Fair Wasserstein Coresets

Zikai Xiong[2]* **Niccolò Dalmasso**[1] **Vamsi K. Potluru**[1]
Tucker Balch[1] **Manuela Veloso**[1]

[1]J.P. Morgan AI Research, New York,
[2]Massachusetts Institute of Technology

niccolo.dalmasso@jpmchase.com

Abstract

Recent technological advancements have given rise to the ability of collecting vast amounts of data, that often exceed the capacity of commonly used machine learning algorithms. Approaches such as coresets and synthetic data distillation have emerged as frameworks to generate a smaller, yet representative, set of samples for downstream training. As machine learning is increasingly applied to decision-making processes, it becomes imperative for modelers to consider and address biases in the data concerning subgroups defined by factors like race, gender, or other sensitive attributes. Current approaches focus on creating fair synthetic representative samples by optimizing local properties relative to the original samples. These methods, however, are not guaranteed to positively affect the performance or fairness of downstream learning processes. In this work, we present Fair Wasserstein Coresets (`FWC`), a novel coreset approach which generates fair synthetic representative samples along with sample-level weights to be used in downstream learning tasks. `FWC` aims to minimize the Wasserstein distance between the original datasets and the weighted synthetic samples while enforcing (an empirical version of) demographic parity, a prominent criterion for algorithmic fairness, via a linear constraint. We show that `FWC` can be thought of as a constrained version of Lloyd's algorithm for k-medians or k-means clustering. Our experiments, conducted on both synthetic and real datasets, demonstrate the scalability of our approach and highlight the competitive performance of `FWC` compared to existing fair clustering approaches, even when attempting to enhance the fairness of the latter through fair pre-processing techniques.

1 Introduction

In recent years, the rapid pace of technological advancement has provided the ability of collecting, storing and processing massive amounts of data from multiple sources [34]. As the volume of data continues to surge, it often surpasses both the available computational resources as well as the capacity of machine learning algorithms. In response to this limitation, dataset distillation approaches aim to reduce the amount of data by creating a smaller, yet representative, set of samples; see [40, 24] for comprehensive reviews on the topic. Among those approaches, coresets provide a weighted subset of the original data that achieve similar performance to the original dataset in (usually) a specific machine learning task, such as clustering [16], Bayesian inference [6], online learning [4] and classification [11], among others.

In tandem with these developments, the adoption of machine learning techniques has seen a surge in multiple decision-making processes that affect society at large [36, 41]. This proliferation of machine

*Work done during an internship at J.P. Morgan AI Research.

NeurIPS 2023 Workshop on Synthetic Data for Empowering ML Research.

learning applications has highlighted the need to mitigate inherent biases in the data, as these biases can significantly impact the equity of machine learning models and their decisions [9]. Among many definitions of algorithmic fairness, demographic parity is one of the most prominently used metric [18], enforcing the distribution of an outcome of a machine learning model to not differ dramatically across different subgroups in the data.

Current methodologies for generating a smaller set of fair representative samples focus on the local characteristics of these samples with respect to the original dataset. For instance, [8, 19, 2, 15] enforce representative points obtained by clustering to maintain the same proportion of points from each subgroup in each cluster. In another line of work, [20, 26, 28, 37, 7] create representative points by ensuring that points in the original dataset each have at least one representative point within a given distance in the feature space. While these methods can successfully reduce clustering cost and ensure a more evenly spread-out distribution of representative points in the feature space, it is unclear whether such representative samples can positively affect performance or discrimination reduction in downstream learning processes. As the induced distribution of the representative points might be far away from the original dataset distribution, downstream machine learning algorithm might lose significant performance without necessarily reducing biases in the original data.

In this work, we introduce Fair Wasserstein Coresets (`FWC`), a novel coreset approach that not only generates synthetic representative samples but also assigns sample-level weights to be used in downstream learning task. The key of `FWC` is generating synthetic samples by minimizing the Wasserstein distance between the distribution of the original datasets and that of the weighted synthetic samples, while simultaneously enforcing an empirical version of demographic parity. The Wasserstein distance is particularly useful when generating coresets, as downstream model performance is tied to the Wasserstein distance's dual formulation (Section 2). Section 3 presents the minimization problem in which demographic parity is introduced via linear constraint, and shows that the problem can be solved by first reformulating the objective function and then using a majority minimization algorithm. In addition, in Section 4.1 we show how the unconstrained version of `FWC` is equivalent to Lloyd's algorithm for k-means and k-medians clustering, extending its applicability beyond fairness applications. Finally, to empirically validate the effectiveness of `FWC`, Section 5 provides experiments on both synthetic and real-world datasets. When compared against current approaches, `FWC` achieves competitive performance, even when we enhance the fairness of existing approaches using existing fair pre-processing techniques.

