entire range. For example, VFC does not provide a fair clustering with $\Delta < 0.06$ on the Bank dataset. Another advantage of FMC over VFC is robustness to data pre-processing methods. As previously discussed, we apply L_2 normalization, since VFC fails to converge without it on the Credit and Census datasets (see Figs. 6 and 7 in the Appendix). We note that L_2 normalization is not a standard pre-processing step, since it maps points onto the unit sphere. In this sense, FMC is more broadly applicable. See Table 3 in the Appendix for additional comparisons between FMC and FCA.

Analysis on Large-Scale Dataset

This section provides experimental results on the Census dataset, a large-scale dataset consisting of more than 2 million samples.

FMC vs. Baseline algorithms For FMC, we apply mini-batch learning with a sub-sampled Δ , where the mini-batch and sub-sample sizes are set to 10% of the full dataset. Fig. 3 displays the results, showing that FMC is competitive with VFC and better than SFC. In addition, consistent with the results on moderate-scale data, FMC retains the advantage of controlling the fairness level across the entire range. Moreover, FMC with mini-batch learning reduces computation time significantly (e.g., about 20x faster than VFC), as shown in Table 1.

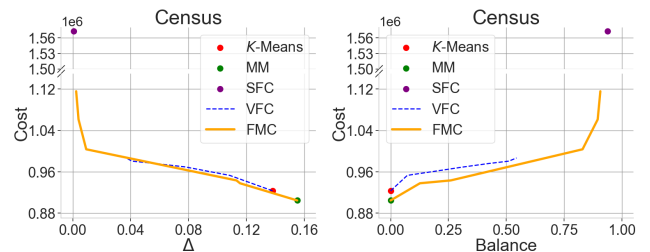


Figure 3: Pareto front lines between $\{\Delta, \text{Balance}\}$ and Cost on Census dataset. See Fig. 7 in Appendix for the similar results without L_2 normalization.

Algorithm	Time (std)
VFC	5053.0 (915.5)
SFC	2218.2 (535.1)
FMC (mini-batch, $n = 0.1N$)	253.0 (12.1)

Table 1: Comparison of the average computation time (seconds) with standard deviations (std) over five random initials on Census dataset ($N = 2,458,285$).

Mini-batch learning vs. Sub-sample learning We further investigate whether sub-sample learning empirically performs similarly to mini-batch learning by varying the sub-sample sizes in $\{1\%, 3\%, 5\%, 10\%\}$. For example, when training with 5% sub-samples, we evaluate the learned model on the full dataset (i.e., perform inference on the remaining 95% of the data and then aggregate the inferred assignments with those of the training data). Table 2 compares the performance of (i) sub-sample learning (sizes $\{1\%, 3\%, 5\%, 10\%\}$) and (ii) mini-batch learning with batch size 10%. Sub-sample learning with size $\geq 5\%$ yields performance similar to mini-batch learning but with lower computational cost. These results suggest that sub-sample learning is useful when the dataset is very large.

Method ($\frac{n}{N}$)	Cost ($\times 10^5$)	Δ	Balance	Time
SS (1%)	11.167	0.005	0.851	53.7%
SS (3%)	10.988	0.004	0.894	65.8%
SS (5%)	10.973	0.003	0.883	71.4%
SS (10%)	10.916	0.003	0.881	77.2%
MB (10%)	10.857	0.002	0.896	100.0%

Table 2: Performances of mini-batch learning (MB) and sub-sample learning (SS) on Census dataset. The computation times of SS are measured by the relative ratio compared to the mini-batch learning with 10%, and the Lagrangian λ for each case is tuned to achieve the lowest Gap value i.e., $\Delta \approx 0$. See Table 4 in Appendix for the similar results without L_2 normalization.

