On Socially Fair Low-Rank Approximation and Column Subset Selection[*]

Zhao Song[†] Ali Vakilian[‡] David P. Woodruff[§] Samson Zhou[▽]

December 10, 2024

Abstract

Low-rank approximation and column subset selection are two fundamental and related problems that are applied across a wealth of machine learning applications. In this paper, we study the question of socially fair low-rank approximation and socially fair column subset selection, where the goal is to minimize the loss over all sub-populations of the data. We show that surprisingly, even constant-factor approximation to fair low-rank approximation requires exponential time under certain standard complexity hypotheses. On the positive side, we give an algorithm for fair low-rank approximation that, for a constant number of groups and constant-factor accuracy, runs in $2^{\text{poly}(k)}$ time rather than the naïve $n^{\text{poly}(k)}$, which is a substantial improvement when the dataset has a large number n of observations. We then show that there exist bicriteria approximation algorithms for fair low-rank approximation and fair column subset selection that run in polynomial time.

1 Introduction

Machine learning algorithms are increasingly used in technologies and decision-making processes that affect our daily lives, from high volume interactions such as online advertising, e-mail filtering, smart devices, or large language models, to more critical processes such as autonomous vehicles, healthcare diagnostics, credit scoring, and sentencing recommendations in courts of law [Cho17, KLL+18, BHJ+21]. Machine learning algorithms frequently require statistical analysis, utilizing fundamental problems from numerical linear algebra, especially low-rank approximation and column subset selection.

In the classical low-rank approximation problem, the input is a data matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ and an integral rank parameter $k > 0$, and the goal is to find the best rank k approximation to $\mathbf{A}$, i.e., to find a set of k vectors in $\mathbb{R}^d$ that span a matrix $\mathbf{B}$, which minimizes $\mathcal{L}(\mathbf{A} - \mathbf{B})$ across all rank-k matrices $\mathbf{B}$, for some loss function $\mathcal{L}$. The rank parameter k should be chosen to accurately

ℓ_1, or
ℓ_2
NORM

[*] The work was conducted in part while Ali Vakilian, David P. Woodruff, and Samson Zhou were visiting the Simons Institute for the Theory of Computing as part of the Sublinear Algorithms program.

[†] Simons Institute and UC Berkeley. E-mail: magic.linuxkde@gmail.com.

[‡] Toyota Technological Institute at Chicago. E-mail: vakilian@ttic.edu.

[§] Carnegie Mellon University and Google Research. E-mail: dwoodruf@cs.cmu.edu. Supported in part by a Simons Investigator Award and NSF CCF-2335412.

[▽] Texas A&M University. E-mail: samsonzhou@gmail.com. Supported in part by NSF CCF-2335411.

represent the complexity of the underlying model chosen to fit the data, and thus the low-rank approximation problem is often used for mathematical modeling and data compression.

Similarly, in the classic column subset selection problem, the goal is to choose k columns $\mathbf{U}$ of $\mathbf{A}$ so as to minimize $\mathcal{L}(\mathbf{A} - \mathbf{U}\mathbf{V})$ across all choices of $\mathbf{V} \in \mathbb{R}^{k \times d}$. Although low-rank approximation can reveal important latent structure among the dataset, the resulting linear combinations may not be as interpretable as simply selecting k features. The column subset selection problem is therefore a version of low-rank approximation with the restriction that the left factor U must be k columns of the data matrix. Column subset selection also tends to result in sparse models. For example, if the columns of $\mathbf{A}$ are sparse, then the columns in the left factor $\mathbf{U}$ are also sparse. Thus in some cases, column subset selection, often also called feature selection, can be more useful than general low-rank approximation.

Algorithmic fairness. Unfortunately, real-world machine learning algorithms across a wide variety of domains have recently produced a number of undesirable outcomes from the lens of generalization. For example, [BS16] noted that decision-making processes using data collected from smartphone devices reporting poor road quality could potentially underserve poorer communities with less smartphone ownership. [KMM15] observed that search queries for CEOs overwhelmingly returned images of white men, while [BG18] observed that facial recognition software exhibited different accuracy rates for white men compared with dark-skinned women.

Initial attempts to explain these issues can largely be categorized into either "biased data" or "biased algorithms", where the former might include training data that is significantly misrepresenting the true statistics of some sub-population, while the latter might sacrifice accuracy on a specific sub-population in order to achieve better global accuracy. As a result, an increasingly relevant line of active work has focused on designing *fair* algorithms. An immediate challenge is to formally define the desiderata demanded from fair algorithmic design and indeed multiple natural quantitative measures of fairness have been proposed [HPS16, Cho17, KMR17, BHJ+21]. However, [KMR17, CPF+17] showed that many of these conditions for fairness cannot be simultaneously achieved.

In this paper, we focus on *socially fair* algorithms, which seek to optimize the performance of the algorithm across all sub-populations. That is, for the purposes of low-rank approximation and column subset selection, the goal is to minimize the maximum cost across all sub-populations. For socially fair low-rank approximation, the input is a set of data matrices $\mathbf{A}^{(1)} \in \mathbb{R}^{n_1 \times d}, \ldots, \mathbf{A}^{(\ell)} \in \mathbb{R}^{n_\ell \times d}$ corresponding to ℓ groups and a rank parameter k, and the goal is to determine a set of k factors $\mathbf{U}_1, \ldots, \mathbf{U}_k \in \mathbb{R}^d$ that span matrices $\mathbf{B}^{(1)}, \ldots, \mathbf{B}^{(\ell)}$, which minimize $\max_{i \in [\ell]} \|\mathbf{A}^{(i)} - \mathbf{B}^{(i)}\|_F$. Due to the Eckart–Young–Mirsky theorem stating that the Frobenius loss is minimized when each $\mathbf{A}^{(i)}$ is projected onto the span of $\mathbf{U} = \mathbf{U}_1 \circ \ldots \circ \mathbf{U}_k$, the problem is equivalent to $\min_{\mathbf{U} \in \mathbb{R}^{k \times d}} \max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{U}^\dagger\mathbf{U} - \mathbf{A}^{(i)}\|_F$, where $\dagger$ denotes the Moore-Penrose pseudoinverse. We remark that as we can scale the columns in each group (even each individual column), our formulation captures min-max normalized cost relative to the total Frobenius norm of each group and the optimal rank-k reconstruction loss for each group.

1.1 Our Contributions and Technical Overview

In this paper, we study socially fair low-rank approximation and socially fair column subset selection.

Fair low-rank approximation. We first describe our results for socially fair low-rank approximation. We first show that under the assumption that P $\neq$ NP, fair low-rank approximation cannot be approximated within any constant factor in polynomial time.

Theorem 1.1. *Fair low-rank approximation is NP-hard to approximate within any constant factor.*

We show Theorem 1.1 by reducing to the problem of minimizing the distance of a set of n points in d-dimensional Euclidean space to all sets of k dimensional linear subspaces, which was shown by [BGK00, DTV11] to be NP-hard to approximate within any constant factor. In fact, [BGK00, DTV11] showed that a constant-factor approximation to this problem requires runtime exponential in k under a stronger assumption, the exponential time hypothesis [IP01]. We show similar results for the fair low-rank approximation problem.

Theorem 1.2. *Under the exponential time hypothesis, the fair low-rank approximation requires $2^{k^{\Omega(1)}}$ time to approximate within any constant factor.*

Together, Theorem 1.1 and Theorem 1.2 show that under standard complexity assumptions, we cannot achieve a constant-factor approximation to fair low-rank approximation using time polynomial in n and sub-exponential in $\text{poly}(k)$. We thus consider additional relaxations, such as bicriteria approximation (Theorem 1.4) or $2^{\text{poly}(k)}$ runtime (Theorem 1.3). On the positive side, we first show that for a constant number of groups and constant-factor accuracy, it suffices to use runtime $2^{\text{poly}(k)}$ rather than the naïve $n^{\text{poly}(k)}$, which is a substantial improvement when the dataset has a large number of observations, i.e., n is large.

Theorem 1.3. *Given an accuracy parameter $\varepsilon \in (0,1)$, there exists an algorithm that outputs $\widetilde{\mathbf{V}} \in \mathbb{R}^{k\times d}$ such that with probability at least $\frac{2}{3}$,*

$$\max_{i\in[\ell]} \|\mathbf{A}^{(i)}(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F \le (1+\varepsilon)\cdot \min_{\mathbf{V}\in\mathbb{R}^{k\times d}} \max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F.$$

The algorithm uses runtime $\frac{1}{\varepsilon}\,\text{poly}(n)\cdot(2\ell)^{\mathcal{O}(N)}$, for $n = \sum_{i=1}^{\ell} n_i$ and $N = \text{poly}\left(\ell, k, \frac{1}{\varepsilon}\right)$.

At a high level, the algorithm corresponding to Theorem 1.3 first finds a value α such that it is an ℓ-approximation to the optimal solution. It then repeatedly decreases α by $(1+\varepsilon)$ factors while checking if the resulting quantity is still feasible. To efficiently check whether α is feasible, we first reduce the dimension of the data using an affine embedding matrix $\mathbf{S}$, so that for all rank k matrices $\mathbf{V}$ and all $i \in [\ell]$,

$$(1-\varepsilon)\|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F^2 \le \|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V}\mathbf{S} - \mathbf{A}^{(i)}\mathbf{S}\|_F^2 \le (1+\varepsilon)\|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F^2.$$

Observe that given $\mathbf{V}$, it is known through the closed form solution to the regression problem that the rank-k minimizer of $\|\mathbf{X}^{(i)}\mathbf{V}\mathbf{S} - \mathbf{A}^{(i)}\mathbf{S}\|_F^2$ is $\mathbf{X}^{(i)} = (\mathbf{A}^{(i)}\mathbf{S})(\mathbf{V}\mathbf{S})^\dagger$. Let $\mathbf{R}^{(i)}$ be defined so that $(\mathbf{A}^{(i)}\mathbf{S})(\mathbf{V}\mathbf{S})^\dagger\mathbf{R}^{(i)}$ has orthonormal columns, so that

$$\|(\mathbf{A}^{(i)}\mathbf{S})(\mathbf{V}\mathbf{S})^\dagger\mathbf{R}^{(i)})((\mathbf{A}^{(i)}\mathbf{S})(\mathbf{V}\mathbf{S})^\dagger\mathbf{R}^{(i)})^\dagger\mathbf{A}^{(i)}\mathbf{S} - \mathbf{A}^{(i)}\mathbf{S}\|_F^2 = \min_{\mathbf{X}^{(i)}} \|\mathbf{X}^{(i)}\mathbf{V}\mathbf{S} - \mathbf{A}^{(i)}\mathbf{S}\|_F^2.$$

It follows that if α is feasible, then

$$\alpha(1+\varepsilon) \ge \|(\mathbf{A}^{(i)}\mathbf{S})(\mathbf{V}\mathbf{S})^\dagger\mathbf{R}^{(i)})((\mathbf{A}^{(i)}\mathbf{S})(\mathbf{V}\mathbf{S})^\dagger\mathbf{R}^{(i)})^\dagger\mathbf{A}^{(i)}\mathbf{S} - \mathbf{A}^{(i)}\mathbf{S}\|_F^2.$$

Unfortunately, $\mathbf{V}$ is not given, so the above approach will not quite work. Instead, we use a polynomial system solver to check whether there exists such a $\mathbf{V}$ by writing $\mathbf{Y} = \mathbf{VS}$ and its pseudoinverse $\mathbf{W} = (\mathbf{VS})^\dagger$ and we check whether there exists a satisfying assignment to the above inequality, given the constraints (1) $\mathbf{YWY} = \mathbf{Y}$, (2) $\mathbf{WYW} = \mathbf{W}$, and (3) $\mathbf{A}^{(i)}\mathbf{SWR}^{(i)}$ has orthonormal columns.

We remark that because $\mathbf{V} \in \mathbb{R}^{k\times d}$, then we cannot naïvely implement the polynomial system, because it would require kd variables and thus use $2^{\Omega(dk)}$ runtime. Instead, we only manipulate $\mathbf{VS}$, which has dimension $k \times m$ for $m = \mathcal{O}\left(\frac{k^2}{\varepsilon^2}\log \ell\right)$, allowing the polynomial system solver to achieve $2^{\text{poly}(mk)}$ runtime.

Next, we show that there exists a bicriteria approximation algorithm for fair low-rank approximation that uses polynomial runtime.

Theorem 1.4. *Given a trade-off parameter $c \in (0,1)$, there exists an algorithm that outputs $\widetilde{\mathbf{V}} \in \mathbb{R}^{t\times d}$ for $t = \mathcal{O}\left(k(\log\log k)(\log^2 d)\right)$ such that with probability at least $\frac{2}{3}$,*

$$\max_{i\in[\ell]} \|\mathbf{A}^{(i)}(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F \le \ell^c \cdot 2^{1/c} \cdot \mathcal{O}\left(k(\log\log k)(\log d)\right) \cdot \min_{\mathbf{V}\in\mathbb{R}^{k\times d}} \max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F.$$

The algorithm uses runtime polynomial in n and d.

The algorithm for Theorem 1.4 substantially differs from that of Theorem 1.3. For one, we can no longer use a polynomial system solver, because it would be infeasible to achieve polynomial runtime. Instead, we observe that for sufficiently large p, we have $\max \|\mathbf{x}\|_\infty = (1 \pm \varepsilon)\|\mathbf{x}\|_p$ and thus focus on optimizing

$$\min_{\mathbf{V}\in\mathbb{R}^{k\times d}} \left(\sum_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F^p\right)^{1/p}.$$

However, the terms $\|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F^p$ are difficult to handle, so we apply Dvoretzky's Theorem, i.e., Theorem 2.11, to generate matrices $\mathbf{G}$ and $\mathbf{H}$ so that

$$(1-\varepsilon)\|\mathbf{GMH}\|_p \le \|\mathbf{M}\|_F \le (1+\varepsilon)\|\mathbf{GMH}\|_p.$$

Although low-rank approximation with L_p loss cannot be well-approximated in polynomial time, we recall that there exists a matrix $\mathbf{S}$ that samples a "small" number of columns of $\mathbf{A}$ to provide a coarse bicriteria approximation to L_p low-rank approximation [WY23a]. However, we require a solution with dimension d and thus we seek to solve the regression problem $\min_{\mathbf{X}} \|\mathbf{GAHSX}-\mathbf{GAH}\|_p$. Thus, we consider a Lewis weight sampling matrix $\mathbf{T}$ such that

$$\frac{1}{2}\|\mathbf{TGAHSX} - \mathbf{TGAH}\|_p \le \|\mathbf{GAHSX} - \mathbf{GAH}\|_p \le 2\|\mathbf{TGAHSX} - \mathbf{TGAH}\|_p.$$

and again note that $(\mathbf{TGAHS})^\dagger\mathbf{TGAH}$ is the closed-form solution to the minimization problem $\min_{\mathbf{X}} \|\mathbf{TGAHSX} - \mathbf{TGAH}\|_F$, which only provides a small distortion to the L_p regression problem, since $\mathbf{TGAH}$ has a small number of rows due to the dimensionality reduction. We then observe that by Dvoretzky's Theorem, $(\mathbf{TGAHS})^\dagger\mathbf{TGA}$ is a "good" approximate solution to the original fair low-rank approximation problem. Given $\delta \in (0,1)$, the success probabilities for both Theorem 1.3 and Theorem 1.4 can be boosted to arbitrary $1-\delta$ by taking the minimum of $\mathcal{O}\left(\log\frac{1}{\delta}\right)$ independent instances of the algorithm, at the cost of increasing the runtime by the same factor.

Fair column subset selection. We next describe our results for fair column subset selection. We give a bicriteria approximation algorithm for fair column subset selection that uses polynomial runtime.

Theorem 1.5. *Given input matrices $\mathbf{A}^{(i)} \in \mathbb{R}^{n_i \times d}$ with $n = \sum n_i$, there exists an algorithm that selects a set S of $k' = \mathcal{O}(k \log k)$ columns such that with probability at least $\frac{2}{3}$, S is a $\mathcal{O}(k(\log\log k)(\log d))$-approximation to the fair column subset selection problem. The algorithm uses runtime polynomial in n and d.*

The immediate challenge in adapting the previous approach for fair low-rank approximation to fair column subset selection is that we required the Gaussian matrices $\mathbf{G}, \mathbf{H}$ to embed the awkward maximum of Frobenius losses $\min \max_{i\in[\ell]} \|\cdot\|_F$ into the more manageable L_p loss $\min \|\cdot\|_p$ through $\mathbf{GAH}$. However, selecting columns of $\mathbf{GAH}$ does not correspond to selecting columns of the input matrices $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(\ell)}$.

Instead, we view the bicriteria solution $\widetilde{\mathbf{V}}$ from fair low-rank approximation as a good starting point for the right factor for fair low-rank approximation. Thus we consider the multi-response regression problem $\max_{i\in[\ell]} \min_{\mathbf{B}^{(i)}} \|\mathbf{B}^{(i)}\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F$. We then argue through Dvoretzky's theorem that a leverage score sampling matrix $\mathbf{S}$ that samples $\mathcal{O}(k \log k)$ columns of $\widetilde{\mathbf{V}}$ will provide a good approximation to the column subset selection problem. We defer the formal exposition to Section 3.