Notation We denote the original dataset samples $\{Z_i\}_{i=1}^n$, with $Z_i = (D_i, X_i, Y_i) \in \mathcal{Z} = (\mathcal{D} \times \mathcal{X} \times \mathcal{Y})$. In this context, D represents one or more sensitive features such as gender or race, X denotes the non-sensitive features, and Y is a discrete decision outcome. Given a set of weights $\{\theta\}_{i=1}^n$, define $p_{Z;\theta}$ the weighted distribution of a dataset $\{Z_i\}_{i=1}^n$ as $p_{Z;\theta} \overset{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^n \theta_i \delta_{Z_i}$, where δ_x stands for the Dirac (unit mass) distribution at point $x \in \mathcal{X}$. In this notation, the empirical distribution of the original dataset can be written by setting $\theta_i = e_i = 1$ for any i, i.e., $p_{Z;e} = \frac{1}{n}\sum_{i=1}^n e_i \delta_{Z_i}$. For a matrix A, $A^\top$ denotes its transpose. For two vectors (or matrices) $\langle u, v\rangle \overset{\text{def.}}{=} \sum_i u_i v_i$ is the canonical inner product (the Frobenius dot-product for matrices). We define $\mathbf{1}_m \overset{\text{def.}}{=} (1, \ldots, 1) \in \mathbb{R}_+^m$. We denote $\mathcal{M}_+^1(\mathcal{X})$ the set of probability distributions over a metric space $\mathcal{X}$.

2 Background

Measure Coreset Let $\mathcal{F}$ be the hypothesis set for a learning problem. Every function $f \in \mathcal{F}$ maps from $\mathcal{Z}$ to $\mathbb{R}$. We follow the definition in [10] and say that a weighted dataset $\{\widehat{Z}_i = (\widehat{X}_i, \widehat{D}_i, \widehat{Y}_i), \theta_i\}_{i=1}^m$ is an ϵ-coreset if

$$\sup_{f \in \mathcal{F}} \left| \text{cost}(Z, w, f) - \text{cost}(\widehat{Z}, \theta, f) \right| \leq \epsilon \text{ , where } \text{cost}(Z, w, f) \overset{\text{def.}}{=} \sum_{i=1}^n w_i f(Z_i). \tag{2.1}$$

In this definition, finding an ϵ-coreset corresponds to obtaining a weighted compressed dataset $\{\widehat{Z}_i, \theta_i\}_{i=1}^m$ such that $m \ll n$ and with a small ϵ. This definition of ϵ-coreset highlights the importance of preserving the performance of downstream learning models on the original dataset $\{Z_i\}_{i=1}^n$.

Wasserstein Distance The general Wasserstein distance (or optimal transport metric) between two probability distributions $(\mu, \nu) \in \mathcal{M}_+^1(\mathcal{X}) \times \mathcal{M}_+^1(\mathcal{X})$ supported on two metric spaces $(\mathcal{X}, \mathcal{X})$ is defined as the optimal objective of the (possibly infinite-dimensional) linear program (LP):

$$\mathcal{W}_c(\mu, \nu) \stackrel{\text{def.}}{=} \min_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X} \times \mathcal{X}} c(x, y) \mathrm{d}\pi(x, y), \tag{2.2}$$

where $\Pi(\mu, \nu)$ is the set of couplings composed of joint probability distributions over the product space $\mathcal{X} \times \mathcal{X}$ with imposed marginals (μ, ν). Equation (2.2) is also called the Kantorovitch formulation of optimal transport [22]. Here, $c(x, y)$ represents the "cost" to move a unit of mass from x to y. A typical choice in space $\mathcal{X}$ with metric $d_{\mathcal{X}}$ is $c(x, y) = d_{\mathcal{X}}(x, y)^p$ for $p \geq 1$, and then $\mathcal{W}_c^{1/p}$ corresponds to the p-Wasserstein distance between probability measures. Using the Wasserstein distance between distributions is particularly useful as it provides a bound for functions applied to samples from those distributions. In other words, define the following deviation:

$$d(\mu, \nu) \stackrel{\text{def.}}{=} \sup_{f \in \mathcal{F}} \left| \mathbb{E}_{z\sim\mu} f(z) - \mathbb{E}_{z\sim\nu} f(z) \right| ,$$

where $\mathcal{F}$ is a family of functions f. If $\mathcal{F} = Lip_1$, the class of Lipschitz-continuous functions with Lipschitz constant of 1, the deviation $d(\mu, \nu)$ is equal to the 1-Wasserstein distance [35, 38]. Analog bounds can be derived for the 2-Wasserstein distance when $\mathcal{F} = \left\{ f \mid \|f\|_{\mathcal{S}^1(\mu)} \leq 1 \right\}$, the class of functions with unitary norm over the Sobolev space $\mathcal{S} = \left\{ f \in L^2 \mid \partial_{x_i} f \in L^2 \right\}$ [10]. This fact provides a theoretical intuition for evaluating the quality of a coreset. The closer the Wasserstein distance between the empirical distribution formed by the coreset and the one formed by the original dataset, then the smaller the left-hand side of Equation (2.1) could be.