Categorical Data Analysis

An additional advantage of FMC is the ability to incorporate categorical data into clustering by using the categorical (multinoulli) mixture model, whose formulation is given in Eqs. (84) and (85) in the Appendix. Tables 5 and 6 in the Appendix show that FMC performs well with categorical data. Note that the baseline methods do not natively handle categorical variables; nontrivial modifications are required.

Modification of FMC for Multinary Sensitive Attributes

To apply FMC to multinary sensitive attributes, i.e., $M > 2$, we propose the following modification of Δ :

$$\max_{k \in [K]} \frac{2}{M(M-1)} \sum_{\substack{s_1, s_2 \in [M] \\ s_1 \neq s_2}} \Delta_{s_1, s_2}, \quad (14)$$

where

$$\Delta_{s_1, s_2} := \left| \frac{\sum_{x_i \in \mathcal{D}^{(s_1)}} \psi_k(x_i; \Theta)}{N_{s_1}} - \frac{\sum_{x_i \in \mathcal{D}^{(s_2)}} \psi_k(x_i; \Theta)}{N_{s_2}} \right|. \quad (15)$$

Table 7 in the Appendix compares VFC and FMC on Bank and Credit datasets where the sensitive attribute has three categories. See the experimental details in the Appendix. The results confirm that the modified FMC performs well.

Parameter Impact Studies

The number of clusters K We analyze how the number of clusters K affects the performance of FMC, and we find that FMC performs well regardless of the choice of K .

That is, the negative log-likelihood decreases as K increases, while the fairness level Δ remains sufficiently low.

See Fig. 8 in the Appendix for the results.

Choice of covariance structure in GMM We consider a diagonal covariance matrix instead of the isotropic one in the GMM to assess whether a more flexible covariance structure improves clustering. Fig. 9 in the Appendix suggests that the diagonal covariance matrix is generally beneficial, which is an additional advantage of FMC over SFC and VFC.

Discussion

In this work, we have proposed FMC, a finite mixture model based fair clustering algorithm, which can be easily scaled-up through mini-batch and sub-sample learning. Moreover, FMC can handle both continuous and categorical data.

The main idea of FMC is to parameterize the assignment map. We leave the application of this idea to existing fair clustering algorithms (e.g., VFC) for future work.

Choosing the number of clusters K is also important. There is extensive research on estimating K in finite mixture models (Schwarz 1978; Biernacki, Celeux, and Govaert 2002; Richardson and Green 1997; Miller and Harrison 2018). We will investigate how to adapt these estimators to fair clustering in future work.

Acknowledgements

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A5A7083908); Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-II220184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics); Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.RS-2021-II211343; Artificial Intelligence Graduate School Program (Seoul National University)], and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00556079).