Empirical evaluations. Finally, in Section 4, we perform a number of experimental results on socially fair low-rank approximation, comparing the performance of the socially fair low-rank objective values associated with the outputs of the bicriteria fair low-rank approximation algorithm and the standard (non-fair) low-rank approximation that outputs the top k right singular vectors of the singular value decomposition.

Our experiments are on the Default of Credit Card Clients dataset [YL09], which is a common human-centric data used for benchmarks on fairness, e.g., see [STM+18]. We perform empirical evaluations comparing the objective value and the runtime of our bicriteria algorithm with the aforementioned baseline, across various subsample sizes and rank parameters. Our results demonstrate that our bicriteria algorithm can perform better than the standard low-rank approximation algorithm across various parameter settings, even when the bicriteria algorithm is not allowed a larger rank than the baseline. Moreover, we show that our algorithm is quite efficient and in fact, the final step of extracting the low-rank factors is faster than the singular value decomposition baseline due to a smaller input matrix. All in all, our empirical evaluations indicate that our bicriteria algorithm can perform well in practice, thereby reinforcing the theoretical guarantees of the algorithm.

1.2 Related Work

Initial insight into *socially fair data summarization* methods were presented by [STM+18], where the concept of *fair PCA* was explored. This study introduced the fairness metric of average reconstruction loss, expressed by the loss function $\text{loss}(\mathbf{A}, \mathbf{B}) := \|\mathbf{A} - \mathbf{B}\|_F^2 - \|\mathbf{A} - \mathbf{A}_k\|_F^2$, aiming to identify a k-dimensional subspace that minimizes the loss across the groups, with $\mathbf{A}_k$ representing the best rank-k approximation of $\mathbf{A}$. Their proposed approach, in a two-group scenario, identifies a fair PCA of up to $k+1$ dimensions that is not worse than the optimal fair PCA with k dimensions. When extended to ℓ groups, this method requires an additional $k + \ell - 1$ dimensions. Subsequently, [TSS+19] explored fair PCA from a distinct objective perspective, seeking a projection matrix $\mathbf{P}$ optimizing $\min_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{P}\|_F^2$. A pivotal difference between these works and ours is our focus on the reconstruction error objective, a widely accepted objective for regression and low-rank approximation tasks. Alternatively, [OA19, LKO+22, KDRZ23] explored a different formulation of fair PCA. The main objective is to ensure that data representations are not influenced by demographic attributes. In particular, when a classifier is exposed only to the projection of points onto the k-dimensional subspace, it should be unable to predict the demographic attributes.

Recently, [MOT24] studied a fair column subset selection objective similar to ours, focusing on the setting with two groups (i.e., $\ell = 2$). They established the problem's NP-hardness and introduced a polynomial-time solution that offers relative-error guarantees while selecting a column subset of size $\mathcal{O}(k)$.

For fair regression, initial research focused on designing models that offer similar treatment to instances with comparable observed results by incorporating *fairness regularizers* [BHJ+17]. However, in [ADW19], the authors studied a fairness notion closer to our optimization problem, termed as "bounded group loss". In their work, the aim is to cap each group's loss within a specific limit while also optimizing the cumulative loss. Notably, their approach diverged from ours, with a focus on the sample complexity and the problem's generalization error bounds.

[AAK+22] studied a similar socially fair regression problem under the name min-max regression. In their setting, the goal is to minimize the maximum loss over a mixture distribution, given samples from the mixture; our fair regression setting can be reduced to theirs. [AAK+22] observed that a maximum of norms is a convex function and can therefore be solved using projected stochastic gradient descent.

The term "socially fair" was first introduced in the context of clustering, aiming to optimize clustering costs across predefined group sets [GSV21, ABV21]. In subsequent studies, tight approximation algorithms [MV21, CMV22], FPT approaches [GJ23], and bicriteria approximation algorithms [GSV22] for socially fair clustering have been presented.

1.3 Preliminaries

We use the notation $[n]$ to represent the set $\{1,\ldots,n\}$ for an integer $n \ge 1$. We use the notation $\text{poly}(n)$ to represent a fixed polynomial in n and we use the notation $\text{polylog}(n)$ to represent $\text{poly}(\log n)$. We use $\text{poly}(n)$ to denote a fixed polynomial in n and $\text{polylog}(n)$ to denote $\text{poly}(\log n)$. We say an event holds with high probability if it holds with probability $1 - \frac{1}{\text{poly}(n)}$.

We generally use bold-font variables to represent vectors and matrices, whereas we use default-font variables to represent scalars. For a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$, we use $\mathbf{A}_i$ to represent the i-th row of $\mathbf{A}$ and $\mathbf{A}^{(j)}$ to represent the j-th column of $\mathbf{A}$. We use $A_{i,j}$ to represent the entry in the i-th row and j-th column of $\mathbf{A}$. For $p \ge 1$, we use

$$\|\mathbf{A}\|_p = \left(\sum_{i\in[n]}\sum_{j\in[d]} A_{i,j}^p\right)^{1/p}$$

to represent the entrywise L_p norm of $\mathbf{A}$ and we use

$$\|\mathbf{A}\|_F = \left(\sum_{i\in[n]}\sum_{j\in[d]} A_{i,j}^2\right)^{1/2}$$

to represent the Frobenius norm of $\mathbf{A}$, which is simply the entrywise L_2 norm of $\mathbf{A}$. We use define the $L_{p,q}$ of $\mathbf{A}$ as the L_p norm of the vector consisting of the L_q norms of each row of $\mathbf{A}$, so that

$$\|\mathbf{A}\|_{p,q} = \left(\sum_{i\in[n]}\left(\sum_{j\in[d]}(A_{i,j})^q\right)^{p/q}\right)^{1/p}.$$

Similarly, we use $\|\mathbf{A}\|_{(p,q)}$ to denote the L_p norm of the vector consisting of the L_q norms of each column of $\mathbf{A}$. Equivalently, we have $\|\mathbf{A}\|_{(p,q)} = \|\mathbf{A}^\top\|_{p,q}$, so that

$$\|\mathbf{A}\|_{(p,q)} = \left(\sum_{j\in[d]}\left(\sum_{i\in[n]}(A_{i,j})^q\right)^{p/q}\right)^{1/p}.$$

We use $\circ$ to represent vertical stacking of matrices, so that

$$\mathbf{A}^{(1)}\circ\ldots\circ\mathbf{A}^{(m)} = \begin{bmatrix}\mathbf{A}^{(1)}\\ \vdots \\ \mathbf{A}^{(m)}\end{bmatrix}.$$

For a matrix $\mathbf{A}\in\mathbb{R}^{n\times d}$ of rank k, its singular decomposition is $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$, where $\mathbf{U}\in\mathbb{R}^{n\times k}$, $\mathbf{\Sigma}\in\mathbb{R}^{k\times k}$, $\mathbf{V}\in\mathbb{R}^{k\times d}$, so that the columns of $\mathbf{U}$ are orthonormal and the rows of $\mathbf{V}$ are orthonormal. The columns of $\mathbf{U}$ are called the left singular vectors of $\mathbf{A}$ and the rows of $\mathbf{V}$ are called the right singular vectors of $\mathbf{A}$.

1.3.1 Regression and Low-Rank Approximation

In this section, we briefly describe some common techniques used to handle both regression and low-rank approximation, thus presenting multiple unified approaches for both problems. Thus in light of the abundance of techniques that can be used to handle both problems, it is somewhat surprising that socially fair regression and socially fair low-rank approximation exhibit vastly different complexities.

Closed form solutions. Given the regression problem $\min_{\mathbf{x}\in\mathbb{R}^d}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2$ for an input matrix $\mathbf{A}\in\mathbb{R}^{n\times d}$ and a label vector $\mathbf{b}\in\mathbb{R}^n$, the closed form solution for the minimizer is $\mathbf{A}^\dagger\mathbf{b} = \text{argmin}_{\mathbf{x}\in\mathbb{R}^d}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2$, where $\mathbf{A}^\dagger$ is the Moore-Penrose pseudoinverse of $\mathbf{A}$. Specifically, for a matrix $\mathbf{A}\in\mathbb{R}^{n\times d}$ written in its singular value decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$, we have $\mathbf{A}^\dagger = \mathbf{V}^\top\mathbf{\Sigma}^{-1}\mathbf{U}^\top$. In particular, for if $\mathbf{A}$ has linearly independent rows, then $\mathbf{A}^\dagger = \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}$. On the other hand, if $\mathbf{A}$ has linearly independent columns, then $\mathbf{A}^\dagger = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$.

Similarly, given an input matrix $\mathbf{A}$ and a rank parameter $k > 0$, there exists a closed form solution for the minimizer $\text{argmin}_{\mathbf{V}\in\mathbb{R}^{k\times d}}\|\mathbf{A}-\mathbf{A}\mathbf{V}^\dagger\mathbf{V}\|_F^2$. Specifically, by the Eckart-Young-Mirsky theorem [EY36], the minimizer is the top k right singular vectors of $\mathbf{A}$.

Dimensionality reduction. We next recall a unified set of dimensionality reduction techniques for both linear regression and low-rank approximation. We consider the "sketch-and-solve" paradigm, so that for both problems, we first acquire a low-dimension representation of the problem, and find the optimal solution in the low dimension using the above closed-form solutions. For "good" designs of the low-dimension representations, the low-dimension solution will also be near-optimal for the original problem.

We first observe that oblivious linear sketches serve as a common dimensionality reduction for both linear regression and low-rank approximation. For example, it is known [Woo14] that there exists a family of Gaussian random matrices $\mathcal{G}_1$ from which $\mathbf{S} \sim \mathcal{G}_1$ satisfies with high probability,

$$(1-\varepsilon)\|\mathbf{SAx}-\mathbf{Sb}\|_2 \le \|\mathbf{Ax}-\mathbf{b}\|_2 \le (1+\varepsilon)\|\mathbf{SAx}-\mathbf{Sb}\|_2,$$

simultaneously for all $\mathbf{x} \in \mathbb{R}^d$. Similarly, there exists [Woo14] a family of Gaussian random matrices $\mathcal{G}_2$ from which $\mathbf{S} \sim \mathcal{G}_1$ satisfies with high probability, that the row space of $\mathbf{SA}$ contains a $(1+\varepsilon)$-approximation of the optimal low-rank approximation to $\mathbf{A}$.

Alternatively, we can achieve dimensionality reduction for both linear regression and low-rank approximation by sampling a small subset of the input in related ways for both problems. For linear regression, we can generate a random matrix $\mathbf{S}$ by sampling rows of $\begin{bmatrix}\mathbf{A} & \mathbf{b}\end{bmatrix}$ by their leverage scores [DMM06a, DMM06b, Mag10, Woo14]. In this manner, we again achieve a matrix $\mathbf{S}$ such that with high probability,

$$(1-\varepsilon)\|\mathbf{SAx}-\mathbf{Sb}\|_2 \le \|\mathbf{Ax}-\mathbf{b}\|_2 \le (1+\varepsilon)\|\mathbf{SAx}-\mathbf{Sb}\|_2,$$

simultaneously for all $\mathbf{x} \in \mathbb{R}^d$. For low-rank approximation, we can generate a random matrix $\mathbf{S}$ by sampling rows of $\mathbf{A}$ with the related ridge-leverage scores [CMM17]. Then with high probability, we have for all $\mathbf{V} \in \mathbb{R}^{k\times d}$,

$$(1-\varepsilon)\|\mathbf{SA}-\mathbf{SAV}^\dagger\mathbf{V}\|_F^2 \le \|\mathbf{A}-\mathbf{AV}^\dagger\mathbf{V}\|_F^2 \le (1+\varepsilon)\|\mathbf{SA}-\mathbf{SAV}^\dagger\mathbf{V}\|_F^2.$$

1.3.2 Leverage Score Sampling and Lewis Weights

For further runtime improvements for sparse inputs, we recall the following notion of subspace embeddings.

Definition 1.6 (Subspace embedding)**.** *Given an input matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ and an accuracy parameter $\varepsilon \in (0,1)$, a subspace embedding for $\mathbf{A}$ is a matrix $\mathbf{M} \in \mathbb{R}^{m\times d}$ such that for all $\mathbf{x} \in \mathbb{R}^d$, we have*

$$(1-\varepsilon)\|\mathbf{Ax}\|_p \le \|\mathbf{Mx}\|_p \le (1+\varepsilon)\|\mathbf{Ax}\|_p.$$

We first provide preliminaries on leverage score sampling.

Definition 1.7. *Given a matrix $\mathbf{M} \in \mathbb{R}^{n\times d}$, we define the leverage score σ_i of each row $\mathbf{m}_i$ with $i \in [n]$ by $\mathbf{m}_i(\mathbf{M}^\dagger\mathbf{M})^{-1}\mathbf{m}_i^\top$. Equivalently, for the singular value decomposition $\mathbf{M} = \mathbf{U\Sigma V}$, the leverage score of row $\mathbf{m}_i$ is also the squared row norm of $\mathbf{u}_i$.*

We recall that it can be shown the sum of the leverage scores for an input matrix $\mathbf{M} \in \mathbb{R}^{n\times d}$ is upper bounded by d and moreover, given the leverage scores of $\mathbf{M}$, it suffices to sample only $\mathcal{O}(d\log d)$ rows of $\mathbf{M}$ to achieve a constant factor subspace embedding of $\mathbf{M}$. Specifically, it is known that the sum of the leverage scores of the rows of a matrix can be bounded by the rank of the matrix.

Theorem 1.8 (Generalization of Foster's Theorem, [Fos53])**.** *Given a matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$, the sum of its leverage scores is* $\text{rank}(\mathbf{M})$.

By sampling rows proportional to their leverage scores, we can obtain a subspace embedding as follows:

Theorem 1.9 (Leverage score sampling)**.** *[DMM06a, DMM06b, Mag10, Woo14] Given a matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$, let σ_i be the leverage score of the i-th row of $\mathbf{M}$. Suppose $p_i = \min(1, \sigma_i \log n)$ for each $i \in [n]$ and let $\mathbf{S}$ be a random diagonal matrix so that the i-th diagonal entry of $\mathbf{S}$ is $\frac{1}{\sqrt{p_i}}$ with probability p_i and 0 with probability $1 - p_i$. Then for all vectors $\mathbf{v} \in \mathbb{R}^n$,*

$$\mathbb{E}\left[\|\mathbf{S}\mathbf{v}\|_2^2\right] = \|\mathbf{v}\|_2^2$$

and with probability at least 0.99, for all vectors $\mathbf{x} \in \mathbb{R}^d$

$$\frac{99}{100}\|\mathbf{M}\mathbf{x}\|_2 \le \|\mathbf{S}\mathbf{M}\mathbf{x}\|_2 \le \frac{101}{100}\|\mathbf{M}\mathbf{x}\|_2.$$

Moreover, $\mathbf{S}$ has at most $\mathcal{O}(d \log d)$ nonzero entries with high probability.

Because the leverage scores of $\mathbf{M}$ can be computed directly from the singular value decomposition of $\mathbf{M}$, which can be computed in $\mathcal{O}(nd^\omega + dn^\omega)$ time where ω is the exponent of matrix multiplication, then the leverage scores of $\mathbf{M}$ can be computed in polynomial time.

Finally, we recall that to provide a constant factor approximation to L_p regression, it suffices to compute a constant factor subspace embedding, e.g., through leverage score sampling. The proof is through the triangle inequality and is well-known among the active sampling literature [CP19, PPP21, MMWY22, MMM$^+$22, MMM$^+$23], e.g., a generalization of Lemma 2.1 in [MMM$^+$22].

Lemma 1.10. *Given a matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$, let $\mathbf{S}$ be a matrix such that for all $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{v} \in \mathbb{R}^n$,*

$$\frac{11}{12}\|\mathbf{M}\mathbf{x}\|_2 \le \|\mathbf{S}\mathbf{M}\mathbf{x}\|_2 \le \frac{13}{12}\|\mathbf{M}\mathbf{x}\|_2, \ \mathbb{E}\left[\|\mathbf{S}\mathbf{v}\|_2^2\right] = \|\mathbf{v}\|_2^2.$$

For a fixed $\mathbf{B} \in \mathbb{R}^{n \times m}$ where $\mathbf{B} = \mathbf{b}_1 \circ \ldots \circ \mathbf{b}_m$ with $\mathbf{b}_i \in \mathbb{R}^n$ for $i \in [m]$, let $\widetilde{\mathbf{x}_i} = (\mathbf{S}\mathbf{M})^\dagger(\mathbf{S}\mathbf{b}_i)$. Let $\widetilde{\mathbf{X}} = \widetilde{\mathbf{x}_1} \circ \ldots \circ \widetilde{\mathbf{x}_m}$. Then with probability at least 0.97, $\|\mathbf{M}\widetilde{\mathbf{X}} - \mathbf{B}\|_2 \le 99 \min_{\mathbf{X}} \|\mathbf{M}\mathbf{X} - \mathbf{B}\|_2$.