Demographic parity Demographic parity (DP), or statistical parity, requires the outcome variable and sensitive features to be independent [12], and is arguably the most widely studied fairness criterion to date [18]. In this work, we adopt the demographic parity definition introduced by [5], which translates to requiring the conditional distribution of the outcome across each sensitive groups $p(y|D = d)$ to be close to the marginal distribution of the outcome $p(y)$ in a given dataset.

3 FWC: Fair Wasserstein Coresets

Given a dataset $\{Z_i\}_{i=1}^n$, our goal is to find a set of samples $\{\widehat{Z}_j\}_{j=1}^m$ and weights $\{\theta_j\}_{j=1}^m$ such that $m \ll n$ and that the Wasserstein distance between $p_{Z;e}$ and $p_{\widehat{Z};\theta}$, $\mathcal{W}_c(p_{\widehat{Z};\theta}, p_{Z;w})$, is as small as possible. To control for discrimination, we adopt the fairness constraints proposed by [5], which translates to requiring the conditional distribution under the weights $\{\theta_i\}_{i\in[n]}$ to closely align with a target distribution p_{Y_T} for all possible values of D,

$$J\left(p_{\widehat{Z};\theta}(\widehat{Y} = y|\widehat{D} = d), p_{Y_T}(y)\right) \leq \epsilon, \; \forall\, d \in \mathcal{D}, y \in \mathcal{Y} \tag{3.1}$$

where $J(\cdot, \cdot)$ denotes a distance function between distributions and ϵ is a parameter that determines the maximum fairness violation. We will use the shorthand $p_{\widehat{Z};\theta}(y|d)$ for $p_{\widehat{Z};\theta}(\widehat{Y} = y|\widehat{D} = d)$. We slightly differ from [5] as we define J as the subsequent symmetric probability ratio measure:

$$J(p, q) = \max\left\{ \tfrac{p}{q} - 1, \tfrac{q}{p} - 1 \right\} , \tag{3.2}$$

which we believe to be more practically as it is symmetric with respect to p and q[2].

Our goal can then be reformulated as the following optimization problem:

$$\begin{aligned} \min_{\theta \in \Delta_m, \widehat{Z} \in \mathcal{Z}^m} \quad & \mathcal{W}_c(p_{\widehat{Z};\theta}, p_{Z;e}) \\ \text{s.t.} \quad & J\left(p_{\widehat{Z};\theta}(y|d), p_{Y_T}(y)\right) \leq \epsilon, \; \forall\, d \in \mathcal{D}, y \in \mathcal{Y}. \end{aligned} \tag{3.3}$$

Here Δ_m is the set of valid weights $\{\theta \in \mathbb{R}_+^m : \sum_{i=1}^m \theta_i = m\}$.

[2] As opposed to the original definition $J(p, q) = |\frac{p}{q} - 1|$ by [5]. The two definitions are equivalent when $p > q$ and similar when p is not much smaller than q.

In practice, all the possible values of Y and D are known a priori, so we can fix the values of $\widehat{Y}$ and $\widehat{D}$ and optimize only over $\widehat{X}$ and θ. The following lemma shows that this in fact does not affect the optimization problem:

As the possible values of Y_i and D_i are known a priori, and there is only a limited number of them, rather than optimizing over them we fix the values of $\widehat{Y}$ and $\widehat{D}$ and optimize only over $\widehat{X}$ and θ.

Lemma 3.1. *For any $m > 0$, the best fair Wasserstein coreset formed by m data points $\{\widehat{Z}_i : i \in [m]\}$ is no better than the best fair Wasserstein coreset formed by $m|\mathcal{D}||\mathcal{Y}|$ data points $\{(d, X_i, y)_i : i \in [m], d \in \mathcal{D}, y \in \mathcal{Y}\}$.*

Proof. Once we generate $m|\mathcal{D}||\mathcal{Y}|$ data points, the feasible set of the latter Wasserstein coreset contains the feasible set of the former Wasserstein coreset. □

Hence, we simply set the proportions of $\{(\widehat{D}_i, \widehat{Y})_i\}_{i\in[m]}$ in the coresets to be similar to their respective proportions in the original dataset. The optimization problem then reduces to finding the features in the coreset $\{\widehat{X}_j\}_{j=1}^m$ and corresponding weights $\{\theta_j\}_{j=1}^m$ such that:

$$\begin{aligned} \min_{\theta\in\Delta_m, \widehat{X}\in\mathcal{X}^m} \quad & \mathcal{W}_c(p_{\widehat{Z};\theta}, p_{Z;e}) \\ \text{s. t.} \quad & J\left(p_{\widehat{Z};\theta}(y|d), p_{Y_T}(y)\right) \le \epsilon, \ \forall\, d \in \mathcal{D}, y \in \mathcal{Y} \,. \end{aligned} \tag{3.4}$$

We now take the following steps to solve the optimization problem in (3.4): (1) we first express the fairness constraints linearly, (2) we add artificial variables to the objective function and (3) we simplify the optimization problem to obtain a continuous non-convex function of the $\{X_j\}_{j=1}^m$ and (4) we propose a majority minimization to solve the optimization problem.