References

- Agarwal, A.; Beygelzimer, A.; Dudík, M.; Langford, J.; and Wallach, H. 2018. A reductions approach to fair classification. In *International conference on machine learning*, 60–69. PMLR.
- Angwin, J.; Larson, J.; Mattu, S.; and Kirchner, L. 2022. Machine bias. In *Ethics of data and analytics*, 254–264. Auerbach Publications.
- Backurs, A.; Indyk, P.; Onak, K.; Schieber, B.; Vakilian, A.; and Wagner, T. 2019. Scalable fair clustering. In *International conference on machine learning*, 405–413. PMLR.
- Barocas, S.; and Selbst, A. D. 2016. Big data’s disparate impact. *Calif. L. Rev.*, 104: 671.
- Bera, S.; Chakrabarty, D.; Flores, N.; and Negahbani, M. 2019. Fair algorithms for clustering. *Advances in Neural Information Processing Systems*, 32.
- Biernacki, C.; Celeux, G.; and Govaert, G. 2002. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE transactions on pattern analysis and machine intelligence*, 22(7): 719–725.
- Brown, P. F.; Della Pietra, V. J.; deSouza, P. V.; Lai, J. C.; and Mercer, R. L. 1992. Class-Based n -gram Models of Natural Language. *Computational Linguistics*, 18(4): 467–480.
- Butnaru, A. M.; and Ionescu, R. T. 2017. From image to text classification: A novel approach based on clustering word embeddings. *Procedia computer science*, 112: 1783–1792.
- Calders, T.; Kamiran, F.; and Pechenizkiy, M. 2009. Building classifiers with independency constraints. In *2009 IEEE international conference on data mining workshops*, 13–18. IEEE.
- Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2017. Fair clustering through fairlets. *Advances in neural information processing systems*, 30.
- Chouldechova, A. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2): 153–163.
- Dempster, A. P.; Laird, N. M.; and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1): 1–22.
- Diez, M.; Burget, L.; Wang, S.; Rohdin, J.; and Černocký, J. 2019. Bayesian HMM Based x-Vector Clustering for Speaker Diarization. In *Interspeech 2019*, 346–350.
- Donini, M.; Oneto, L.; Ben-David, S.; Shawe-Taylor, J. S.; and Pontil, M. 2018. Empirical risk minimization under fairness constraints. *Advances in neural information processing systems*, 31.
- Dua, D.; Graff, C.; et al. 2017. UCI machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>, 7(1): 62. Accessed: 2025-01-10.
- Esmaeili, S.; Brubach, B.; Srinivasan, A.; and Dickerson, J. 2021. Fair clustering under a bounded cost. *Advances in Neural Information Processing Systems*, 34: 14345–14357.
- Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 259–268.
- Gepperth, A.; and Pfüll, B. 2021. Gradient-Based Training of Gaussian Mixture Models for High-Dimensional Streaming Data. *Neural Process. Lett.*, 53(6): 4331–4348.
- Gitiaux, X.; and Rangwala, H. 2021. Learning Smooth and Fair Representations. In Banerjee, A.; and Fukumizu, K., eds., *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, 253–261. PMLR.
- Guo, J.; Zhu, X.; Zhao, C.; Cao, D.; Lei, Z.; and Li, S. Z. 2020. Learning meta face recognition in unseen domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6163–6172.
- Harb, E.; and Lam, H. S. 2020. KFC: A Scalable Approximation Algorithm for k -center Fair Clustering. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 14509–14519. Curran Associates, Inc.
- Hosseini, R.; and Sra, S. 2020. An alternative to EM for Gaussian mixture models: batch and stochastic Riemannian optimization. *Math. Program.*, 181(1): 187–223.
- Karimi, B.; Lavielle, M.; and Moulines, É. 2019. On the Convergence Properties of the Mini-Batch EM and MCEM Algorithms. Working paper or preprint.
- Kim, K.; Lee, J.; Park, S.; and Kim, Y. 2025. Fair Clustering via Alignment. arXiv:2505.09131.
- Kleinberg, J.; Ludwig, J.; Mullainathan, S.; and Rambachan, A. 2018. Algorithmic fairness. In *Aea papers and proceedings*, volume 108, 22–27. American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203.
- Kleindessner, M.; Samadi, S.; Awasthi, P.; and Morgenstern, J. 2019. Guarantees for spectral clustering with fairness constraints. In *International conference on machine learning*, 3458–3467. PMLR.
- Le, Q. V. 2013. Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing*, 8595–8598. IEEE.
- Lee, S.; Choi, C.; and Son, Y. 2024. Deep time-series clustering via latent representation alignment. *Knowledge-Based Systems*, 303: 112434.
- Li, P.; Zhao, H.; and Liu, H. 2020. Deep fair clustering for visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9070–9079.
- Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2021. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6): 1–35.
- Miller, J. W.; and Harrison, M. T. 2018. Mixture models with a prior on the number of components. *Journal of the American Statistical Association*, 113(521): 340–356.
- Mittal, H.; Pandey, A. C.; Saraswat, M.; Kumar, S.; Pal, R.; and Modwel, G. 2022. A comprehensive survey of image segmentation: clustering methods, performance parameters,