Proof. Let $\mathbf{X}^* = \text{argmin}_{\mathbf{X}} \|\mathbf{M}\mathbf{X} - \mathbf{B}\|_2$ and $\mathsf{OPT} = \|\mathbf{M}\mathbf{X}^* - \mathbf{B}\|_2$. By triangle inequality,

$$\|\mathbf{S}\mathbf{M}\mathbf{X} - \mathbf{S}\mathbf{B}\|_2 \ge \|\mathbf{S}\mathbf{M}(\mathbf{X} - \mathbf{X}^*)\|_2 - \|\mathbf{S}\mathbf{M}\mathbf{X}^* - \mathbf{S}\mathbf{B}\|_2.$$

We have

$$\frac{99}{100}\|\mathbf{M}\mathbf{X}\|_2 \le \|\mathbf{S}\mathbf{M}\mathbf{X}\|_2 \le \frac{101}{100}\|\mathbf{M}\mathbf{X}\|_2$$

for all $\mathbf{X} \in \mathbb{R}^{n \times m}$. Thus,

$$\|\mathbf{S}\mathbf{M}\mathbf{X} - \mathbf{S}\mathbf{B}\|_2 \ge \frac{99}{100}\|\mathbf{M}(\mathbf{X} - \mathbf{X}^*)\|_2 - \|\mathbf{S}\mathbf{M}\mathbf{X}^* - \mathbf{S}\mathbf{B}\|_2.$$

By triangle inequality,

$$\|\mathbf{S}\mathbf{M}\mathbf{X} - \mathbf{S}\mathbf{B}\|_2 \ge \frac{99}{100}\left(\|\mathbf{M}\mathbf{X} - \mathbf{B}\|_2 - \|\mathbf{M}\mathbf{X}^* - \mathbf{B}\|_2\right) - \|\mathbf{S}\mathbf{M}\mathbf{X}^* - \mathbf{S}\mathbf{B}\|_2.$$

Since $\mathbb{E}\left[\|\mathbf{S}\mathbf{v}\|_2^2\right] = \|\mathbf{v}\|_2^2$ for all $\mathbf{x} \in \mathbb{R}^d$, then by concavity and Markov's inequality, we have that

$$\mathbf{Pr}\left[\|\mathbf{S}\mathbf{M}\mathbf{X}^* - \mathbf{S}\mathbf{B}\|_2 \geq 49\|\mathbf{M}\mathbf{X}^* - \mathbf{B}\|_2\right] \leq \frac{1}{49}.$$

Thus with probability at least 0.97,

$$\|\mathbf{S}\mathbf{M}\mathbf{X} - \mathbf{S}\mathbf{B}\|_2 \geq \frac{99}{100}\left(\|\mathbf{M}\mathbf{X} - \mathbf{B}\|_2 - \|\mathbf{M}\mathbf{X}^* - \mathbf{B}\|_2\right) - 49\|\mathbf{M}\mathbf{X}^* - \mathbf{B}\|_2.$$

Now since we have $\|\mathbf{S}\mathbf{M}\mathbf{X}^* - \mathbf{S}\mathbf{B}\|_2 \leq 49\|\mathbf{M}\mathbf{X}^* - \mathbf{B}\|_2$ and $\|\mathbf{S}\mathbf{M}\widetilde{\mathbf{X}} - \mathbf{S}\mathbf{B}\|_2 \leq \|\mathbf{S}\mathbf{M}\mathbf{X}^* - \mathbf{S}\mathbf{B}\|_2$, then

$$49\|\mathbf{M}\mathbf{X}^* - \mathbf{B}\|_2 \geq \frac{99}{100}\left(\|\mathbf{M}\widetilde{\mathbf{X}} - \mathbf{B}\|_2 - \|\mathbf{M}\mathbf{X}^* - \mathbf{B}\|_2\right) - 49\|\mathbf{M}\mathbf{X}^* - \mathbf{B}\|_2,$$

so that

$$\|\mathbf{M}\widetilde{\mathbf{X}} - \mathbf{B}\|_2 \leq 99\|\mathbf{M}\mathbf{X}^* - \mathbf{B}\|_2,$$

as desired. □

We also recall the following construction to use Lewis weights to achieve an L_p subspace embedding.

Theorem 1.11 ([CP15]). *Let $\varepsilon \in (0,1)$ and $p \geq 2$. Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ and $s = \mathcal{O}\left(d^{p/2}\log d\right)$. Then there exists a polynomial-time algorithm that outputs a matrix $\mathbf{S} \in \mathbb{R}^{s\times n}$ that samples and reweights s rows of $\mathbf{A}$, such that with probability at least 0.99, simultaneously for all $\mathbf{x} \in \mathbb{R}^d$, $(1-\varepsilon)\|\mathbf{A}\mathbf{x}\|_p^p \leq \|\mathbf{S}\mathbf{A}\mathbf{x}\|_p^p \leq (1+\varepsilon)\|\mathbf{A}\mathbf{x}\|_p^p$.*

Further constructions of subspace embeddings have been well-studied. We utilize the following constructions:

Definition 1.12 ([CP15, Lee16, JLS22, WY23b]). *Given an input matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ and an accuracy parameter $\varepsilon \in (0,1)$, let $m = \mathcal{O}\left(\frac{d^{p/2}}{\varepsilon^2}\cdot\log^2 d\cdot\log n\right)$ for $p > 2$ and $m = \tilde{\mathcal{O}}\left(\frac{d}{\varepsilon^2}\right)$ for $p \leq 2$. There exists an algorithm outputs a matrix $\mathbf{M} \in \mathbb{R}^{m\times d}$ such that with probability 0.99, simultaneously for all $\mathbf{x} \in \mathbb{R}^d$, we have*

$$(1-\varepsilon)\|\mathbf{A}\mathbf{x}\|_p \leq \|\mathbf{M}\mathbf{x}\|_p \leq (1+\varepsilon)\|\mathbf{A}\mathbf{x}\|_p.$$

The algorithm uses $\tilde{\mathcal{O}}\left(\mathrm{nnz}(\mathbf{A})\right) + \mathrm{poly}\left(d, \frac{1}{\varepsilon}\right)$ runtime.

2 Socially Fair Low-Rank Approximation

In this section, we consider algorithms and hardness for socially fair low-rank approximation. Let $n_1, \ldots, n_\ell$ be positive integers and for each $i \in [\ell]$, let $\mathbf{A}^{(i)} \in \mathbb{R}^{n_i\times d}$. Then for a norm $\|\cdot\|$, we define the fair low-rank approximation problem to be $\min_{\mathbf{V}\in\mathbb{R}^{k\times d}}\max_{i\in[\ell]}\|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|$.

2.1 Lower Bounds

In this section, we show that it is NP-hard to approximate fair low-rank approximation within any constant factor in polynomial time and moreover, under the exponential time hypothesis, it requires exponential time to achieve a constant factor approximation.

Given points $\mathbf{v}^{(1)},\ldots,\mathbf{v}^{(n)}\in\mathbb{R}^d$, their outer $(d-k)$-radius is defined as the minimum, over all k-dimensional linear subspaces, of the maximum Euclidean distance of these points to the subspace. We define this problem as $\mathsf{Subspace}(k,\infty)$. It is known that it is NP-hard to approximate the $\mathsf{Subspace}(n-1,\infty)$ problem within any constant factor:

Theorem 2.1 ([BGK00, DTV11])**.** *The* $\mathsf{Subspace}(n-1,\infty)$ *problem is NP-hard to approximate within any constant factor.*

Utilizing the NP-hardness of approximation of the $\mathsf{Subspace}(n-1,\infty)$ problem, we show the NP-hardness of approximation of fair low-rank approximation.

Theorem 1.1. *Fair low-rank approximation is NP-hard to approximate within any constant factor.*

Proof. Given an instance $\mathbf{v}^{(1)},\ldots,\mathbf{v}^{(n)}\in\mathbb{R}^d$ of $\mathsf{Subspace}(n-1,\infty)$ with $n<d$, we set $\ell=k=n-1$ and $\mathbf{A}^{(i)}=\mathbf{v}^{(i)}$ for all $i\in[n]$. Then for a k-dimensional linear subspace $\mathbf{V}\in\mathbb{R}^{k\times d}$, we have that $\|\mathbf{A}^{(i)}\mathbf{V}^\top\mathbf{V}-\mathbf{A}^{(i)}\|_F^2$ is the distance from $\mathbf{v}^{(i)}$ to the subspace. Hence, $\max_{i\in[\ell]}\|\mathbf{A}^{(i)}\mathbf{V}^\top\mathbf{V}-\mathbf{A}^{(i)}\|_F^2$ is the maximum Euclidean distance of these points to the subspace and so the fair low-rank approximation problem is exactly $\mathsf{Subspace}(n-1,\infty)$. By Theorem 2.1, the $\mathsf{Subspace}(n-1,\infty)$ problem is NP-hard to approximate within any constant factor. Thus, fair low-rank approximation is NP-hard to approximate within any constant factor. □

We next introduce a standard complexity assumption beyond NP-hardness. Recall that in the 3-SAT problem, the input is a Boolean satisfiability problem written in conjunctive normal form, consisting of n clauses, each with 3 literals, either a variable or the negation of a variable. The goal is to determine whether there exists a Boolean assignment to the variables to satisfy the formula.

Hypothesis 2.2 (Exponential time hypothesis [IP01])**.** *The 3-SAT problem requires* $2^{\Omega(n)}$ *runtime.*

Observe that while NP-hardness simply conjectures that the 3-SAT problem cannot be solved in polynomial time, the exponential time hypothesis conjectures that the 3-SAT problem requires *exponential* time.

We remark that in the context of Theorem 2.1, [BGK00] showed the hardness of approximation of $\mathsf{Subspace}(n-1,\infty)$ through a reduction from the Max-Not-All-Equal-3-SAT problem, whose NP-hardness itself is shown through a reduction from 3-SAT. Thus under the exponential time hypothesis, Max-Not-All-Equal-3-SAT problem requires $2^{\Omega(n)}$ to solve. Then it follows that:

Theorem 2.3 ([BGK00, DTV11])**.** *Assuming the exponential time hypothesis, then the* $\mathsf{Subspace}(n-1,\infty)$ *problem requires* $2^{n^{\Omega(1)}}$ *time to approximate within any constant factor.*

It follows that under the exponential time hypothesis, any constant-factor approximation to socially fair low-rank approximation requires exponential time.

Theorem 1.2. *Under the exponential time hypothesis, the fair low-rank approximation requires* $2^{k^{\Omega(1)}}$ *time to approximate within any constant factor.*

Proof. Given an instance $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(k)} \in \mathbb{R}^d$ of $\mathsf{Subspace}(k-1, \infty)$ with $k < d$, we set $\ell = k - 1$ and $\mathbf{A}^{(i)} = \mathbf{v}^{(i)}$ for all $i \in [k]$. Then for a $(k-1)$-dimensional linear subspace $\mathbf{V} \in \mathbb{R}^{(k-1) \times d}$, we have that $\|\mathbf{A}^{(i)}\mathbf{V}^\top\mathbf{V} - \mathbf{A}^{(i)}\|_F^2$ is the distance from $\mathbf{v}^{(i)}$ to the subspace. Hence, $\max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^\top\mathbf{V} - \mathbf{A}^{(i)}\|_F^2$ is the maximum Euclidean distance of these points to the subspace and so the fair low-rank approximation problem is exactly $\mathsf{Subspace}(k-1, \infty)$. By Theorem 2.3, the $\mathsf{Subspace}(k-1, \infty)$ problem requires $2^{k^{\Omega(1)}}$ time to approximate within any constant factor. Thus, fair low-rank approximation requires $2^{k^{\Omega(1)}}$ time to approximate within any constant factor. □

2.2 $(1+\varepsilon)$-Approximation Algorithm for Fair Low-Rank Approximation

We first give a $(1+\varepsilon)$-approximation algorithm for fair low-rank approximation that uses runtime $\frac{1}{\varepsilon}\,\mathrm{poly}(n) \cdot (2\ell)^{\mathcal{O}(N)}$, for $n = \sum_{i=1}^{\ell} n_i$ and $N = \mathrm{poly}\left(\ell, k, \frac{1}{\varepsilon}\right)$.

The algorithm first finds a value α that is an ℓ-approximation to the optimal solution, i.e., $\min_{\mathbf{V} \in \mathbb{R}^{k \times d}} \max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F$ is at most $\alpha \le \ell \cdot \min_{\mathbf{V} \in \mathbb{R}^{k \times d}} \max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F$. We then repeatedly decrease α by $(1+\varepsilon)$ while checking if the resulting quantity is still achievable. To efficiently check if α is achievable, we first apply dimensionality reduction to each of the matrices by right-multiplying by an affine embedding matrix $\mathbf{S}$, so that

$$(1-\varepsilon)\|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F^2 \le \|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V}\mathbf{S} - \mathbf{A}^{(i)}\mathbf{S}\|_F^2 \le (1+\varepsilon)\|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F^2,$$

for all rank k matrices $\mathbf{V}$ and all $i \in [\ell]$.

Now if we knew $\mathbf{V}$, then for each $i \in [\ell]$, we can find $\mathbf{X}^{(i)}$ minimizing $\|\mathbf{X}^{(i)}\mathbf{V}\mathbf{S} - \mathbf{A}^{(i)}\mathbf{S}\|_F^2$ and the resulting quantity will approximate $\|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F^2$. In fact, we know that the minimizer is $(\mathbf{A}^{(i)}\mathbf{S})(\mathbf{V}\mathbf{S})^\dagger$ through the closed form solution to the regression problem. Let $\mathbf{R}^{(i)}$ be defined so that $(\mathbf{A}^{(i)}\mathbf{S})(\mathbf{V}\mathbf{S})^\dagger\mathbf{R}^{(i)}$ has orthonormal columns, so that

$$\|(\mathbf{A}^{(i)}\mathbf{S})(\mathbf{V}\mathbf{S})^\dagger\mathbf{R}^{(i)})((\mathbf{A}^{(i)}\mathbf{S})(\mathbf{V}\mathbf{S})^\dagger\mathbf{R}^{(i)})^\dagger\mathbf{A}^{(i)}\mathbf{S} - \mathbf{A}^{(i)}\mathbf{S}\|_F^2 = \min_{\mathbf{X}^{(i)}} \|\mathbf{X}^{(i)}\mathbf{V}\mathbf{S} - \mathbf{A}^{(i)}\mathbf{S}\|_F^2,$$

and so we require that if α is feasible, then $\alpha \ge \|(\mathbf{A}^{(i)}\mathbf{S})(\mathbf{V}\mathbf{S})^\dagger\mathbf{R}^{(i)})((\mathbf{A}^{(i)}\mathbf{S})(\mathbf{V}\mathbf{S})^\dagger\mathbf{R}^{(i)})^\dagger\mathbf{A}^{(i)}\mathbf{S} - \mathbf{A}^{(i)}\mathbf{S}\|_F^2$. Unfortunately, we do not know $\mathbf{V}$, so instead we use a polynomial solver to check whether there exists such a $\mathbf{V}$. We remark that similar guessing strategies were employed by [RSW16, KPRW19, BWZ19, VVWZ23] and in particular, [RSW16] also uses a polynomial system in conjunction with the guessing strategy. Thus we write $\mathbf{Y} = \mathbf{V}\mathbf{S}$ and its pseudoinverse $\mathbf{W} = (\mathbf{V}\mathbf{S})^\dagger$ and check whether there exists a satisfying assignment to the above inequality, given the constraints (1) $\mathbf{Y}\mathbf{W}\mathbf{Y} = \mathbf{Y}$, (2) $\mathbf{W}\mathbf{Y}\mathbf{W} = \mathbf{W}$, and (3) $\mathbf{A}^{(i)}\mathbf{S}\mathbf{W}\mathbf{R}^{(i)}$ has orthonormal columns. Note that since $\mathbf{V} \in \mathbb{R}^{k \times d}$, then implementing the polynomial solver naïvely could require kd variables and thus use $2^{\Omega(dk)}$ runtime. Instead, we note that we only work with $\mathbf{V}\mathbf{S}$, which has dimension $k \times m$ for $m = \mathcal{O}\left(\frac{k^2}{\varepsilon^2}\log \ell\right)$, so that the polynomial solver only uses $2^{\mathrm{poly}(mk)}$ time.

Algorithm 1 Input to polynomial solver

```
Input: $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(\ell)}$, $\mathbf{S}$, $\alpha$
Output: Feasibility of polynomial system
Polynomial variables
Let $\mathbf{Y} = (\mathbf{VS}) \in \mathbb{R}^{k \times m}$ be $mk$ variables
Let $\mathbf{W} = (\mathbf{VS})^\dagger \in \mathbb{R}^{m \times k}$ be $mk$ variables
Let $\mathbf{R}^{(i)} \in \mathbb{R}^{k \times k}$ for each $i \in [\ell]$ be $\ell k^2$ variables
System constraints
$\mathbf{YWY} = \mathbf{Y}$, $\mathbf{WYW} = \mathbf{W}$
$\mathbf{A}^{(i)}\mathbf{SWR}^{(i)}$ has orthonormal columns
$\alpha \ge \|(\mathbf{A}^{(i)}\mathbf{SWR}^{(i)})(\mathbf{A}^{(i)}\mathbf{SWR}^{(i)})^\dagger\mathbf{A}^{(i)} - \mathbf{A}^{(i)}\|_F^2$
Run polynomial system solver
If feasible, output $\mathbf{V} = (\mathbf{A}^{(1)}\mathbf{SWR}^{(1)})^\dagger\mathbf{A}^{(1)}$. Otherwise, output $\perp$.
```

We first recall the following result for polynomial system satisfiability solvers.