Step 1. Equivalent linear constraints. Firstly, we show that the constraint in Equation (3.1) can be reformulated as linear constraints on θ of the form $A\theta \ge \mathbf{0}$. The conditional probability in constraint (3.1) can be rewritten as:

$$p_{\widehat{Z};\theta}(y|d) = \frac{\sum_{i\in[m]:\widehat{D}_i=d,\widehat{Y}_i=y} \theta_i}{\sum_{i\in[n]:\widehat{D}_i=d} \theta_i} \,.$$

By substituting the definition of the distance $J(\cdot,\cdot)$ from Equation (3.2), the fairness constraints equivalently become linear constraints on $\{\theta_i\}_{i=1}^n$ (via inverting a fractional linear transformation), taking the following form for all $d \in \mathcal{D}, y \in \mathcal{Y}$:

$$\begin{aligned} \sum_{i\in[m]:\widehat{D}_i=d,\widehat{Y}_i=y} \theta_i &\le (1+\epsilon)\cdot p_{Y_T}(y) \cdot \sum_{i\in[m]:\widehat{D}_i=d} \theta_i \,, \\ \sum_{i\in[m]:\widehat{D}_i=d,\widehat{Y}_i=y} \theta_i &\ge \frac{1}{1+\epsilon}\cdot p_{Y_T}(y) \cdot \sum_{i\in[m]:\widehat{D}_i=d} \theta_i \,. \end{aligned} \tag{3.5}$$

In total, Equation (3.5) defines $2|\mathcal{Y}||\mathcal{D}|$ linear constraints on θ in the format of $A\theta \ge \mathbf{0}$, where A is a $2|\mathcal{Y}||\mathcal{D}|$-row matrix[3].

Step 2. Reformulate the objective function by introducing artificial variables. Regarding the objective, when fixing $\widehat{X}$, the objective function of the Wasserstein distance can be equivalently formulated as a linear program with mn variables [32]. Let $C(\widehat{X})$ denote the matrix of the transportation costs, in which the components are defined as follows,

$$(C(\widehat{X}))_{ij} \overset{\text{def.}}{=} c(Z_i, \widehat{Z}_j), \ \text{for } i \in [n], j \in [m] \,.$$

Therefore, now the problem in Equation (3.4) is equivalent to

$$\begin{aligned} \min_{\widehat{X}\in\mathcal{X}^m, \theta\in\Delta_m, P\in\mathbb{R}^{n\times m}} \quad & \langle C(\widehat{X}), P\rangle \\ \text{s.t.} \quad & Pe = \frac{1}{n}\cdot \mathbf{1}_n, \ P^\top e = \frac{1}{m}\cdot\theta, P \ge 0, \ A\theta \ge 0 \,. \end{aligned} \tag{3.6}$$

[3]Note that when Y is binary, e.g., $\mathcal{Y} = \{0, 1\}$, half of the linear constraints induced by Equation (3.1) are redundant and can be removed.

Step 3. Reduce to optimization problem of $\widehat{X}$. Note that for any feasible $(\widehat{X}, \theta, P)$ of Equation (3.6), it holds that $\theta = m \cdot P^\top e$. Furthermore, given $\frac{1}{n} \cdot \mathbf{1}_n^\top e = 1$, if $Pe = \frac{1}{n} \cdot \mathbf{1}_n$, it follows that $\theta^\top e = m \cdot e^\top Pe = \frac{m}{n} \cdot \mathbf{1}_n^\top e = m$. Consequently, Equation (3.6) is then simplified by replacing variables θ with $m \cdot P^\top e$.

$$\begin{aligned} \min_{\widehat{X} \in \mathcal{X}^m, P \in \mathbb{R}^{n \times m}} \quad & \langle C(\widehat{X}), P \rangle \\ \text{s.t.} \quad & Pe = \frac{1}{n} \cdot \mathbf{1}_n,\ P \geq 0,\ AP^\top e \geq 0\,. \end{aligned} \tag{3.7}$$

Define the function F of C by the following optimization problem

$$F(C) \stackrel{\text{def.}}{=} \left(\begin{array}{ll} \min_{P \in \mathbb{R}^{n \times m}} & \langle C, P \rangle \\ \text{s.t.} & Pe = \frac{1}{n} \cdot \mathbf{1}_n,\ P \geq 0,\ AP^\top e \geq 0 \end{array} \right) \tag{3.8}$$

and then Equation (3.7) is equivalent to

$$\min_{\widehat{X} \in \mathcal{X}^m} \ F(C(\widehat{X}))\,. \tag{3.9}$$

Here the objective is continuous but nonconvex with respect to $\widehat{X}$. Once the optimal $\widehat{X}^\star$ is solved, then the optimal $P^\star$ of Equation 3.7 is obtained by solving problem 3.8 with C replaced with $C(\widehat{X}^\star)$. Finally, the optimal $\theta^\star$ follows by the equation $\theta^\star = m \cdot (P^\star)^\top e$. We now provide a majority minimization method for solving Equation (3.9).