- and benchmark datasets. *Multimedia Tools and Applications*, 81(24): 35001–35026.
- Mohri, M.; Rostamizadeh, A.; and Talwalkar, A. 2018. *Foundations of machine learning*. MIT press.
- Muhammad, M. 2021. Recommendation System Using User-Based Collaborative Filtering and Spectral Clustering. EAI.
- Nesterov, Y. 2013. Gradient methods for minimizing composite functions. *Mathematical programming*, 140(1): 125–161.
- Paparrizos, J.; and Gravano, L. 2016. k-Shape: Efficient and Accurate Clustering of Time Series. *SIGMOD Rec.*, 45(1): 69–76.
- Quadrianto, N.; Sharmanska, V.; and Thomas, O. 2019. Discovering Fair Representations in the Data Domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Richardson, S.; and Green, P. J. 1997. On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 59(4): 731–792.
- Robbins, H.; and Siegmund, D. 1971. A convergence theorem for non negative almost supermartingales and some applications. In *Optimizing methods in statistics*, 233–257. Elsevier.
- Schwarz, G. 1978. Estimating the dimension of a model. *The annals of statistics*, 461–464.
- Shalev-Shwartz, S.; and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.
- Titterton, D.; Smith, A.; and Makov, U. 1985. *Statistical Analysis of Finite Mixture Distributions*. Applied section. Wiley. ISBN 9780471907633.
- Vershynin, R. 2018. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.
- Widiyaningtyas, T.; Hidayah, I.; and Adji, T. B. 2021. Recommendation algorithm using clustering-based upcsim (Cb-upcsim). *Computers*, 10(10): 123.
- Wolf, M. M. 2023. Mathematical foundations of supervised learning.
- Wu, C. F. J. 1983a. On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1): 95–103.
- Wu, C. J. 1983b. On the convergence properties of the EM algorithm. *The Annals of statistics*, 95–103.
- Xu, D.; Yuan, S.; Zhang, L.; and Wu, X. 2018. FairGAN: Fairness-aware Generative Adversarial Networks. In *2018 IEEE International Conference on Big Data (Big Data)*, 570–575.
- Xu, L.; and Jordan, M. I. 1996a. On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Computation*, 8(1): 129–151.
- Xu, L.; and Jordan, M. I. 1996b. On convergence properties of the EM algorithm for Gaussian mixtures. *Neural computation*, 8(1): 129–151.
- Zafar, M. B.; Valera, I.; Rognier, M. G.; and Gummadi, K. P. 2017. Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*, 962–970. PMLR.
- Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; and Dwork, C. 2013. Learning Fair Representations. In Dasgupta, S.; and McAllester, D., eds., *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, 325–333. Atlanta, Georgia, USA: PMLR.
- Zeng, P.; Li, Y.; Hu, P.; Peng, D.; Lv, J.; and Peng, X. 2023. Deep fair clustering via maximizing and minimizing mutual information: Theory, algorithm and metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23986–23995.
- Zeng, Z.; Islam, R.; Keya, K. N.; Foulds, J.; Song, Y.; and Pan, S. 2021. Fair Representation Learning for Heterogeneous Information Networks. arXiv:2104.08769.
- Zhang, Y.; Wang, Z.; and Shang, J. 2023. Clusterllm: Large language models as a guide for text clustering. arXiv preprint arXiv:2305.14871.
- Zhou, Y.; Huang, S.-C.; Fries, J. A.; Youssef, A.; Amrhein, T. J.; Chang, M.; Banerjee, I.; Rubin, D.; Xing, L.; Shah, N.; et al. 2021. Radfusion: Benchmarking performance and fairness for multimodal pulmonary embolism detection from ct and ehr. arXiv preprint arXiv:2111.11665.
- Ziko, I. M.; Yuan, J.; Granger, E.; and Ayed, I. B. 2021. Variational fair clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11202–11209.