Theorem 2.4 ([Ren92a, Ren92b, BPR96])**.** *Given a polynomial system $P(x_1, \ldots, x_n)$ over real numbers and m polynomial constraints $f_i(x_1, \ldots, x_n) \otimes_i 0$, where $\otimes \in \{>, \ge, =, \neq, \le, <\}$ for all $i \in [m]$, let d denote the maximum degree of all the polynomial constraints and let B denote the maximum size of the bit representation of the coefficients of all the polynomial constraints. Then there exists an algorithm that determines whether there exists a solution to the polynomial system P in time $(md)^{\mathcal{O}(n)} \cdot \mathrm{poly}(B)$.*

To apply Theorem 2.4, we utilize the following statement upper bounding the sizes of the bit representation of the coefficients of the polynomial constraints in our system.

Theorem 2.5 ([JPT13])**.** *Let $\mathcal{T} = \{x \in \mathbb{R}^n \mid f_1(x) \ge 0, \ldots, f_m(x) \ge 0\}$ be defined by m polynomials $f_i(x_1, \ldots, x_n)$ for $i \in [m]$ with degrees bounded by an even integer d and coefficients of magnitude at most M. Let $\mathcal{C}$ be a compact connected component of $\mathcal{T}$. Let $g(x_1, \ldots, x_n)$ be a polynomial of degree at most d with integer coefficients of magnitude at most M. Then the minimum nonzero magnitude that g takes over $\mathcal{C}$ is at least $(2^{4-n/2}\widetilde{M}d^n)^{-n2^nd^n}$, where $\widetilde{M} = \max(M, 2n + 2m)$.*

To perform dimensionality reduction, recall the following definition of affine embedding.

Definition 2.6 (Affine embedding)**.** *We say a matrix $\mathbf{S} \in \mathbb{R}^{n \times m}$ is an affine embedding for a matrix $\mathbf{A} \in \mathbb{R}^{d \times n}$ and a vector $\mathbf{b} \in \mathbb{R}^n$ if we have*

$$(1 - \varepsilon)\|\mathbf{xA} - \mathbf{b}\|_F^2 \le \|\mathbf{xAS} - \mathbf{bS}\|_F^2 \le (1 + \varepsilon)\|\mathbf{xA} - \mathbf{b}\|_F^2,$$

for all vectors $\mathbf{x} \in \mathbb{R}^d$.

We then apply the following affine embedding construction.

Lemma 2.7 (Lemma 11 in [CEM+15])**.** *Given $\delta, \varepsilon \in (0, 1)$ and a rank parameter $k > 0$, let $m = \mathcal{O}\left(\frac{k^2}{\varepsilon^2} \log \frac{1}{\delta}\right)$. For any matrix $\mathbf{A} \in \mathbb{R}^{d \times n}$, there exists a family $\mathcal{S}$ of random matrices in $\mathbb{R}^{n \times m}$, such that for $\mathbf{S} \sim \mathcal{S}$, we have that with probability at least $1 - \delta$, $\mathbf{S}$ is a one-sided affine embedding for a matrix $\mathbf{A} \in \mathbb{R}^{d \times n}$ and a vector $\mathbf{b} \in \mathbb{R}^n$.*

We now show a crucial structural property that allows us to distinguish between the case where a guess α for the optimal value OPT exceeds $(1+\varepsilon)\mathsf{OPT}$ or is smaller than $(1-\varepsilon)\mathsf{OPT}$ by simply looking at a polynomial system solver on an affine embedding.

Lemma 2.8. *Let $\mathbf{V}\in\mathbb{R}^{k\times d}$ be the optimal solution to the fair low-rank approximation problem for inputs $\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(\ell)}$, where $\mathbf{A}^{(i)}\in\mathbb{R}^{n_i\times d}$, and suppose $\mathsf{OPT}=\max_{i\in[\ell]}\|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V}-\mathbf{A}^{(i)}\|_F^2$. Let $\mathbf{S}$ be an affine embedding for $\mathbf{V}$ and let $\mathbf{W}=(\mathbf{VS})^\dagger\in\mathbb{R}^{k\times m}$. For $i\in[\ell]$, let $\mathbf{Z}^{(i)}=\mathbf{A}^{(i)}\mathbf{SW}\in\mathbb{R}^{n_i\times k}$ and $\mathbf{R}^{(i)}\in\mathbb{R}^{k\times k}$ be defined so that $\mathbf{A}^{(i)}\mathbf{SWR}^{(i)}$ has orthonormal columns. If $\alpha\ge(1+\varepsilon)\cdot\mathsf{OPT}$, then for each $i\in[\ell]$, $\alpha\ge\|(\mathbf{A}^{(i)}\mathbf{SWR}^{(i)})(\mathbf{A}^{(i)}\mathbf{SWR}^{(i)})^\dagger\mathbf{A}^{(i)}-\mathbf{A}^{(i)}\|_F^2$. If $\alpha<(1-\varepsilon)\cdot\mathsf{OPT}$, then there exists $i\in[\ell]$, such that $\alpha<\|(\mathbf{A}^{(i)}\mathbf{SWR}^{(i)})(\mathbf{A}^{(i)}\mathbf{SWR}^{(i)})^\dagger\mathbf{A}^{(i)}-\mathbf{A}^{(i)}\|_F^2$.*

Proof. By the Pythagorean theorem, we have that

$$(\mathbf{A}^{(i)}\mathbf{S})(\mathbf{VS})^\dagger=\underset{\mathbf{X}^{(i)}}{\operatorname{argmin}}\|\mathbf{X}^{(i)}\mathbf{VS}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2.$$

Thus, $\mathbf{Z}^{(i)}=\operatorname{argmin}_{\mathbf{X}^{(i)}}\|\mathbf{X}^{(i)}\mathbf{VS}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2$ for $\mathbf{Z}^{(i)}=\mathbf{A}^{(i)}\mathbf{SW}\in\mathbb{R}^{n_i\times k}$ and $\mathbf{W}=(\mathbf{VS})^\dagger\in\mathbb{R}^{m\times k}$.

Let $\mathbf{R}^{(i)}\in\mathbb{R}^{k\times k}$ be defined so that $\mathbf{A}^{(i)}\mathbf{SWR}^{(i)}$ has orthonormal columns. Thus, we have

$$\begin{aligned}\|(\mathbf{Z}^{(i)}\mathbf{R}^{(i)})(\mathbf{Z}^{(i)}\mathbf{R}^{(i)})^\dagger\mathbf{A}^{(i)}\mathbf{S}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2&=\|(\mathbf{Z}^{(i)})(\mathbf{Z}^{(i)})^\dagger\mathbf{A}^{(i)}\mathbf{S}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2\\&=\|\mathbf{Z}^{(i)}\mathbf{VS}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2\\&=\min_{\mathbf{X}^{(i)}}\|\mathbf{X}^{(i)}\mathbf{VS}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2,\end{aligned}$$

by the definition of $\mathbf{Z}^{(i)}$.

Suppose $\alpha\ge(1+\varepsilon)\cdot\mathsf{OPT}$. Since $\mathbf{S}$ is an affine embedding for $\mathbf{V}$, then we have that for all $\mathbf{X}^{(i)}\in\mathbb{R}^{n_i\times k}$,

$$\|\mathbf{X}^{(i)}\mathbf{VS}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2\le(1+\varepsilon)\|\mathbf{X}^{(i)}\mathbf{V}-\mathbf{A}^{(i)}\|_F^2.$$

In particular, we have

$$\min_{\mathbf{X}^{(i)}}\|\mathbf{X}^{(i)}\mathbf{VS}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2\le(1+\varepsilon)\min_{\mathbf{X}^{(i)}}\|\mathbf{X}^{(i)}\mathbf{V}-\mathbf{A}^{(i)}\|_F^2\le(1+\varepsilon)\mathsf{OPT}.$$

Then from the above argument, we have

$$\begin{aligned}\|(\mathbf{Z}^{(i)}\mathbf{R}^{(i)})(\mathbf{Z}^{(i)}\mathbf{R}^{(i)})^\dagger\mathbf{A}^{(i)}\mathbf{S}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2&=\min_{\mathbf{X}^{(i)}}\|\mathbf{X}^{(i)}\mathbf{VS}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2\\&\le(1+\varepsilon)\mathsf{OPT}\le\alpha.\end{aligned}$$

Since $\mathbf{Z}^{(i)}=\mathbf{A}^{(i)}\mathbf{SW}\in\mathbb{R}^{n_i\times k}$, then for $\alpha\ge(1+\varepsilon)\cdot\mathsf{OPT}$, we have that for each $i\in[\ell]$,

$$\alpha\ge\|(\mathbf{A}^{(i)}\mathbf{SWR}^{(i)})(\mathbf{A}^{(i)}\mathbf{SWR}^{(i)})^\dagger\mathbf{A}^{(i)}-\mathbf{A}^{(i)}\|_F^2.$$

On the other hand, suppose $\alpha<(1-\varepsilon)\cdot\mathsf{OPT}$. Let $i\in[\ell]$ be fixed so that

$$\mathsf{OPT}=\min_{\mathbf{X}^{(i)}}\|\mathbf{X}^{(i)}\mathbf{V}-\mathbf{A}^{(i)}\|_F^2.$$

Since $\mathbf{S}$ is an affine embedding for $\mathbf{V}$, we have that for all $\mathbf{X}^{(i)}\in\mathbb{R}^{n_i\times k}$,

$$(1-\varepsilon)\|\mathbf{X}^{(i)}\mathbf{V}-\mathbf{A}^{(i)}\|_F^2\le\|\mathbf{X}^{(i)}\mathbf{VS}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2,$$

Therefore,

$$(1-\varepsilon)\min_{\mathbf{X}^{(i)}}\|\mathbf{X}^{(i)}\mathbf{V}-\mathbf{A}^{(i)}\|_F^2 \le \min_{\mathbf{X}^{(i)}}\|\mathbf{X}^{(i)}\mathbf{V}\mathbf{S}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2$$

From the above, we have

$$\|(\mathbf{Z}^{(i)}\mathbf{R}^{(i)})(\mathbf{Z}^{(i)}\mathbf{R}^{(i)})^\dagger\mathbf{A}^{(i)}\mathbf{S}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2 = \min_{\mathbf{X}^{(i)}}\|\mathbf{X}^{(i)}\mathbf{V}\mathbf{S}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2.$$

Hence, putting these relations together,

$$\begin{aligned}\alpha < (1-\varepsilon)\mathsf{OPT} &= (1-\varepsilon)\min_{\mathbf{X}^{(i)}}\|\mathbf{X}^{(i)}\mathbf{V}-\mathbf{A}^{(i)}\|_F^2\\ &\le \min_{\mathbf{X}^{(i)}}\|\mathbf{X}^{(i)}\mathbf{V}\mathbf{S}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2\\ &= \|(\mathbf{Z}^{(i)}\mathbf{R}^{(i)})(\mathbf{Z}^{(i)}\mathbf{R}^{(i)})^\dagger\mathbf{A}^{(i)}\mathbf{S}-\mathbf{A}^{(i)}\mathbf{S}\|_F^2,\end{aligned}$$

as desired. □

We can thus utilize the structural property of Lemma 2.8 by using the polynomial system solver in Algorithm 1 on an affine embedding.

Corollary 2.9. *If $\alpha \ge (1+\varepsilon)\cdot\mathsf{OPT}$, then Algorithm 1 outputs a vector $\mathbf{U}\in\mathbb{R}^{n\times k}$ such that*

$$\alpha \ge \|\mathbf{U}\mathbf{U}^\dagger\mathbf{A}^{(i)}-\mathbf{A}^{(i)}\|_F^2.$$

If $\alpha < (1-\varepsilon)\cdot\mathsf{OPT}$, then Algorithm 1 outputs $\perp$

Algorithm 2 $(1+\varepsilon)$-approximation for fair low-rank approximation

```
Input: $\mathbf{A}^{(i)} \in \mathbb{R}^{n_i \times d}$ for all $i \in [\ell]$, rank parameter $k > 0$, accuracy parameter $\varepsilon \in (0,1)$
Output: $(1+\varepsilon)$-approximation for fair LRA
Let $\alpha$ be an $\ell$-approximation for the fair LRA problem
Let $\mathbf{S}$ be generated from a random affine embedding distribution
while Algorithm 1 on input $\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(\ell)}$, $\mathbf{S}$, and $\alpha$ does not return $\perp$ do
    Let $\mathbf{V}$ be the output of Algorithm 1 on input $\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(\ell)}$, $\mathbf{S}$, and $\alpha$
    $\alpha \leftarrow \frac{\alpha}{1+\varepsilon}$
end while
Return $\mathbf{V}$
```

Correctness of Algorithm 2 then follows from Corollary 2.9 and binary search on α. We now analyze the runtime of Algorithm 2.

Lemma 2.10. *The runtime of Algorithm 2 is at most $\frac{1}{\varepsilon}\operatorname{poly}(n)\cdot(2\ell)^{\mathcal{O}(N)}$, for $n=\sum_{i=1}^{\ell} n_i$ and $N=\operatorname{poly}\left(\ell,k,\frac{1}{\varepsilon}\right)$.*

Proof. Suppose the coefficients of $\mathbf{A}^{(i)}$ are bounded in magnitude by $2^{\operatorname{poly}(n)}$, where $n=\sum_{i=1}^{\ell} n_i$. The number of variables in the polynomial system is at most

$$N := 2mk+\ell k^2 = \operatorname{poly}\left(\ell,k,\frac{1}{\varepsilon}\right).$$

Each of the $\mathcal{O}(\ell)$ polynomial constraints has degree at most 20. Thus by Theorem 2.5, the minimum nonzero magnitude that any polynomial constraint takes over $\mathcal{C}$ is at least $(2^{4-N/2}2^{\mathrm{poly}(n)}20^N)^{-N2^N 20^N}$. Hence, the maximum bit representation required is $B = \mathrm{poly}(n) \cdot 2^{\mathcal{O}(N)}$. By Theorem 2.4, the runtime of the polynomial system solver is at most $(\mathcal{O}(\ell) \cdot 20)^{\mathcal{O}(N)} \cdot \mathrm{poly}(B) = \mathrm{poly}(n) \cdot (2\ell)^{\mathcal{O}(N)}$. We require at most $\mathcal{O}\left(\frac{1}{\varepsilon}\log \ell\right)$ iterations of the polynomial system solver. Thus, the total runtime is at most $\frac{1}{\varepsilon}\mathrm{poly}(n) \cdot (2\ell)^{\mathcal{O}(N)}$. □

Putting things together, we have:

Theorem 1.3. *Given an accuracy parameter $\varepsilon \in (0,1)$, there exists an algorithm that outputs $\widetilde{\mathbf{V}} \in \mathbb{R}^{k\times d}$ such that with probability at least $\frac{2}{3}$,*

$$\max_{i\in[\ell]} \|\mathbf{A}^{(i)}(\widetilde{\mathbf{V}})^{\dagger}\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F \le (1+\varepsilon) \cdot \min_{\mathbf{V}\in\mathbb{R}^{k\times d}} \max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^{\dagger}\mathbf{V} - \mathbf{A}^{(i)}\|_F.$$

The algorithm uses runtime $\frac{1}{\varepsilon}\mathrm{poly}(n) \cdot (2\ell)^{\mathcal{O}(N)}$, for $n = \sum_{i=1}^{\ell} n_i$ and $N = \mathrm{poly}\left(\ell, k, \frac{1}{\varepsilon}\right)$.

2.3 Bicriteria Algorithm

To achieve polynomial time for our bicriteria algorithm, we can no longer use a polynomial system solver. Instead, we observe that for sufficiently large p, we have $\max \|\mathbf{x}\|_\infty = (1 \pm \varepsilon)\|\mathbf{x}\|_p$. Thus, in place of optimizing $\min_{\mathbf{V}\in\mathbb{R}^{k\times d}} \max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^{\dagger}\mathbf{V} - \mathbf{A}^{(i)}\|_F$, we instead optimize $\min_{\mathbf{V}\in\mathbb{R}^{k\times d}} \left(\sum_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^{\dagger}\mathbf{V} - \mathbf{A}^{(i)}\|_F^p\right)^{1/p}$. However, the terms $\|\mathbf{A}^{(i)}\mathbf{V}^{\dagger}\mathbf{V} - \mathbf{A}^{(i)}\|_F^p$ are unwieldy to work with. Thus we instead use Dvoretzky's Theorem, i.e., Theorem 2.11, to embed L_2 into L_p, by generating matrices $\mathbf{G}$ and $\mathbf{H}$ so that $(1-\varepsilon)\|\mathbf{GMH}\|_p \le \|\mathbf{M}\|_F \le (1+\varepsilon)\|\mathbf{GMH}\|_p$, for all matrices $\mathbf{M} \in \mathbb{R}^{n\times d}$.