4 Majority Minimization for Solving the Reformulated Problem.

The problem in Equation (3.8) is a huge-scale linear program with $O(n)$ constraints and $O(mn)$ nonnegative variables. Its size becomes computationally prohibitive for large values of n and m. [39] proposes a fast algorithm for (3.8), via applying cutting plane methods on the Lagrangian dual problems with reduced dimension. Although [39] primarily addresses the scenario where $m = n$, their approach is directly applicable to cases where $m \neq n$. The theoretical complexities of [39] can also be extended to this setting. Experiments in [39] demonstrate that the overall computational complexity of this method is considerably lower than that of interior-point or simplex methods. Following the approach established in [39], we have the following lemmas on computing subgradients and separation oracles, which are going to be used to solve the problem in Equation (3.8).

Lemma 4.1. *An easily accessible oracle exists for obtaining a minimizer $P^\star$ for problem (3.8).*

Lemma 4.2. *The function $F(C)$ is a concave continuous function of C and has easily computed function values, subgradients.*

Proof. The proof directly uses the concavity of the minimum LP's optimal objective on the cost function. Further, since the feasible set of Equation (3.8) is bounded, the optimal solution $F(C)$ is continuous with respect to C.

Next, the subgradient at point C is equal to the corresponding optimal solution $P^\star$ in Equation (3.8), using a similar proof with Lemma 4.1. Finally, due to Lemma 4.1, computing the subgradients the function values of F at C is also feasible. □

We now approach solving problem in Equation (3.9) by using a majority minimization algorithm. Majority minimization refers to the process of defining a surrogate function that upper bounds the objective function, so that optimizing the surrogate function improves the objective function [29, 23]. We define a surrogate function of any $\widehat{X}^k \in \mathcal{X}^m$. Let $P_k^\star$ be the minimizer of Equation (3.8) with the cost $C = C(\widehat{X}^k)$, then let

$$g(\widehat{X}; \widehat{X}^k) \stackrel{\text{def.}}{=} \langle C(\widehat{X}), P_k^\star \rangle\,. \tag{4.1}$$

We have that $g(\widehat{X}; \widehat{X}^k) = F(C(\widehat{X}))$ when $\widehat{X} = \widehat{X}^k$. When $\widehat{X} \neq \widehat{X}^k$, $g(\widehat{X}; \widehat{X}^k) \geq F(C(\widehat{X}))$, which means $g(\widehat{X}; \widehat{X}^k)$ is an upper bound of the objective function $F(C(\widehat{X}))$. Moreover, although

$F(C(\widehat{X}))$ might not be convex with respect to $\widehat{X}$, $g(\widehat{X};\widehat{X}^k)$, as an upper bound of $F(C(\widehat{X}))$, is convex and the minimizer on $\mathcal{X}^m$,

$$\min_{\widehat{X}\in\mathcal{X}^m} g(\widehat{X};\widehat{X}^k) \tag{4.2}$$

can be efficiently solved. The overall algorithm to minimize the problem in Equation (3.9) is then summarized in Algorithm 1.

Algorithm 1 Majority Minimization for Solving Equation (3.9)

```
1:  Initial feature vectors X̂^k and k = 0
2:  while True do
3:     Update the cost matrix: C ← C(X̂^k)
4:     Update the assignment matrix: P* solves Equation (3.8) via Algorithm 1 in [39]
5:     if X̂^k ∈ arg min_X̂ g(X̂; X̂^k) then
6:        return X̂^k
7:     else
8:        Update the feature vectors of the coreset: X̂^{k+1} ← arg min_X̂ g(X̂; X̂^k)
9:        k ← k + 1
10:    end if
11: end while
```

We note that the problem in Equation (4.2) can be actually written as the following unconstrained problem

$$\min_{\widehat{X}_i\in\mathcal{X}:i\in[m]} \sum_{i\in[m]}\sum_{j\in[n]} c(\widehat{Z}_i, Z_j)P_{ij} \tag{4.3}$$

in which each component of P is nonnegative. Furthermore, as shown in Lemma 4.2, the matrix P is very sparse, sometimes with at most n non-zeros. Moreover, Equation (4.3) could be separated into the following m problems,

$$\min_{\widehat{X}_i\in\mathcal{X}} \sum_{j\in[n]} c(\widehat{Z}_i, Z_j)P_{ij} \quad \text{for } i\in[m]\ . \tag{4.4}$$

Note that each problem is a convex problem so gradient methods could already converge to global minimizers of Equation (4.4). Furthermore, under some special conditions, it could be solved even more efficiently.

1. If $\mathcal{X}$ is convex and $c(Z,\widehat{Z}) \overset{\text{def.}}{=} \|Z-\widehat{Z}\|_2^2$, then the minimizer of Equation (4.4) is the weighted average $\sum_{j\in[n]} P_{ij}X_j / \sum_{j\in[n]} P_{ij}$ for each $i\in[m]$.
2. If $\mathcal{X}$ is convex and $c(Z,\widehat{Z}) \overset{\text{def.}}{=} \|Z-\widehat{Z}\|_1$, then the minimizer of Equation (4.4) requires sorting the costs coordinate-wisely and finding the median.
3. If creating new feature vectors is not permitted and then $\mathcal{X} = \{X_i : i\in[n]\}$, solving Equation (4.4) only requires finding the smallest $\sum_{j\in[n]} c((d,X_i,y),Z_j)P_{ij}$ over $i\in[n]$. Note that the matrix P is highly sparse so it is not expensive.

Finally, before providing a convergence result, we include the following assumption regarding the minimizer of the problem in Equation (3.9):

Assumption 1 (Assumption 1 of [39])**.** *The problem* $\min_{\widehat{X}\in\mathcal{X}^m} g(\widehat{X};\widehat{X}^k)$ *has a unique minimizer for the optimal solution.*

We are now ready to show the convergence of Algorithm 1 below:

Lemma 4.3 (Informal convergence results)**.** *Under Assumption 1, the objective value is monotonically decreasing, i.e.,* $F(C(\widehat{X}^{k+1})) \le F(C(\widehat{X}^k))$ *for any* $k\ge 0$*. And once* $\widehat{X}^k \in \arg\min_{\widehat{X}} g(\widehat{X};\widehat{X}^k)$*, then* $\widehat{X}^k$ *is a first-order stationary point. Moreover, the algorithm method must terminate within finite iterations.*

Proof. The monotonically decreasing follows:

$$F(C(\widehat{X}^{k+1})) \leq g(\widehat{X}^{k+1};\widehat{X}^k) \leq g(\widehat{X}^k;\widehat{X}^k) = F(C(\widehat{X}^k))\;. \tag{4.5}$$

To be specific, the inequality in Equation (4.5) holds strictly when $\widehat{X}^{k+1} \neq \widehat{X}^k$, or equivalently $\widehat{X}^k \notin \arg\min_{\widehat{X}} g(\widehat{X};\widehat{X}^k)$. Once $\widehat{X}^k \in \arg\min_{\widehat{X}} g(\widehat{X};\widehat{X}^k)$, then $\widehat{X}^k$ is already a global minimizer of the convex upper bound $g(\cdot;\widehat{X}^k)$ for $F(C(\cdot))$, which means X^k is a first-order stationary point of $F(C(\cdot))$.

As for the finite convergence, there are only finite possible $P^\star$ as the minimizer of Equation 3.8, as proven in Lemma 1 by [39]. However, before the majority minimization converges, Equation (4.5) holds strictly. Therefore, after finite iterations, Equation (4.5) must hold with the inequality holds at equality. □

4.1 An alternative view: Generalized clustering algorithm

When the fairness constraints are absent, the optimization problem in Equation (3.8) becomes:

$$F(C) \stackrel{\text{def.}}{=} \begin{pmatrix} \min_{P\in\mathbb{R}^{n\times m}} & \langle C, P\rangle \\ \text{s.t.} & Pe = \frac{1}{n}\cdot \mathbf{1}_n,\; P \geq \mathbf{0}_{n\times m} \end{pmatrix}\;. \tag{4.6}$$

The minimizer $P^\star$ of Problem 4.6 can be written in closed form. For each $i \in [n]$, let $C_{ij_i^\star}$ denote a smallest component on the i-th row of C. Then a minimizer $P^\star$ can be written as:

$$P^\star_{ij} \stackrel{\text{def.}}{=} \begin{cases} 0\,, & \text{if } j \neq j_i^\star \\ 1/n\,, & \text{if } j = j_i^\star \end{cases} \quad \text{for } i \in [n] \text{ and } j \in [m]\;. \tag{4.7}$$

Hence, without constraints, `FWC` corresponds to Lloyd's algorithm for clustering. Specifically, Lloyd's algorithm iteratively computes the centroid for each subset in the partition and subsequently re-partitions the input based on the closeness to these centroids [25]; these are the same operations FWC does in optimizing the surrogate function and solving problem 3.8. Thus, when $c(x, y)$ is correspondingly defined as $\|x - y\|_1$ or $\|x - y\|_2^2$, FWC corresponds to Lloyd's algorithm applied to k-medians or k-means problems, except the centroids have fixed values for $\widehat{D}$ and $\widehat{Y}$ (see Section 3).

5 Experiments

Runtime Analysis Firstly, we evaluate the runtime performance of `FWC` by creating a synthetic dataset of dimension n and features of dimension p, with the goal of creating a coreset of size m (see Supp. Mat. A.1 for more details). We fix two out of the three parameters to their default values $(n, m, p) = (5000, 250, 25)$ and vary the other across suitable ranges, to analyse the runtime and total number of iterations. Figure 1, top left, shows the runtime and number of iterations when increasing the original dataset sample size n from $1,000$ to $50,000$ (with average and standard deviation over 10 separate runs); both the runtime and number of iterations grow proportionally to the sample size n, with each iteration running below 2 seconds and a maximum runtime of around 5 minutes. Figure 2 in Supp. Mat. A.1 also shows that requiring a larger coreset size m converges faster in terms of number of iterations, but increases the iteration runtime as more representatives need to be computed.