Now, writing $\mathbf{A} = \mathbf{A}^{(1)} \circ \ldots \circ \mathbf{A}^{(\ell)}$, it suffices to approximately solve $\min_{\mathbf{X}\in\mathbb{R}^{k\times d}} \|\mathbf{GAHSX} - \mathbf{GAH}\|_p$. Unfortunately, low-rank approximation with L_p loss still cannot be approximated to $(1+\varepsilon)$-factor in polynomial time, and in fact $\mathbf{GAH}$ has dimension $n' \times d'$ with $n' \ge n$ and $d' \ge d$. Hence, we first apply dimensionality reduction by appealing to a result of [WY23a] showing that there exists a matrix $\mathbf{S}$ that samples a "small" number of columns of $\mathbf{A}$ to provide a coarse bicriteria approximation to L_p low-rank approximation. Now to lift the solution back to dimension d, we would like to solve regression problem $\min_{\mathbf{X}} \|\mathbf{GAHSX} - \mathbf{GAH}\|_p$. To that end, we consider a Lewis weight sampling matrix $\mathbf{T}$ such that

$$\frac{1}{2}\|\mathbf{TGAHSX} - \mathbf{TGAH}\|_p \le \|\mathbf{GAHSX} - \mathbf{GAH}\|_p \le 2\|\mathbf{TGAHSX} - \mathbf{TGAH}\|_p.$$

We then note that $(\mathbf{TGAHS})^{\dagger}\mathbf{TGAH}$ is the minimizer of the problem $\min_{\mathbf{x}} \|\mathbf{TGAHSX} - \mathbf{TGAH}\|_F$, which only provides a small distortion to the L_p regression problem, since $\mathbf{TGAH}$ has a small number of rows due to the dimensionality reduction. By Dvoretzky's Theorem, we have that $(\mathbf{TGAHS})^{\dagger}\mathbf{TGA}$ is a "good" approximate solution to the original fair low-rank approximation problem. The algorithm appears in full in Algorithm 3.

We use the following notion of Dvoretzky's theorem to embed the problem into entrywise L_p loss.

Algorithm 3 Bicriteria approximation for fair low-rank approximation

```
Input: $\mathbf{A}^{(i)} \in \mathbb{R}^{n_i \times d}$ for all $i \in [\ell]$, rank parameter $k > 0$, trade-off parameter $c \in (0,1)$
Output: Bicriteria approximation for fair LRA
Generate Gaussian matrices $\mathbf{G} \in \mathbb{R}^{n' \times n}, \mathbf{H} \in \mathbb{R}^{d \times d'}$ through Theorem 2.11
Let $\mathbf{S} \in \mathbb{R}^{n' \times t}, \mathbf{Z} \in \mathbb{R}^{t \times d'}$ be the output of Theorem 2.12 on input $\mathbf{GAH}$
Let $\mathbf{T} \in \mathbb{R}^{s \times n'}$ be a Lewis weight sampling matrix for $\mathbf{GAHSX} - \mathbf{GAH}$
Let $\widetilde{\mathbf{V}} \leftarrow (\mathbf{TGAHS})^\dagger(\mathbf{TGA})$
Return $\widetilde{\mathbf{V}}$
```

Theorem 2.11 (Dvoretzky's Theorem, e.g., Theorem 1.2 in [PVZ17], Fact 15 in [SW18])**.** *Let $p \geq 1$ be a parameter and let*

$$m \gtrsim m(n,p,\varepsilon) = \begin{cases} \frac{p^p n}{\varepsilon^2}, & \varepsilon \leq (Cp)^{\frac{p}{2}} n^{-\frac{p-2}{2(p-1)}} \\ \frac{(np)^{p/2}}{\varepsilon}, & \varepsilon \in \left((Cp)^{\frac{p}{2}} n^{-\frac{p-2}{2(p-1)}}, \frac{1}{p} \right] \\ \frac{n^{p/2}}{p^{p/2}\varepsilon^{p/2}} \log^{p/2} \frac{1}{\varepsilon}, & \frac{1}{p} < \varepsilon < 1. \end{cases}$$

Then there exists a family $\mathcal{G}$ of random scaled Gaussian matrices with dimension $\mathbb{R}^{m \times n}$ such that for $G \sim \mathcal{G}$, with probability at least $1-\delta$, simultaneously for all $\mathbf{y} \in \mathbb{R}^n$, $(1-\varepsilon)\|\mathbf{y}\|_2 \leq \|\mathbf{Gy}\|_p \leq (1+\varepsilon)\|\mathbf{y}\|_2$.

We use the following algorithm from [WY23a] to perform dimensionality reduction so that switching between L_2 and L_p loss will incur smaller error. See also [CGK+17].

Theorem 2.12 (Theorem 1.5 in [WY23a])**.** *Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ and let $k \geq 1$. Let $s = \mathcal{O}(k \log\log k)$. Then there exists a polynomial-time algorithm that outputs a matrix $\mathbf{S} \in \mathbb{R}^{d \times t}$ that samples $t = \mathcal{O}\left(k(\log\log k)(\log^2 d)\right)$ columns of $\mathbf{A}$ and a matrix $\mathbf{Z} \in \mathbb{R}^{t \times d}$ such that $\|\mathbf{A} - \mathbf{ASZ}\|_p \leq 2^p \cdot \mathcal{O}(\sqrt{s}) \cdot \min_{\mathbf{U} \in \mathbb{R}^{n \times k}, \mathbf{V} \in \mathbb{R}^{k \times d}} \|\mathbf{A} - \mathbf{UV}\|_p$.*

We then show that Algorithm 3 provides a bicriteria approximation.

Lemma 2.13. *Let $\widetilde{\mathbf{V}}$ be the output of Algorithm 3. Then with probability at least $\frac{9}{10}$, $\max_{i \in [\ell]} \|\mathbf{A}^{(i)}(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F$ is at most $\ell^c \cdot 2^{1/c} \cdot \mathcal{O}(k(\log\log k)(\log d)) \max_{i \in [\ell]} \|\mathbf{A}^{(i)}(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F$, where c is the trade-off parameter input.*

Proof. We have

$$\max_{i \in [\ell]} \|\mathbf{A}^{(i)}(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F \leq \left(\sum_{i \in [\ell]} \|\mathbf{A}^{(i)}(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F^p \right)^{1/p}.$$

Let $\mathbf{A} = \mathbf{A}^{(1)} \circ \ldots \circ \mathbf{A}^{(\ell)}$ and let $\widetilde{\mathbf{U}} := \mathbf{AHS}$. For $i \in [\ell]$, let $\widetilde{\mathbf{U}^{(i)}}$ be the matrix of $\widetilde{\mathbf{U}}$ whose rows correspond with the rows of $\mathbf{A}^{(i)}$ in $\mathbf{A}$, i.e., let $\widetilde{\mathbf{U}^{(i)}}$ be the i-th block of rows of $\widetilde{\mathbf{U}}$.

By the optimality of $\mathbf{A}^{(i)}(\widetilde{\mathbf{V}})^\dagger$ with respect to the Frobenius norm, we have

$$\left(\sum_{i \in [\ell]} \|\mathbf{A}^{(i)}(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F^p \right)^{1/p} \leq \left(\sum_{i \in [\ell]} \|\widetilde{\mathbf{U}^{(i)}}\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F^p \right)^{1/p}$$

By Dvoretzky's Theorem, Theorem 2.11, with distortion $\varepsilon = \Theta(1)$, we have that with probability at least 0.99,

$$\left(\sum_{i\in[\ell]} \|\widetilde{\mathbf{U}^{(i)}}\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F^p\right)^{1/p} \le 2\left(\sum_{i\in[\ell]} \|\mathbf{G}\widetilde{\mathbf{U}^{(i)}}\widetilde{\mathbf{V}}\mathbf{H} - \mathbf{G}\mathbf{A}^{(i)}\mathbf{H}\|_p^p\right)^{1/p},$$

where we use $\|\cdot\|_p$ to denote the entry-wise p-norm. Writing $\widetilde{\mathbf{U}} = \widetilde{\mathbf{U}^{(1)}} \circ \ldots \circ \widetilde{\mathbf{U}^{(\ell)}}$, then we have

$$\left(\sum_{i\in[\ell]} \|\mathbf{G}\widetilde{\mathbf{U}^{(i)}}\widetilde{\mathbf{V}}\mathbf{H} - \mathbf{G}\mathbf{A}^{(i)}\mathbf{H}\|_p^p\right)^{1/p} = \|\mathbf{G}\widetilde{\mathbf{U}}\widetilde{\mathbf{V}}\mathbf{H} - \mathbf{G}\mathbf{A}\mathbf{H}\|_p.$$

By the choice of the Lewis weight sampling matrix $\mathbf{T}$, we have that with probability 0.99,

$$\begin{aligned}\|\mathbf{G}\widetilde{\mathbf{U}}\widetilde{\mathbf{V}}\mathbf{H} - \mathbf{G}\mathbf{A}\mathbf{H}\|_p &\le 2\|\mathbf{T}\mathbf{G}\widetilde{\mathbf{U}}\widetilde{\mathbf{V}}\mathbf{H} - \mathbf{T}\mathbf{G}\mathbf{A}\mathbf{H}\|_p\\ &\le 2\|\mathbf{T}\mathbf{G}\widetilde{\mathbf{U}}\widetilde{\mathbf{V}}\mathbf{H} - \mathbf{T}\mathbf{G}\mathbf{A}\mathbf{H}\|_{(p,2)}\\ &= 2\|\mathbf{TGAHS}(\mathbf{TGAHS})^\dagger(\mathbf{TGA})\mathbf{H} - \mathbf{TGAH}\|_{(p,2)},\end{aligned}$$

where $\|\mathbf{M}\|_{(p,2)}$ denotes the L_p norm of the vector consisting of the L_2 norms of the columns of $\mathbf{M}$, and the last equality follows due to our setting of $\widetilde{\mathbf{U}} = \mathbf{AHS}$ and $\widetilde{\mathbf{V}} \leftarrow (\mathbf{TGAHS})^\dagger(\mathbf{TGA})$, the latter in Algorithm 3. By optimality of $(\mathbf{TGAHS})^\dagger(\mathbf{TGA})\mathbf{H}$ for the choice of $\mathbf{X}$ in the minimization problem

$$\min_{\mathbf{X}\in\mathbb{R}^{t\times d'}} \|\mathbf{TGAHSX} - \mathbf{TGAH}\|_{(p,2)},$$

we have

$$\|\mathbf{TGAHS}(\mathbf{TGAHS})^\dagger(\mathbf{TGA})\mathbf{H} - \mathbf{TGAH}\|_{(p,2)} = \min_{\mathbf{X}\in\mathbb{R}^{t\times d'}} \|\mathbf{TGAHSX} - \mathbf{TGAH}\|_{(p,2)}.$$

Since $\mathbf{S} \in \mathbb{R}^{n'\times t}$ and $\mathbf{T}$ is a Lewis weight sampling matrix for $\mathbf{GAHSX} - \mathbf{GAH}$, then $\mathbf{T}$ has t rows, where $t = \mathcal{O}\left(k(\log\log k)(\log^2 d)\right)$ by Theorem 2.12. Thus, each column of $\mathbf{TGAH}$ has t entries, so that

$$\min_{\mathbf{X}\in\mathbb{R}^{t\times d'}} \|\mathbf{TGAHSX} - \mathbf{TGAH}\|_{(p,2)} \le \sqrt{t}\min_{\mathbf{X}\in\mathbb{R}^{t\times d'}} \|\mathbf{TGAHSX} - \mathbf{TGAH}\|_p.$$

By the choice of the Lewis weight sampling matrix $\mathbf{T}$, we have that with probability 0.99,

$$\min_{\mathbf{X}\in\mathbb{R}^{t\times d'}} \|\mathbf{TGAHSX} - \mathbf{TGAH}\|_p \le 2\min_{\mathbf{X}\in\mathbb{R}^{t\times d'}} \|\mathbf{GAHSX} - \mathbf{GAH}\|_p.$$

By Theorem 2.12, we have that with probability 0.99,

$$\min_{\mathbf{X}\in\mathbb{R}^{t\times d'}} \|\mathbf{GAHSX} - \mathbf{GAH}\|_p \le 2^p \cdot \mathcal{O}\left(\sqrt{s}\right) \cdot \min_{\mathbf{U}\in\mathbb{R}^{n'\times t},\mathbf{V}\in\mathbb{R}^{t\times d'}} \|\mathbf{UV} - \mathbf{GAH}\|_p,$$

for $s = \mathcal{O}\left(k\log\log k\right)$. Let $\mathbf{V}^* = \operatorname{argmin}_{\mathbf{V}\in\mathbb{R}^{k\times d}} \max_{i\in[\ell]} \|\mathbf{A}^{(i)} - \mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V}\|_p$. Then since $\mathbf{UV}$ has rank t with $t \ge k$, we have

$$\min_{\mathbf{U}\in\mathbb{R}^{n'\times t},\mathbf{V}\in\mathbb{R}^{t\times d'}} \|\mathbf{UV} - \mathbf{GAH}\|_p \le \|\mathbf{GA}(\mathbf{V}^*)^\dagger\mathbf{V}^*\mathbf{H} - \mathbf{GAH}\|_p$$

$$= \left(\sum_{i \in [\ell]} \| \mathbf{G} \mathbf{A}^{(i)} (\mathbf{V}^*)^\dagger \mathbf{V}^* \mathbf{H} - \mathbf{G} \mathbf{A}^{(i)} \mathbf{H} \|_p^p \right)^{1/p}.$$

By Dvoretzky's Theorem, Theorem 2.11, with distortion $\varepsilon = \Theta(1)$, we have that with probability at least 0.99,

$$\left(\sum_{i \in [\ell]} \| \mathbf{G} \mathbf{A}^{(i)} (\mathbf{V}^*)^\dagger \mathbf{V}^* \mathbf{H} - \mathbf{G} \mathbf{A}^{(i)} \mathbf{H} \|_p^p \right)^{1/p} \le 2 \left(\sum_{i \in [\ell]} \| \mathbf{A}^{(i)} (\mathbf{V}^*)^\dagger \mathbf{V}^* - \mathbf{A}^{(i)} \|_F^p \right)^{1/p}.$$

For $p = c \log \ell$ with $c < 1$, we have

$$\left(\sum_{i \in [\ell]} \| \mathbf{A}^{(i)} (\mathbf{V}^*)^\dagger \mathbf{V}^* - \mathbf{A}^{(i)} \|_F^p \right)^{1/p} \le 2^{1/c} \max_{i \in [\ell]} \| \mathbf{A}^{(i)} (\widetilde{\mathbf{V}})^\dagger \widetilde{\mathbf{V}} - \mathbf{A}^{(i)} \|_F.$$

Putting together these inequalities successively, we ultimately have

$$\max_{i \in [\ell]} \| \mathbf{A}^{(i)} (\widetilde{\mathbf{V}})^\dagger \widetilde{\mathbf{V}} - \mathbf{A}^{(i)} \|_F \le 2^p \cdot 2^{1/c} \cdot \mathcal{O}\left(\sqrt{st}\right) \max_{i \in [\ell]} \| \mathbf{A}^{(i)} (\widetilde{\mathbf{V}})^\dagger \widetilde{\mathbf{V}} - \mathbf{A}^{(i)} \|_F,$$

for $p = c \log \ell$, $s = \mathcal{O}(k \log \log k)$, and $t = \mathcal{O}\left(k (\log \log k)(\log^2 d)\right)$. Therefore, we have

$$\max_{i \in [\ell]} \| \mathbf{A}^{(i)} (\widetilde{\mathbf{V}})^\dagger \widetilde{\mathbf{V}} - \mathbf{A}^{(i)} \|_F \le \ell^c \cdot 2^{1/c} \cdot \mathcal{O}\left(k (\log \log k)(\log d)\right) \max_{i \in [\ell]} \| \mathbf{A}^{(i)} (\widetilde{\mathbf{V}})^\dagger \widetilde{\mathbf{V}} - \mathbf{A}^{(i)} \|_F.$$

□

Since the generation of Gaussian matrices and the Lewis weight sampling matrix both only require polynomial time, it follows that our algorithm uses polynomial time overall. Hence, we have:

Theorem 1.4. *Given a trade-off parameter $c \in (0,1)$, there exists an algorithm that outputs $\widetilde{\mathbf{V}} \in \mathbb{R}^{t \times d}$ for $t = \mathcal{O}\left(k (\log \log k)(\log^2 d)\right)$ such that with probability at least $\frac{2}{3}$,*

$$\max_{i \in [\ell]} \| \mathbf{A}^{(i)} (\widetilde{\mathbf{V}})^\dagger \widetilde{\mathbf{V}} - \mathbf{A}^{(i)} \|_F \le \ell^c \cdot 2^{1/c} \cdot \mathcal{O}\left(k (\log \log k)(\log d)\right) \cdot \min_{\mathbf{V} \in \mathbb{R}^{k \times d}} \max_{i \in [\ell]} \| \mathbf{A}^{(i)} \mathbf{V}^\dagger \mathbf{V} - \mathbf{A}^{(i)} \|_F.$$

The algorithm uses runtime polynomial in n and d.

3 Socially Fair Column Subset Selection

In this section, we consider socially fair column subset selection, where the goal is to identify a matrix $\mathbf{C} \in \mathbb{R}^{d \times k}$ that selects k columns to minimize $\min_{\mathbf{C} \in \mathbb{R}^{d \times k}, \|\mathbf{C}\|_0 \le k, \mathbf{B}^{(i)}} \max_{i \in [\ell]} \| \mathbf{A}^{(i)} \mathbf{C} \mathbf{B}^{(i)} - \mathbf{A}^{(i)} \|_F$. Note that each matrix $\mathbf{B}^{(i)} \in \mathbb{R}^{n_i \times d}$ is implicitly defined in the minimization problem. Intuitively, the goal is to select k columns of $\mathbf{A}$ so that each group $\mathbf{A}^{(i)}$ can be well-approximated by some matrix $\mathbf{B}^{(i)}$ spanned by those k columns.