Real Datasets We evaluate the performance of `FWC` on 4 real datasets widely used in the fairness literature [13]: (i) the Adult dataset [3], (ii) the German Credit dataset [17], (iii) the Communities and Crime dataset [33] and (iv) the Drug dataset [14]. For each datasets, we consider 3 different coreset sizes m, 5%, 10% and 20% (apart from the Adult dataset, in which we select coreset sizes of size 0.5%, 1% and 2% due to the large dataset size). Across these datasets, we compare with (a) `Fairlets` and `IndFair`, two fair clustering approaches by [2] and [7], (b) `K-Median Coresets`, a coreset approach by [1], (c) k-means [25] and k-medoids [27, 30], two classic clustering approaches and (d) `Uniform Subsampling` of the original dataset. For `FWC`, we consider three different values of the fairness violation hyper-parameters ϵ for the optimization problem in 3.4. Table 1 shows that `FWC` consistently achieves coresets that are closer in distribution to the original dataset with respect to the other methods, and are competitive in terms of clustering cost (although `FWC` not natively targeting clustering cost). Note that creating coresets for the Credit datasets is more challenging for `FWC` due

	Wasserstein Distance				Clustering Cost			
	Adult	*Credit*	*Crime*	*Drug*	*Adult*	*Credit*	*Crime*	*Drug*
`FWC` Ranking	**1/1/1**	**1**/3/3	**1/1/1**	**1/1/1**	5/5/5	**1**/5/4	**1/1**/2	**1/1**/2

Table 1: Ranking of `FWC` coresets for Wasserstein distance to the original datasets and clustering cost, for three coreset sizes m. `FWC` consistently creates coresets that are closer in distribution to the original dataset, and can obtain competitive clustering performance (see Supp. Mat. A for details).

to the high number of discrete features, which makes the minimization non-smooth in the feature space. Finally, Figure 1 shows the fairness-utility tradeoff indicated by the AUC and demographic disparity of a downstream multi-layer perception (MLP) classifier trained on the coresets created by each approach. `FWC` obtains a smaller disparity at the same accuracy compared to other approaches for the Drug and Adult datasets, and performance remains competitive even when using a fairness pre-processing approach [21]. See Figure 5 and Supp. Mat. A for more details on the comparison.

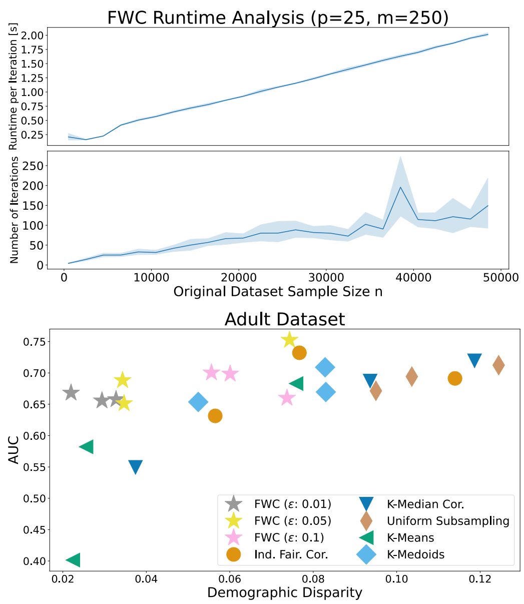

Figure 1: *Top left:* Runtime of `FWC` when changing the original dataset size n. *Others*: Fairness-utility tradeoff on Drug and Adult datasets for a downstream MLP classifier. `FWC` consistently achieves a comparable/better tradeoff, even when adjusting the other coresets with a fairness pre-processing technique [21]. Averages and standard deviations over 10 runs, see Supp. Mat. A for details.

6 Discussion and Conclusions

This paper introduces `FWC`, a novel coreset approach that not only generates synthetic representative samples but also assigns sample-level weights to enhance downstream learning tasks. `FWC` focuses on minimizing the Wasserstein distance between the distribution of the original datasets and that of the weighted synthetic samples, while simultaneously enforcing an empirical version of demographic parity, which ensures the generated samples to be representative but also fair with respect to sensitive attributes. We have demonstrated the effectiveness and scalability of `FWC` through a series of experiments conducted on both synthetic and real-world datasets. Our results have shown that `FWC` achieves competitive performance with existing fair clustering approaches, even when we apply fair pre-processing techniques to enhance the fairness of the latter. Future work avenues include the incorporation of other fairness metrics and constraints to address different aspects of fairness such as equalized odds or disparate impact. As fairness enters the optimization problem as a linear constraint, one could extend `FWC` beyond algorithmic fairness applications, focusing on other properties of the downstream learning process such as differential privacy or domain-specific constraints.

Acknowledgments. ND is grateful to Alan Mishler for insightful discussions on the idea and applicability of the approach. This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co. and its affiliates ("JP Morgan"), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

[1] O. Bachem, M. Lucic, and S. Lattanzi. One-shot coresets: The case of k-clustering. In *International conference on artificial intelligence and statistics*, pages 784–792. PMLR, 2018.

[2] A. Backurs, P. Indyk, K. Onak, B. Schieber, A. Vakilian, and T. Wagner. Scalable fair clustering. In *International Conference on Machine Learning*, pages 405–413. PMLR, 2019.

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

[4] Z. Borsos, M. Mutny, and A. Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in neural information processing systems*, 33:14879–14890, 2020.

[5] F. Calmon, D. Wei, B. Vinzamuri, K. Natesan Ramamurthy, and K. R. Varshney. Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems*, 30, 2017.

[6] T. Campbell and T. Broderick. Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*, pages 698–706. PMLR, 2018.

[7] R. Chhaya, A. Dasgupta, J. Choudhari, and S. Shit. On coresets for fair regression and individually fair clustering. In *AISTATS*, pages 9603–9625, 2022.

[8] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii. Fair clustering through fairlets. *Advances in neural information processing systems*, 30, 2017.

[9] A. Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

[10] S. Claici, A. Genevay, and J. Solomon. Wasserstein measure coresets. *arXiv preprint arXiv:1805.07412*, 2018.

[11] C. Coleman, C. Yeh, S. Mussmann, B. Mirzasoleiman, P. Bailis, P. Liang, J. Leskovec, and M. Zaharia. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019.

[12] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226, 2012.

[13] A. Fabris, S. Messina, G. Silvello, and G. A. Susto. Algorithmic fairness datasets: The story so far. *Data Mining and Knowledge Discovery*, 36(6):2074–2152, 2022.

[14] E. Fehrman, A. K. Muhammad, E. M. Mirkes, V. Egan, and A. N. Gorban. The five factor model of personality and evaluation of drug consumption risk. In *Data Science: Innovative Developments in Data Analysis and Clustering*, pages 231–242. Springer, 2017.

[15] M. Ghadiri, S. Samadi, and S. Vempala. Socially fair k-means clustering. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 438–448, 2021.

[16] S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 291–300, 2004.

[17] H. Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: https://doi.org/10.24432/C5NC77.

[18] M. Hort, Z. Chen, J. M. Zhang, F. Sarro, and M. Harman. Bias mitigation for machine learning classifiers: A comprehensive survey. *arXiv preprint arXiv:2207.07068*, 2022.

[19] L. Huang, S. Jiang, and N. Vishnoi. Coresets for clustering with fairness constraints. *Advances in Neural Information Processing Systems*, 32, 2019.

[20] C. Jung, S. Kannan, and N. Lutz. A center in your neighborhood: Fairness in facility location. *arXiv preprint arXiv:1908.09041*, 2019.

[21] F. Kamiran and T. Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems*, 33(1):1–33, 2012.

[22] L. Kantorovitch. On the translocation of masses. *Management Science*, 5(1):1–4, 1958.

[23] K. Lange. *MM optimization algorithms*. SIAM, 2016.

[24] S. Lei and D. Tao. A comprehensive survey to dataset distillation. *arXiv preprint arXiv:2301.05603*, 2023.

[25] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[26] S. Mahabadi and A. Vakilian. Individual fairness for k-clustering. In *International conference on machine learning*, pages 6586–6596. PMLR, 2020.

[27] F. E. Maranzana. On the location of supply points to minimize transportation costs. *IBM Systems Journal*, 2(2):129–135, 1963.

[28] M. Negahbani and D. Chakrabarty. Better algorithms for individually fair k-clustering. *Advances in Neural Information Processing Systems*, 34:13340–13351, 2021.

[29] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000.

[30] H.-S. Park and C.-H. Jun. A simple and fast algorithm for k-medoids clustering. *Expert systems with applications*, 36(2):3336–3341, 2009.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[32] G. Peyré, M. Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[33] M. Redmond. Communities and Crime. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C53W3X.

[34] S. Sagiroglu and D. Sinanc. Big data: A review. In *2013 international conference on collaboration technologies and systems (CTS)*, pages 42–47. IEEE, 2013.

[35] F. Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[36] M. Sloane, E. Moss, and R. Chowdhury. A silicon valley love triangle: Hiring algorithms, pseudo-science, and the quest for auditability. *Patterns*, 3(2), 2022.

[37] A. Vakilian and M. Yalciner. Improved approximation algorithms for individually fair clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 8758–8779. PMLR, 2022.

[38] C. Villani et al. *Optimal transport: Old and new*, volume 338. Springer, 2009.

[39] Z. Xiong, N. Dalmasso, A. Mishler, V. K. Potluru, T. Balch, and M. Veloso. Fairwasp: Fast and optimal fair wasserstein pre-processing. *arXiv preprint arXiv:2311.00109*, 2023.

[40] R. Yu, S. Liu, and X. Wang. Dataset distillation: A comprehensive review. *arXiv preprint arXiv:2301.07014*, 2023.

[41] A. Zhang, L. Xing, J. Zou, and J. C. Wu. Shifting machine learning for healthcare from development to deployment and from models to data. *Nature Biomedical Engineering*, 6(12):1330–1345, July 2022.