To prove correctness of Algorithm 4, we first require the following structural property:

Algorithm 4 Bicriteria approximation for fair column subset selection

```
Input: $\mathbf{A}^{(i)} \in \mathbb{R}^{n_i \times d}$ for all $i \in [\ell]$, rank parameter $k > 0$, trade-off parameter $c \in (0,1)$
Output: Bicriteria approximation for fair column subset selection
Acquire $\widetilde{\mathbf{V}}$ from Algorithm 3
Generate Gaussian $\mathbf{G} \in \mathbb{R}^{n' \times n}$ through Theorem 2.11
Let $\mathbf{S} \in \mathbb{R}^{d \times k'}$ be a leverage score sampling matrix that samples $k' = \mathcal{O}(k \log k)$ columns of $\widetilde{\mathbf{V}}$
$\mathbf{M}^{(i)} = \mathbf{S}^\dagger (\widetilde{\mathbf{V}})^\dagger \widetilde{\mathbf{V}}$ for all $i \in [\ell]$
Return $\mathbf{A}^{(i)}\mathbf{S}$, $\{\mathbf{M}^{(i)}\}$
```

Lemma 3.1. *Let $\varepsilon \in (0,1)$ and $\mathbf{x} \in \mathbb{R}^\ell$ and let $p = \mathcal{O}\left(\frac{1}{\varepsilon} \log \ell\right)$. Then $\|\mathbf{x}\|_\infty \le \|\mathbf{x}\|_p \le (1+\varepsilon)\|\mathbf{x}\|_\infty$.*

Proof. Since it is clear that $\|\mathbf{x}\|_\infty \le \|\mathbf{x}\|_p$, then it remains to prove $\|\mathbf{x}\|_p \le (1+\varepsilon)\|\mathbf{x}\|_\infty$. Note that we have $\|\mathbf{x}\|_p^p \le \|\mathbf{x}\|_\infty^p \cdot \ell$. To achieve $\ell^{1/p} \le (1+\varepsilon)$, it suffices to have $\frac{1}{p} \log \ell \le \log(1+\varepsilon)$. Since $\log(1+\varepsilon) = \mathcal{O}(\varepsilon)$ for $\varepsilon \in (0,1)$, then for $p = \mathcal{O}\left(\frac{1}{\varepsilon} \log \ell\right)$, we have $\ell^{1/p} \le (1+\varepsilon)$, and the desired claim follows. □

We now give the correctness guarantees of Algorithm 4.

Lemma 3.2. *Let $\mathbf{S}, \mathbf{M}^{(1)}, \ldots, \mathbf{M}^{(\ell)}$ be the output of Algorithm 4. Then with probability at least* 0.8*, $\max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{S}\mathbf{M}^{(i)} - \mathbf{A}^{(i)}\|_F$ is at most $\ell^c \cdot 2^{1/c} \cdot \mathcal{O}(k(\log\log k)(\log d)) \min_{\mathbf{V} \in \mathbb{R}^{k \times d}} \max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F$.*

Proof. Let $\widetilde{\mathbf{V}} \in \mathbb{R}^{t \times d}$ be the output of Algorithm 3, where $t = \mathcal{O}\left(k(\log\log k)(\log^2 d)\right)$. Then with probability at least $\frac{2}{3}$, we have

$$\max_{i \in [\ell]} \|\mathbf{A}^{(i)}(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F \le \ell^c \cdot 2^{1/c} \cdot \mathcal{O}(k(\log\log k)(\log d)) \min_{\mathbf{V} \in \mathbb{R}^{k \times d}} \max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F.$$

Therefore, we have

$$\max_{i \in [\ell]} \min_{\mathbf{B}^{(i)}} \|\mathbf{B}^{(i)}\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F \le \ell^c \cdot 2^{1/c} \cdot \mathcal{O}(k(\log\log k)(\log d)) \min_{\mathbf{V} \in \mathbb{R}^{k \times d}} \max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F.$$

Let p be a sufficiently large parameter to be fixed. By Dvoretzky's theorem, i.e., Theorem 2.11, with $\varepsilon = \mathcal{O}(1)$, we have

$$\max_{i \in [\ell]} \min_{\mathbf{B}^{(i)}} \|\mathbf{G}^{(i)}\mathbf{B}^{(i)}\widetilde{\mathbf{V}} - \mathbf{G}^{(i)}\mathbf{A}^{(i)}\|_{p,2} \le \mathcal{O}(1) \cdot \max_{i \in [\ell]} \min_{\mathbf{B}^{(i)}} \|\mathbf{B}^{(i)}\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F.$$

For sufficiently large $p = \mathcal{O}(\log \ell)$, we have by Lemma 3.1,

$$\max_{i \in [\ell]} \min_{\mathbf{B}^{(i)}} \|\mathbf{B}^{(i)}\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F. \le \mathcal{O}(1) \cdot \min_{\mathbf{B}^{(i)}} \|\mathbf{G}^{(i)}\mathbf{B}^{(i)}\widetilde{\mathbf{V}} - \mathbf{G}^{(i)}\mathbf{A}^{(i)}\|_{p,2}.$$

By a change of variables, we have

$$\min_{\mathbf{X}} \|\mathbf{X}\widetilde{\mathbf{V}} - \mathbf{G}\mathbf{A}\|_{p,2} \le \min_{\mathbf{B}} \|\mathbf{G}\mathbf{B}\widetilde{\mathbf{V}} - \mathbf{G}\mathbf{A}\|_{p,2}.$$

Note that minimizing $\|\mathbf{X}\widetilde{\mathbf{V}} - \mathbf{G}\mathbf{A}\|_{p,2}$ over all $\mathbf{X}$ corresponds to minimizing $\|\mathbf{X}_i\widetilde{\mathbf{V}} - \mathbf{G}_i\mathbf{A}\|_2$ over all $i \in [n']$. However, an arbitrary choice of $\mathbf{X}_i$ may not correspond to selecting columns of $\mathbf{A}$. Thus we apply a leverage score sampling matrix $\mathbf{S}$ to sample columns of $\widetilde{\mathbf{V}}$ which will correspondingly sample columns of $\mathbf{GA}$. Then by Lemma 1.10, we have

$$\min_{\mathbf{X}} \|\mathbf{X}\widetilde{\mathbf{V}}\mathbf{S} - \mathbf{GAS}\|_{p,2} \le \mathcal{O}\left(1\right) \cdot \min_{\mathbf{X}} \|\mathbf{X}\widetilde{\mathbf{V}} - \mathbf{GA}\|_{p,2}.$$

Putting these together, we have

$$\min_{\mathbf{X}} \|\mathbf{X}\widetilde{\mathbf{V}}\mathbf{S} - \mathbf{GAS}\|_{p,2} \le \ell^c \cdot 2^{1/c} \cdot \mathcal{O}\left(k(\log\log k)(\log d)\right) \min_{\mathbf{V}\in\mathbb{R}^{k\times d}} \max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F. \tag{1}$$

Let S be the selected columns from Algorithm 4 by $\mathbf{S}$ and note that $\mathbf{A}^{(i)}\mathbf{S}\mathbf{S}^\dagger(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}}$ is in the column span of $\mathbf{A}^{(i)}\mathbf{S}$ for each $i \in [\ell]$. By Dvoretzky's theorem, i.e., Theorem 2.11, with $\varepsilon = \mathcal{O}\left(1\right)$ and a fixed parameter p, we have

$$\max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{S}\mathbf{S}^\dagger(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F \le \mathcal{O}\left(1\right) \cdot \max_{i\in[\ell]} \|\mathbf{G}^{(i)}\mathbf{A}^{(i)}\mathbf{S}\mathbf{S}^\dagger(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{G}^{(i)}\mathbf{A}^{(i)}\|_{p,2}.$$

For sufficiently large p, we have

$$\max_{i\in[\ell]} \|\mathbf{G}^{(i)}\mathbf{A}^{(i)}\mathbf{S}\mathbf{S}^\dagger(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{G}^{(i)}\mathbf{A}^{(i)}\|_{p,2} \le \mathcal{O}\left(1\right) \cdot \|\mathbf{GAS}\mathbf{S}^\dagger(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{GA}\|_{p,2}.$$

By the correctness of the leverage score sampling matrix, we have

$$\|\mathbf{GAS}\mathbf{S}^\dagger(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{GA}\|_{p,2} \le \mathcal{O}\left(1\right) \cdot \|\mathbf{GAS}\mathbf{S}^\dagger(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}}\mathbf{S} - \mathbf{GAS}\|_{p,2}.$$

Observe that minimizing $\|\mathbf{X}\widetilde{\mathbf{V}}\mathbf{S} - \mathbf{GAS}\|_{p,2}$ over all $\mathbf{X}$ corresponds to minimizing $\|\mathbf{X}_i\widetilde{\mathbf{V}}\mathbf{S} - \mathbf{G}_i\mathbf{AS}\|_2$ over all $i \in [n']$. Moreover, the closed-form solution of the L_2 minization problem is

$$\mathbf{G}_i\mathbf{AS}\mathbf{S}^\dagger(\widetilde{\mathbf{V}})^\dagger = \operatorname*{argmin}_{\mathbf{X}_i} \|\mathbf{X}_i\widetilde{\mathbf{V}}\mathbf{S} - \mathbf{G}_i\mathbf{AS}\|_2.$$

Therefore, we have

$$\|\mathbf{GAS}\mathbf{S}^\dagger(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}}\mathbf{S} - \mathbf{GAS}\|_{p,2} \le \min_{\mathbf{X}} \|\mathbf{X}\widetilde{\mathbf{V}}\mathbf{S} - \mathbf{GAS}\|_{p,2}.$$

Putting things together, we have

$$\max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{S}\mathbf{S}^\dagger(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F \le \mathcal{O}\left(1\right) \cdot \min_{\mathbf{X}} \|\mathbf{X}\widetilde{\mathbf{V}}\mathbf{S} - \mathbf{GAS}\|_{p,2}. \tag{2}$$

By Equation 1 and Equation 2 and a rescaling of the constant hidden inside the big Oh notation, we have

$$\max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{S}\mathbf{S}^\dagger(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}} - \mathbf{A}^{(i)}\|_F \le \ell^c \cdot 2^{1/c} \cdot \mathcal{O}\left(k(\log\log k)(\log d)\right) \min_{\mathbf{V}\in\mathbb{R}^{k\times d}} \max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{V}^\dagger\mathbf{V} - \mathbf{A}^{(i)}\|_F.$$

The desired claim then follows from the setting of $\mathbf{M}^{(i)} = \mathbf{S}^\dagger(\widetilde{\mathbf{V}})^\dagger\widetilde{\mathbf{V}}$ for $i \in [\ell]$ by Algorithm 4. □

We also clearly have the following runtime guarantees on Algorithm 4.

Lemma 3.3. *The runtime of Algorithm 4 is* $\mathrm{poly}(n, d)$*.*

By combining Lemma 3.2 and Lemma 3.3, we have:

Theorem 1.5. *Given input matrices* $\mathbf{A}^{(i)} \in \mathbb{R}^{n_i\times d}$ *with* $n = \sum n_i$*, there exists an algorithm that selects a set* S *of* $k' = \mathcal{O}\left(k\log k\right)$ *columns such that with probability at least* $\frac{2}{3}$*,* S *is a* $\mathcal{O}\left(k(\log\log k)(\log d)\right)$*-approximation to the fair column subset selection problem. The algorithm uses runtime polynomial in* n *and* d*.*

4 Empirical Evaluations for Socially Fair Low-Rank Approximation

We conduct a number of experiments for socially fair low-rank approximation.

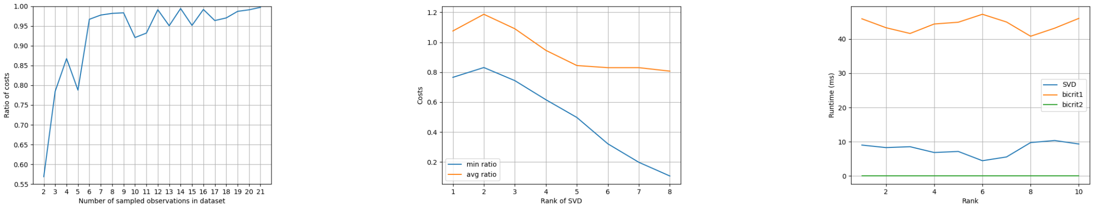
4.1 Real-World Datasets

In this section, we describe our empirical evaluations for socially fair low-rank approximation on real-world datasets.

Credit card dataset. We used the Default of Credit Card Clients dataset [YL09], which has 30,000 observations across 23 features, including 17 numeric features. The dataset is a common human-centric data for experiments on fairness and was previously used as a benchmark by [STM+18] for studies on fair PCA. The study collected information from various customers including multiple previous payment statements, previous payment delays, and upcoming bill statements, as well as if the customer was able to pay the upcoming bill statement or not, i.e., defaulted on the bill statement. The dataset was accessed through the UCI repository [Yeh16].

Experimental setup. For the purposes of reproducibility, our empirical evaluations were conducted using Python 3.10 using a 64-bit operating system on an AMD Ryzen 7 5700U CPU, with 8GB RAM and 8 cores with base clock 1.80 GHz. Our code is available at `https://github.com/samsonzhou/SVWZ24`. We compare our bicriteria algorithm from Algorithm 3 against the standard non-fair low-rank approximation algorithm that outputs the top k right singular vectors from the singular value decomposition. Gender was used as the sensitive attribute, so that all observations with one gender formed the matrix $\mathbf{A}^{(1)}$ and the other observations formed the matrix $\mathbf{A}^{(2)}$. As in Algorithm 3, we generate normalized Gaussian matrices $\mathbf{G}$ and $\mathbf{H}$ and then use L_p Lewis weight sampling to generate a matrix $\mathbf{T}$. We generate matrices $\mathbf{T}$, $\mathbf{G}$, and $\mathbf{H}$ with a small number of dimensions and thus do not compute the sampling matrix $\mathbf{S}$ but instead use the full matrix. We first sampled a small number s of rows from $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$ and compared our bicriteria algorithm to the standard non-fair low-rank approximation algorithm baseline. We plot the minimum ratio for $s \in \{2, 3, \ldots, 21\}$, $k = 1$, and $p = 1$ over $10,000$ iterations for each setup in Figure 1a. Similarly, we plot both the minimum and average ratios for $s = 1000$, $k \in \{1, 2, \ldots, 7, 8\}$, and $p = 1$ over 200 iterations in Figure 1b, where we permit the bicriteria solution to have rank $2k$. Finally, we compare the runtimes of the algorithms in Figure 1c, separating runtimes for our bicriteria into the total runtime `bicrit1` that includes the process of generating the Gaussian matrices and performing the Lewis weight sampling, as well as the runtime `bicrit2` for the step for extracting the factors similar to `SVD`, which measures the runtime for extracting the factors. Across all experiments in Figure 1, we used Gaussian matrices $\mathbf{G}$ with 30 rows and $\mathbf{H}$ with 30 columns.

Results and discussion. Our empirical evaluations serve as a simple proof-of-concept demonstrating that our bicriteria algorithm can perform significantly better for socially fair low-rank approximation, even without allowing for more than k factors. In particular, all ratios in Figure 1a and most ratios in are less than 1, indicating that the cost of our algorithm is less than the cost of the baseline, so that smaller ratios represent better performance. In fact, in some cases the ratio was less than 0.6, demonstrating the superiority of our algorithm. Finally, we show in Figure 1c that due to the small number of rows acquired by Lewis weight sampling, the final extraction of the factors is significantly faster by our algorithm. However, the main runtime bottleneck is the



(a) Costs over subsampled data (b) Costs over different ranks k (c) Runtime evaluations

Fig. 1: Empirical evaluations on the Default Credit dataset.

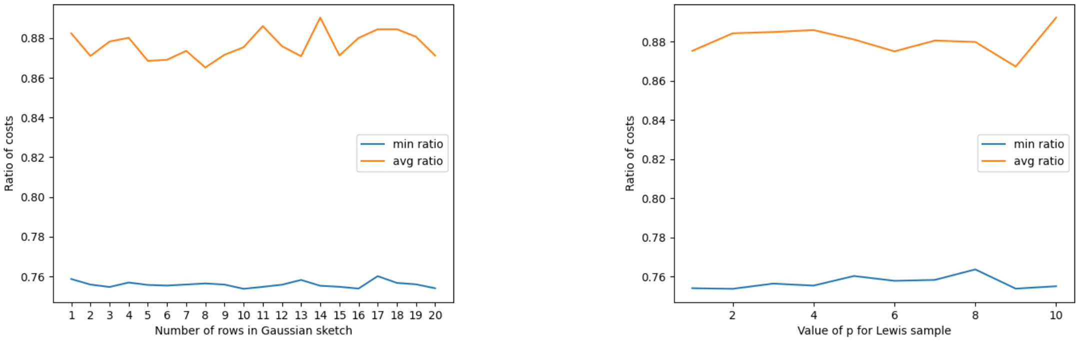
process of generating the Gaussian matrices and performing the subsequent Lewis weight sampling. As we use an iterative procedure to approximate the L_p Lewis weights with 10 iterations, this translates to computing 10 matrix inversions. Hence it is likely this gap can be improved with further optimizations to Lewis weight approximation algorithms.

4.2 Synthetic Datasets

In this section, we remark on a number of additional empirical evaluations for socially fair low-rank approximation. We first compare our bicriteria algorithm to the standard low-rank approximation baseline over a simple synthetic example. We then describe a simple fixed input matrix for socially fair low-rank approximation that serves as a proof-of-concept similarly demonstrating the improvement of fair low-rank approximation optimal solutions over the optimal solutions for standard low-rank approximation.

Synthetic dataset. Next, we show that for a simple dataset with two groups, each with two observations across four features, the performance of the fair low-rank approximation algorithm can be much better than standard low-rank approximation algorithm on the socially fair low-rank objective, even without allowing for bicriteria rank. For our synthetic dataset, we generate simple $\mathbf{A}^{(1)} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$ and $\mathbf{A}^{(2)} = \begin{bmatrix} 0 & 0 & 1.99 & 0 \\ 0 & 0 & 0 & 1.99 \end{bmatrix}$. An optimal fair low-rank solution for $k = 2$ is $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$, which gives cost 4. However, due to the weight of the items in $\mathbf{A}^{(1)}$, the standard low-rank algorithm will output low-rank factors with cost $2 \cdot 1.99^2 \approx 7.92$, such as the factors $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$. Thus the ratio between the objectives is roughly $\frac{1}{2}$, i.e., the improvement is roughly a factor of two. We show in Figure 2 that our bicriteria algorithm achieves similar improvement.

Experimental setup. Again, we compare our bicriteria algorithm from Algorithm 3 against the standard non-fair low-rank approximation algorithm that outputs the top k right singular vectors from the singular value decomposition. Per Dvoretzky's Theorem, c.f., Theorem 2.11, we generate normalized Gaussian matrices $\mathbf{G}$ and $\mathbf{H}$ of varying size and then use L_p Lewis weight sampling to generate a matrix $\mathbf{T}$ with varying values of p. We generate matrices $\mathbf{T}$, $\mathbf{G}$, and $\mathbf{H}$ with a small number of dimensions and thus do not compute the sampling matrix $\mathbf{S}$ but instead use the full matrix. We first fix $p = 1$ and iterate over the number of rows/columns in the Gaussian sketch in the range $\{1, 2, \ldots, 19, 20\}$ in Figure 2a. We then fix the Gaussian sketch to have three rows/columns and iterate over $p \in \{1, 2, \ldots, 9, 10\}$ in Figure 2b. For each of the variables, we compare the outputs of the two algorithms across 100 iterations and plot the ratio of their fair costs. In particular, we set the same rank parameter of $k = 2$ for both algorithms; we remark that the theoretical guarantees for our bicriteria algorithm are even stronger when we permit the solution to have rank k' for $k' > k$.



(a) Ratio of costs over different number of rows/columns in the Gaussian sketch matrix.

(b) Ratio of costs over different values of p in the Lewis sampling process.

Fig. 2: Ratio of the cost of our bicriteria algorithm to the cost of the standard low-rank approximation solution for $k = 2$, across 100 iterations.

Results and discussion. Our empirical evaluations in Figure 2 show that our algorithms can perform significantly better for socially fair low-rank approximation. We note that in both across all values of the number of rows/columns in the Gaussian sketch in Figure 2a and across all values of p in the Lewis weight sampling parameter in Figure 2b, the average ratio between our bicriteria solution and the standard low-rank approximation is less than 0.9. We remark that any ratio less than 1 demonstrates the superiority of our algorithm, with smaller ratios indicating better performance. In fact, the minimum ratio is as low as 0.76.

Proof-of-concept. Finally, we give a toy example using a synthetic dataset showing the importance of considering fairness in low-rank approximation. Namely, we show that even for a simple dataset with four groups, each with a single observation across two features, the performance of the fair low-rank approximation algorithm can be much better than standard low-rank approximation algorithm on the socially fair low-rank objective. In this setup, we repeatedly generate matrices $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}, \mathbf{A}^{(4)} \in \{0, 1\}^2$, with $\mathbf{A}^{(1)} = (1, 0)$ and $\mathbf{A}^{(2)} = \mathbf{A}^{(3)} = \mathbf{A}^{(4)} = (0, 1)$. The optimal fair low-rank solution is $\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)$ but due to the extra instances of $(0, 1)$, the standard low-rank algorithm will output the factor $(0, 1)$. Thus the optimal fair solution achieves value $\frac{1}{4}$ on the socially fair low-rank approximation objective while the standard low-rank approximation solution achieves value $\frac{1}{2}$, so that the ratio is a 50% improvement.

5 Conclusion

In this paper, we study algorithms for socially fair low-rank approximation and column subset selection. Although we show that even a constant-factor approximation to fair low-rank approximation requires exponential time under certain standard complexity hypotheses, we give an algorithm that is a substantial improvement over the naïve approach for any constant number of groups. We also give a bicriteria approximation algorithms for fair low-rank approximation and fair column subset selection that runs in polynomial time. Finally, we perform a number of empirical evaluations serving as a simple proof-of-concept demonstrating the practical nature of our theoretical findings. It is our hope that our work is an important step toward better understanding of fairness in numerical linear algebra.

Our work also leads to a number of interesting future directions. For example, there are various notions of fairness, including (1) *disparate impact*, which requires that the output of an algorithm respects a desired distribution that protects various subpopulations in the decision-making process [CKLV17, RS18, BIO$^+$19, BCFN19, BGK$^+$18, AEK$^+$20, EBTD20, DMV22, CXZC24], (2) *individual fairness*, where each element should incur a cost that is within a reasonable amount with respect to the overall dataset [JKL20b, MV20, NC21, VY22], or (3) *representative fairness*, where the number of representative elements of each subpopulation should be relatively balanced [KAM19, JNN20, AKSZ22, NNJ22, TGOO22, HMV23]. A natural direction is the capabilities and limitations of randomized linear algebra for other notions of fairness. Another question is the efficient construction of ε-coresets with minimal size for socially fair subset selection. Finally, one can ask whether similar guarantees are possible in settings whether the input matrix is not centralized, but rather for example, distributed or arrives over a data stream. Indeed, there are known analogs [CDW18, BDM$^+$20, JLL$^+$21, WY23a] for a number of the central tools used in the main algorithms of this paper.

References

[AAK$^+$22] Jacob D. Abernethy, Pranjal Awasthi, Matthäus Kleindessner, Jamie Morgenstern, Chris Russell, and Jie Zhang. Active sampling for min-max fairness. In *International Conference on Machine Learning, ICML*, pages 53–65, 2022. 6, 33

[ABV21] Mohsen Abbasi, Aditya Bhaskara, and Suresh Venkatasubramanian. Fair clustering via equitable group representations. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 504–514, 2021. 6

[ADW19] Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pages 120–129. PMLR, 2019. 6

[AEK$^+$20] Sara Ahmadian, Alessandro Epasto, Marina Knittel, Ravi Kumar, Mohammad Mahdian, Benjamin Moseley, Philip Pham, Sergei Vassilvitskii, and Yuyan Wang. Fair hierarchical clustering. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, 2020. 25

[AKSZ22] Haris Angelidakis, Adam Kurpisz, Leon Sering, and Rico Zenklusen. Fair and fast k-center clustering for data summarization. In *International Conference on Machine Learning, ICML*, pages 669–702, 2022. 25

[BCFN19] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. Fair algorithms for clustering. *Advances in Neural Information Processing Systems*, 32, 2019. 25

[BDM$^+$20] Vladimir Braverman, Petros Drineas, Cameron Musco, Christopher Musco, Jalaj Upadhyay, David P. Woodruff, and Samson Zhou. Near optimal linear algebra in the online and sliding window models. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 517–528, 2020. 25

[BG18] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency, FAT*, volume 81 of *Proceedings of Machine Learning Research*, pages 77–91, 2018. 2

[BGK00] Andreas Brieden, Peter Gritzmann, and Victor Klee. Inapproximability of some geometric and quadratic optimization problems. *Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems*, pages 96–115, 2000. 3, 11

[BGK$^+$18] Ioana O Bercea, Martin Groß, Samir Khuller, Aounon Kumar, Clemens Rösner, Daniel R Schmidt, and Melanie Schmidt. On the cost of essentially fair clusterings. *arXiv preprint arXiv:1811.10319*, 2018. 25

[BHJ$^+$17] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*, 2017. 6

[BHJ$^+$21] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 50(1):3–44, 2021. 1, 2

[BIO$^+$19] Arturs Backurs, Piotr Indyk, Krzysztof Onak, Baruch Schieber, Ali Vakilian, and Tal Wagner. Scalable fair clustering. In *International Conference on Machine Learning*, pages 405–413. PMLR, 2019. 25

[BPR96] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. On the combinatorial and algebraic complexity of quantifier elimination. *J. ACM*, 43(6):1002–1045, 1996. 13

[BS16] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *California law review*, pages 671–732, 2016. 2

[BWZ19] Frank Ban, David P. Woodruff, and Qiuyi (Richard) Zhang. Regularized weighted low rank approximation. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 4061–4071, 2019. 12

[CDW18] Graham Cormode, Charlie Dickens, and David P. Woodruff. Leveraging well-conditioned bases: Streaming and distributed summaries in minkowski p-norms. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, Proceedings of Machine Learning Research, pages 1048–1056, 2018. 25

[CEM$^+$15] Michael B. Cohen, Sam Elder, Cameron Musco, Christopher Musco, and Madalina Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC*, pages 163–172, 2015. 13

[CGK$^+$17] Flavio Chierichetti, Sreenivas Gollapudi, Ravi Kumar, Silvio Lattanzi, Rina Panigrahy, and David P. Woodruff. Algorithms for l_p low-rank approximation. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, pages 806–814, 2017. 17

[Cho17] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017. 1, 2

[CKLV17] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 5029–5037, 2017. 25

[CLS19] Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC)*, 2019. 35

[CMM17] Michael B. Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1758–1777, 2017. 8

[CMV22] Eden Chlamtáč, Yury Makarychev, and Ali Vakilian. Approximating fair clustering with cascaded norm objectives. In *Proceedings of the 2022 annual ACM-SIAM symposium on discrete algorithms (SODA)*, pages 2664–2683. SIAM, 2022. 6

[CP15] Michael B. Cohen and Richard Peng. L_p row sampling by lewis weights. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC*, pages 183–192, 2015. 10

[CP19] Xue Chen and Eric Price. Active regression via linear-sample sparsification. In *Conference on Learning Theory, COLT*, pages 663–695, 2019. 9

[CPF$^+$17] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–806, 2017. 2

[CXZC24] Wenjing Chen, Shuo Xing, Samson Zhou, and Victoria G. Crawford. Fair submodular cover. *CoRR*, abs/2407.04804, 2024. 25

[DMM06a] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-based methods. In *9th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX) and 10th International Workshop on Randomization and Computation (RANDOM)*, pages 316–326, 2006. 8, 9

[DMM06b] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-row-based methods. In *Algorithms - ESA 2006, 14th Annual European Symposium, Proceedings*, pages 304–314, 2006. 8, 9

[DMV22] Zhen Dai, Yury Makarychev, and Ali Vakilian. Fair representation clustering with several protected classes. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 814–823, 2022. 25

[DTV11] Amit Deshpande, Madhur Tulsiani, and Nisheeth K. Vishnoi. Algorithms and hardness for subspace approximation. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 482–496, 2011. 3, 11

[EBTD20] Seyed A. Esmaeili, Brian Brubach, Leonidas Tsepenekas, and John Dickerson. Probabilistic fair clustering. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*, 2020. 25

[EY36] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. 7

[Fos53] Frederic G Foster. On the stochastic matrices associated with certain queuing processes. *The Annals of Mathematical Statistics*, 24(3):355–360, 1953. 9

[GJ23] Dishant Goyal and Ragesh Jaiswal. Tight FPT approximation for socially fair clustering. *Information Processing Letters*, 182:106383, 2023. 6

[GSV21] Mehrdad Ghadiri, Samira Samadi, and Santosh Vempala. Socially fair k-means clustering. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 438–448, 2021. 6

[GSV22] Mehrdad Ghadiri, Mohit Singh, and Santosh S Vempala. Constant-factor approximation algorithms for socially fair k-clustering. *arXiv preprint arXiv:2206.11210*, 2022. 6

[HJS$^+$22] Baihe Huang, Shunhua Jiang, Zhao Song, Runzhou Tao, and Ruizhe Zhang. Solving SDP faster: A robust IPM framework and efficient implementation. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 233–244, 2022. 35

[HMV23] Sedjro Salomon Hotegni, Sepideh Mahabadi, and Ali Vakilian. Approximation algorithms for fair range clustering. In *International Conference on Machine Learning*, pages 13270–13284. PMLR, 2023. 25

[HPS16] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, pages 3315–3323, 2016. 2

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. 3, 11

[JKL$^+$20a] Haotian Jiang, Tarun Kathuria, Yin Tat Lee, Swati Padmanabhan, and Zhao Song. A faster interior point method for semidefinite programming. In *FOCS*, 2020. 35

[JKL20b] Christopher Jung, Sampath Kannan, and Neil Lutz. Service in your neighborhood: Fairness in center location. In Aaron Roth, editor, *1st Symposium on Foundations of Responsible Computing, FORC 2020*, pages 5:1–5:15, 2020. 25

[JLL$^+$21] Shuli Jiang, Dennis Li, Irene Mengze Li, Arvind V. Mahankali, and David P. Woodruff. Streaming and distributed algorithms for robust column subset selection. In *Proceedings of the 38th International Conference on Machine Learning, ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 4971–4981, 2021. 25

[JLS22] Arun Jambulapati, Yang P. Liu, and Aaron Sidford. Improved iteration complexities for overconstrained p-norm regression. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 529–542, 2022. 10

[JLSW20] Haotian Jiang, Yin Tat Lee, Zhao Song, and Sam Chiu-wai Wong. An improved cutting plane method for convex optimization, convex-concave games, and its applications. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 944–953, 2020. 34

[JNN20] Matthew Jones, Huy Nguyen, and Thy Nguyen. Fair k-centers via maximum matching. In *International conference on machine learning*, pages 4940–4949. PMLR, 2020. 25

[JPT13] Gabriela Jeronimo, Daniel Perrucci, and Elias P. Tsigaridas. On the minimum of a polynomial function on a basic closed semialgebraic set and applications. *SIAM J. Optim.*, 23(1):241–255, 2013. 13

[JSWZ21] Shunhua Jiang, Zhao Song, Omri Weinstein, and Hengjie Zhang. A faster algorithm for solving general lps. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 823–832, 2021. 35

[KAM19] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-center clustering for data summarization. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, pages 3448–3457, 2019. 25

[KDRZ23] Matthäus Kleindessner, Michele Donini, Chris Russell, and Muhammad Bilal Zafar. Efficient fair pca for fair representation learning. In *International Conference on Artificial Intelligence and Statistics*, pages 5250–5270. PMLR, 2023. 6

[KLL$^+$18] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. Human decisions and machine predictions. *The quarterly journal of economics*, 133(1):237–293, 2018. 1

[KMM15] Matthew Kay, Cynthia Matuszek, and Sean A. Munson. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI*, pages 3819–3828, 2015. 2

[KMR17] Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference, ITCS*, pages 43:1–43:23, 2017. 2

[KPRW19] Ravi Kumar, Rina Panigrahy, Ali Rahimi, and David P. Woodruff. Faster algorithms for binary matrix factorization. In *Proceedings of the 36th International Conference on Machine Learning, ICML*, pages 3551–3559, 2019. 12

[Lee16] Yin Tat Lee. Faster algorithms for convex and combinatorial optimization, 2016. 10

[LKO$^+$22] Junghyun Lee, Gwangsu Kim, Mahbod Olfat, Mark Hasegawa-Johnson, and Chang D Yoo. Fast and efficient mmd-based fair pca via optimization over stiefel manifold, 2022. 6

[LSW15] Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS*, pages 1049–1065, 2015. 34

[Mag10] Malik Magdon-Ismail. Row sampling for matrix algorithms via a non-commutative bernstein bound. *CoRR*, abs/1008.0587, 2010. 8, 9

[MMM$^+$22] Raphael A. Meyer, Cameron Musco, Christopher Musco, David P. Woodruff, and Samson Zhou. Fast regression for structured inputs. In *The Tenth International Conference on Learning Representations, ICLR*, 2022. 9

[MMM$^+$23] Raphael A. Meyer, Cameron Musco, Christopher Musco, David P. Woodruff, and Samson Zhou. Near-linear sample complexity for L_p polynomial regression. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 3959–4025, 2023. 9

[MMWY22] Cameron Musco, Christopher Musco, David P. Woodruff, and Taisuke Yasuda. Active linear regression for L_p norms and beyond. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 744–753, 2022. 9

[MOT24] Antonis Matakos, Bruno Ordozgoiti, and Suhas Thejaswi. Fair column subset selection. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2189–2199, 2024. 6

[MV20] Sepideh Mahabadi and Ali Vakilian. Individual fairness for k-clustering. In *Proceedings of the 37th International Conference on Machine Learning, ICML*, pages 6586–6596, 2020. 25

[MV21] Yury Makarychev and Ali Vakilian. Approximation algorithms for socially fair clustering. In *Conference on Learning Theory*, pages 3246–3264. PMLR, 2021. 6

[NC21] Maryam Negahbani and Deeparnab Chakrabarty. Better algorithms for individually fair k-clustering. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS*, pages 13340–13351, 2021. 25

[NNJ22] Huy Lê Nguyen, Thy Nguyen, and Matthew Jones. Fair range k-center. *arXiv preprint arXiv:2207.11337*, 2022. 25

[OA19] Matt Olfat and Anil Aswani. Convex formulations for fair principal component analysis, 2019. 6

[PPP21] Aditya Parulekar, Advait Parulekar, and Eric Price. L_1 regression with lewis weights subsampling. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 49:1–49:21, 2021. 9

[PVZ17] Grigoris Paouris, Petros Valettas, and Joel Zinn. Random version of dvoretzky's theorem in ℓ_p^n. *Stochastic Processes and their Applications*, 127(10):3187–3227, 2017. 17

[Ren92a] James Renegar. On the computational complexity and geometry of the first-order theory of the reals. part i: Introduction. preliminaries. the geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. *Journal of symbolic computation*, 13(3):255–299, 1992. 13

[Ren92b] James Renegar. On the computational complexity and geometry of the first-order theory of the reals. part ii: The general decision problem. preliminaries for quantifier elimination. *Journal of Symbolic Computation*, 13(3):301–327, 1992. 13

[RS18] Clemens Rösner and Melanie Schmidt. Privacy preserving clustering with constraints. In *45th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 96:1–96:14, 2018. 25

[RSW16] Ilya P. Razenshteyn, Zhao Song, and David P. Woodruff. Weighted low rank approximations with provable guarantees. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 250–263, 2016. 12

[STM$^+$18] Samira Samadi, Uthaipon Tantipongpipat, Jamie H Morgenstern, Mohit Singh, and Santosh Vempala. The price of fair pca: One extra dimension. *Advances in neural information processing systems*, 31, 2018. 5, 22

[SW18] Christian Sohler and David P. Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 802–813, 2018. 17

[TGOO22] Suhas Thejaswi, Ameet Gadekar, Bruno Ordozgoiti, and Michal Osadnik. Clustering with fair-center representation: Parameterized approximation algorithms and heuristics. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1749–1759, 2022. 25

[TSS$^+$19] Uthaipon Tantipongpipat, Samira Samadi, Mohit Singh, Jamie H Morgenstern, and Santosh Vempala. Multi-criteria dimensionality reduction with applications to fairness. *Advances in neural information processing systems*, 32, 2019. 6

[VVWZ23] Ameya Velingker, Maximilian Vötsch, David P. Woodruff, and Samson Zhou. Fast $(1+\epsilon)$-approximation algorithms for binary matrix factorization. In *International Conference on Machine Learning, ICML*, pages 34952–34977, 2023. 12

[VY22] Ali Vakilian and Mustafa Yalciner. Improved approximation algorithms for individually fair clustering. In *International conference on artificial intelligence and statistics*, pages 8758–8779. PMLR, 2022. 25

[Woo14] David P. Woodruff. Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2):1–157, 2014. 8, 9

[WY23a] David P. Woodruff and Taisuke Yasuda. New subset selection algorithms for low rank approximation: Offline and online. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC*, pages 1802–1813, 2023. 4, 16, 17, 25

[WY23b] David P. Woodruff and Taisuke Yasuda. Online lewis weight sampling. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 4622–4666, 2023. 10

[Yeh16] I-Cheng Yeh. Default of Credit Card Clients. UCI Machine Learning Repository, 2016. DOI: https://doi.org/10.24432/C55S3H. 22

[YL09] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications*, 36(2):2473–2480, 2009. 5, 22

A Socially Fair Regression

In this section, we present simple and intuitive algorithms for fair regression. Recall that in the classical regression problem, the input is a data matrix $\mathbf{A} \in \mathbb{R}^{n \times d}$ and a label matrix $\mathbf{b} \in \mathbb{R}^n$, and the task is to determine the hidden vector $\mathbf{x} \in \mathbb{R}^d$ that minimizes $\mathcal{L}(\mathbf{A}\mathbf{x} - \mathbf{b})$ for some loss function $\mathcal{L}$. The vector $\mathbf{x}$ can then be subsequently used to label future observations $\mathbf{v}$ through the operation $\langle \mathbf{v}, \mathbf{x} \rangle$. Hence, regression analysis is frequently used in machine learning to infer causal relationships between the independent and dependent variables, and thus also used for prediction and forecasting.

Regression and low-rank approximation share key connections and thus are often studied together, e.g., a number of sketching techniques can often be applied to both problems, resulting in significant runtime improvements due to dimensionality reduction. See Section 1.3.1 for more information. Surprisingly, we show that socially fair regression and socially fair low-rank approximation exhibit drastically different complexities. Specifically, we show that while fair regression can be solved up to arbitrary accuracy in polynomial time for a wide variety of loss functions, even constant-factor approximation to fair low-rank approximation requires exponential time under certain standard complexity hypotheses. Let ℓ be the number of groups, $n_1, \ldots, n_\ell$ be positive integers and for each $i \in [\ell]$, let $\mathbf{A}^{(i)} \in \mathbb{R}^{n_i \times d}$ and $\mathbf{b}^{(i)} \in \mathbb{R}^{n_i}$. Then for a norm $\| \cdot \|$, we define the fair regression problem to be

$$\min_{\mathbf{x} \in \mathbb{R}^d} \max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{x} - \mathbf{b}^{(i)}\|.$$

We first show that the optimal solution to the standard regression minimization problem also admits a ℓ-approximation to the fair regression problem.

Theorem A.1. *The optimal solution to the standard regression problem that computes a vector $\widehat{\mathbf{x}} \in \mathbb{R}^d$ also satisfies*

$$\max_{i\in[\ell]} \|\mathbf{A}^{(i)}\widehat{\mathbf{x}} - \mathbf{b}^{(i)}\|_2 \le \ell \cdot \min_{\mathbf{x}\in\mathbb{R}^d} \max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{x} - \mathbf{b}^{(i)}\|_2,$$

i.e., the algorithm outputs a ℓ-approximation to the fair regression problem. For L_2 loss, the algorithm uses $\mathcal{O}\left(nd^{\omega-1}\right)$ runtime, where $n = n_1 + \ldots + n_\ell$ and ω is the matrix multiplication exponent.

Proof. Let $\widehat{\mathbf{x}} = \text{argmin}_{\mathbf{x}\in\mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|$, where $\mathbf{A} = \mathbf{A}^{(1)} \circ \ldots \circ \mathbf{A}^{(\ell)}$ and $\mathbf{b} = \mathbf{b}^{(1)} \circ \ldots \circ \mathbf{b}^{(\ell)}$. Let $\mathbf{x}^* = \text{argmin}_{\mathbf{x}\in\mathbb{R}^d} \max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{x} - \mathbf{b}^{(i)}\|$. Then by the optimality of $\widehat{\mathbf{x}}$, we have

$$\|\mathbf{A}\widehat{\mathbf{x}} - \mathbf{b}\| \le \|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|.$$

By triangle inequality,

$$\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\|_2 \le \sum_{i=1}^{\ell} \|\mathbf{A}^{(i)}\mathbf{x}^* - \mathbf{b}^{(i)}\| \le \ell \cdot \max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{x}^* - \mathbf{b}^{(i)}\|.$$

Therefore,

$$\|\mathbf{A}\widehat{\mathbf{x}} - \mathbf{b}\| \le \ell \cdot \max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{x}^* - \mathbf{b}^{(i)}\|,$$

so that $\widehat{\mathbf{x}}$ produces an ℓ-approximation to the fair regression problem.

Finally, note that for L_2 loss, the closed form solution of $\widehat{\mathbf{x}}$ is $\widehat{\mathbf{x}} = (\mathbf{A}^{(i)})^\dagger \mathbf{b}^{(i)}$ and can computed in runtime $\mathcal{O}\left(nd^{\omega-1}\right)$. □

More generally, we can apply the same principles to observe that for any norm $\|\cdot\|$, $\|\mathbf{A}\mathbf{x}^* - \mathbf{b}\| \le \sum_{i=1}^{\ell} \|\mathbf{A}^{(i)}\mathbf{x}^* - \mathbf{b}^{(i)}\| \le \ell \cdot \max_{i\in[\ell]} \|\mathbf{A}^{(i)}\mathbf{x}^* - \mathbf{b}^{(i)}\|$. Thus if $\|\cdot\|$ admits an efficient algorithm for regression, then $\|\cdot\|$ also admits an efficient ℓ-approximation algorithm for fair regression. See Algorithm 5 for reference.

We now give a general algorithm for achieving additive ε approximation to fair regression. [AAK+22] observed that every norm is convex, since $\|\lambda\mathbf{u} + (1-\lambda)\mathbf{v}\| \le \lambda\|\mathbf{u}\| + (1-\lambda)\|\mathbf{v}\|$ by triangle inequality. Therefore, the function $g(\mathbf{x}) := \max_{i\in[\ell]}\{\|\mathbf{A}^{(i)}\mathbf{x} - \mathbf{b}^{(i)}\|\}$ is convex because the maximum of convex functions is also a convex function. Hence, the objective $\min_{\mathbf{x}} g(\mathbf{x})$ is the minimization of a convex function and can be solved using standard tools in convex optimization. [AAK+22] leveraged this observation by applying projected stochastic gradient descent. Here we use a convex solvers based on separating hyperplanes. The resulting algorithm is quite simple and appears in Algorithm 6.

Algorithm 5 ℓ-approximation for Socially Fair Regression

```
Input: $\mathbf{A}^{(i)} \in \mathbb{R}^{n_i \times d}$, $\mathbf{b}^{(i)} \in \mathbb{R}^{n_i}$ for all $i \in [\ell]$
Output: $\ell$-approximation for fair regression
$\mathbf{A} \leftarrow \mathbf{A}^{(1)} \circ \ldots \circ \mathbf{A}^{(\ell)}$
$\mathbf{b} \leftarrow \mathbf{A}^{(1)} \circ \ldots \circ \mathbf{b}^{(\ell)}$
Return $\text{argmin}_{\mathbf{x}\in\mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|$
```

Algorithm 6 Algorithm for Socially Fair Regression

```
Input: $\mathbf{A}^{(i)} \in \mathbb{R}^{n_i \times d}$, $\mathbf{b}^{(i)} \in \mathbb{R}^{n_i}$ for all $i \in [\ell]$
Output: Optimal for socially fair regression
1: Use a convex solver to return $\text{argmin}_{\mathbf{x} \in \mathbb{R}^d} \max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{x} - \mathbf{b}^{(i)}\|$
```

We first require the following definition of a separation oracle.

Definition A.2 (Separation oracle)**.** *Given a set $K \subset \mathbb{R}^d$ and $\varepsilon > 0$, a separation oracle for K is a function that takes an input $x \in \mathbb{R}^d$, either outputs that x is in K or outputs a separating hyperplane, i.e., a half-space of the form $H := \{z \mid c^\top z \le c^\top x + b\} \supseteq K$ with $b \le \varepsilon \|c\|_2$ and $c \ne 0^d$.*

We next recall the following statement on the runtime of convex solvers given access to a separation oracle.

Theorem A.3 ([LSW15, JLSW20])**.** *Suppose there exists a set K that is contained in a box of radius R and a separation oracle that, given a point x and using time T, either outputs that x is in K or outputs a separating hyperplane. Then there exists an algorithm that either finds a point in K or proves that K does not contain a ball of radius ε. The algorithm uses $\mathcal{O}(dT \log \kappa) + d^3 \cdot \operatorname{polylog} \kappa$ runtime, for $\kappa = \frac{dR}{\varepsilon}$.*

We use the following reduction from the computation of subgradients to separation oracles, given by Lemma 38 in [LSW15].

Lemma A.4 ([LSW15])**.** *Given $\alpha > 0$, suppose K is a set defined by $\{x \in [-\Delta, \Delta]^d \mid \|Ax\| \le \alpha\}$, where $\Delta = \operatorname{poly}(n)$ and the subgradient of $\|Ax\|$ can be computed in $\operatorname{poly}(n, d)$ time. Then there exists a separation oracle for K that uses $\operatorname{poly}\left(\frac{nd}{\varepsilon}\right)$ time.*

Putting things together, we now have our main algorithm for fair regression:

Theorem A.5. *Let $n_1, \ldots, n_\ell$ be positive integers and for each $i \in [\ell]$, let $\mathbf{A}^{(i)} \in \mathbb{R}^{n_i \times d}$ and $\mathbf{b}^{(i)} \in \mathbb{R}^{n_i}$. Let $\Delta = \operatorname{poly}(n)$ for $n = n_1 + \ldots + n_\ell$ and let $\varepsilon \in (0, 1)$. For any norm $\|\cdot\|$ whose subgradient can be computed in $\operatorname{poly}(n, d)$ time, there exists an algorithm that outputs $\mathbf{x}^* \in [-\Delta, \Delta]^d$ such that $\max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{x}^* - \mathbf{b}^{(i)}\| \le \varepsilon + \min_{\mathbf{x} \in [-\Delta, \Delta]^d} \max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{x} - \mathbf{b}^{(i)}\|$. The algorithm uses $\operatorname{poly}\left(n, d, \frac{1}{\varepsilon}\right)$ runtime.*

Proof. Recall that every norm is convex, since $\|\lambda \mathbf{u} + (1 - \lambda)\mathbf{v}\| \le \lambda \|\mathbf{u}\| + (1 - \lambda)\|\mathbf{v}\|$ by triangle inequality. Therefore, the function $g(\mathbf{x}) := \max_{i \in [\ell]}\{\|\mathbf{A}^{(i)}\mathbf{x} - \mathbf{b}^{(i)}\|\}$ is convex because the maximum of convex functions is also a convex function. Hence, the objective $\min_{\mathbf{x}} g(\mathbf{x})$ is the minimization of a convex function and can be solved using a convex program. □

A.1 Algorithms for L_1 and L_2 Regression

We now give an efficient algorithm for $(1 + \varepsilon)$-approximation for fair L_1 regression using linear programs.

Definition A.6 (Linear program formulation for min-max L_1 formulation)**.** *Let $n = \sum_{i=1}^{\ell} n_i$. Given $\mathbf{A}^{(i)} \in \mathbb{R}^{n_i \times d}$, $\mathbf{b}^{(i)} \in \mathbb{R}^{n_i}$ and a parameter $L > 0$, the linear program formulation (with $d + n$ variables and $O(n)$ constraints) can be written as follows*

$$\min_{x \in \mathbb{R}^d, t \in \mathbb{R}^{\sum_{i=1}^{\ell} n_i}} \langle x, \mathbf{1}_d \rangle$$

$$
\begin{aligned}
\text{subject to } & (\mathbf{A}^{(i)}x - \mathbf{b}^{(i)})_j \leq t_{i,j} \cdot \mathbf{1}_d, && \forall i \in [\ell], \forall j \in [n_i] \\
& (\mathbf{A}^{(i)}x - \mathbf{b}^{(i)})_j \geq -t_{i,j} \cdot \mathbf{1}_d, && \forall i \in [\ell], \forall j \in [n_i] \\
& t_{i,j} \geq 0, && \forall i \in [\ell], \forall j \in [n_i] \\
& \sum_{j=1}^{n_i} t_{i,j} \leq L, && \forall i \in [\ell]
\end{aligned}
$$

Here $\mathbf{1}_d$ is a length-d where all the entries are ones.

Observation A.7. *The linear program in Definition A.6 has a feasible solution if and only if*

$$L \geq \min_{\mathbf{x} \in \mathbb{R}^d} \max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{x} - \mathbf{b}^{(i)}\|_1.$$

Theorem A.8. *There exists an algorithm that uses $\mathcal{O}\left((n+d)^\omega\right)$ runtime and outputs whether the linear program in Definition A.6 has a feasible solution. Here $\omega \approx 2.373$.*

Proof. The proof directly follows using linear programming solver [CLS19, JSWZ21] as a black-box. □

Given Theorem A.8 and the ℓ-approximation algorithm as a starting point, we can achieve a $(1+\varepsilon)$-approximation to fair L_1 regression using binary search in $\mathcal{O}\left(\frac{1}{\varepsilon} \log \ell\right)$ iterations.

Definition A.9 (Quadratic program formulation for min-max L_2 formulation)**.** *Given $\mathbf{A}^{(i)} \in \mathbb{R}^{n_i \times d}$, $\mathbf{b}^{(i)} \in \mathbb{R}^{n_i}$ and a parameter $L > 0$, the quadratic constraint quadratic program formulation can be written as follows*

$$
\begin{aligned}
& \min_{x \in \mathbb{R}^d} x^\top I_d x \\
& \text{subject to } x^\top (A^{(i)})^\top A^{(i)} x - 2\langle A^{(i)}x, b^{(i)} \rangle + \|b^{(i)}\|_2^2 \leq L, && \forall i \in [\ell]
\end{aligned}
$$

We now describe a similar approach for fair L_2 regression using a quadratic program.

Observation A.10. *The quadratically constrained quadratic program in Definition A.9 has a feasible solution if and only if*

$$L \geq \min_{\mathbf{x} \in \mathbb{R}^d} \max_{i \in [\ell]} \|\mathbf{A}^{(i)}\mathbf{x} - \mathbf{b}^{(i)}\|_2^2.$$

Theorem A.11. *There exists an algorithm that uses $\mathrm{poly}(n+d)$ runtime and outputs whether the quadratically constrained quadratic program in Definition A.9 has a feasible solution. Here $n = \sum_{i=1}^{\ell} n_i$. The polynomial factor is at least 4.*

Proof. This directly follows using semi-definite programming solver as a black-box [JKL+20a, HJS+22]. Note that in general, quadratic programming is NP-hard to solve, however the formulation we have, all the matrices are semi-definite matrix. Thus, it is straightforward to reduce it to an SDP and then run an SDP solver. □