

Fair Dimensionality Reduction and Iterative Rounding for SDPs

Jamie Morgenstern

Samira Samadi

Mohit Singh

Uthaipon (Tao) Tantipongpipat

Santosh Vempala

Georgia Institute of Technology

JAMIEEMMT.CS@GATECH.EDU

SSAMADI6@GATECH.EDU

MOHIT.SINGH@ISYE.GATECH.EDU

TAO@GATECH.EDU

VEMPALA@CC.GATECH.EDU

Abstract

We model *fair* dimensionality reduction as an optimization problem. A central example is *fair PCA* (Samadi et al., 2018a): the input data is divided into k groups, and the goal is to find a single d -dimensional representation for all groups for which the maximum variance (or minimum reconstruction error) is optimized for all groups in a fair (or balanced) manner, e.g., by maximizing the minimum variance over the k groups of the projection to a d -dimensional subspace. This problem was introduced by Samadi et al. (2018a) who gave a polynomial-time algorithm which, for $k = 2$ groups, returns a $(d + 1)$ -dimensional solution of value at least the best d -dimensional solution.

In this work, we give an exact polynomial-time algorithm for $k = 2$ groups. The result relies on extending results of Pataki (1998) regarding rank of extreme point solutions to semi-definite programs. This approach applies more generally to any monotone concave function of the individual group objectives. For $k > 2$ groups, our results generalize to give a $(d + \sqrt{2k + 0.25} - 1.5)$ -dimensional solution with objective value at least as good as the optimal d -dimensional solution for arbitrary k, d in polynomial time.

Using our extreme point characterization result for SDPs, we give an iterative rounding framework for general SDPs which generalizes the well-known iterative rounding approach for LPs. It returns low-rank solutions with bounded violation of constraints. As an application for the FAIR-PCA problem, we obtain a d -dimensional projection where the violation in the objective can be bounded additively in terms of the top $O(\sqrt{k})$ -singular values of the data matrices. We also give an exact polynomial-time algorithm for any fixed number of groups and target dimension, and very general objective function over the groups, via the algorithm of Grigoriev and Pasechnik (2005) for covering zeros of functions of quadratic maps. In contrast, when the number of groups is part of the input, even for target dimension $d = 1$, we show this problem is NP-hard.

Keywords: Fairness, Dimensionality Reduction, PCA, Approximation Algorithms, Semi-definite programming.

1. Introduction

Choosing a low-dimensional representation of a large, high-dimensional data set is a basic computational task with many applications, and is a core primitive for modern machine learning. Perhaps the most ubiquitous and effective of these in practice is *principal component analysis* (PCA), which finds a subspace that maximizes the squared lengths of data

points projected to the subspace (equivalently, minimizes the sum of squared distances, or regression error, to a d -dimensional subspace). When viewing the data as the rows of an $m \times n$ matrix A , the objective is to find an $n \times d$ projection matrix P that maximizes the Frobenius norm, $\|AP\|_F^2$.

Fairness and multi-objective optimization. In this age of large-dimensional data, PCA is an indispensable tool for data analysis and a common preprocessing step. Previous work (Samadi et al., 2018a) investigated the case when data falls into two or more groups (e.g., based on gender or education level) A_1, \dots, A_k . A single global objective for the entire data set need not result in a solution which has high fidelity for all groups. In practice, even when the groups have equal size, PCA often results in much higher reconstruction error for some groups. This suggests some necessary tradeoff between accurately representing each group’s data in a single lower-dimensional subspace. How might one redefine dimensionality reduction to produce projections which optimize different groups’ representation in a balanced way?

The definition of these groups need not be a partition; each group could be defined as a different weighting of the data set (rather than a subset, which is a 0/1 weighting). Framed this way, asking for balance or fairness of this optimization can be viewed as dealing with multiple competing objectives. One way to balance multiple objectives is to find a projection P that maximizes the minimum objective value over each of the groups (weightings), i.e.,

$$\max_{P: P^T P = I_d} \min_{1 \leq i \leq k} \|A_i P\|_F^2 = \langle A_i^T A_i, P P^T \rangle. \quad (\text{FAIR-PCA})$$

More generally, let \mathcal{P}_d denote the set of all $n \times d$ projection matrices P , i.e., matrices with d orthonormal columns. For each group A_i , we associate a function $f_i : \mathcal{P}_d \rightarrow \mathbb{R}$ that denotes the group’s objective value for a particular projection. For any $g : \mathbb{R}^k \rightarrow \mathbb{R}$, we define the (f, g) -fair dimensionality reduction problem as finding a d -dimensional projection P which optimizes

$$\max_{P \in \mathcal{P}_d} g(f_1(P), f_2(P), \dots, f_k(P)). \quad (\text{FAIR-DIMENSION-REDUCTION})$$

In the above example of max-min fair PCA, g is simply the min function and $f_i(P) = \|A_i P\|^2$ is the total squared norm of the projection of vectors in A_i . Other examples include: defining each f_i as the average squared norm of the projections rather than the total, or the marginal variance — the difference in total squared norm when using P rather than the best possible projection for that group. (Note that for PCA, max variance and min reconstruction error are the same objective.) One could also choose the product function $g(y_1, \dots, y_k) = \prod_i y_i$ for the accumulating function g . This is also a natural choice, famously introduced in Nash’s solution to the bargaining problem. This framework can also describe the p th power mean of the projections, e.g. $f_i(P) = \|A_i P\|^2$ and $g(y_1, \dots, y_k) = \left(\sum_{i \in [k]} y_i^{p/2} \right)^{1/p}$.

The appropriate weighting of k objectives often depends on the context and application. We define *fair dimensionality reduction* to be the general problem with functions f_1, \dots, f_k, g . The central motivating questions of this paper are the following:

- What is the complexity of FAIR-PCA ?

- More generally, what is the complexity of FAIR-DIMENSION-REDUCTION ?

Framed another way, we ask whether these “fair” optimization problems force us to incur substantial computational cost? In [Samadi et al. \(2018a\)](#), we considered FAIR-PCA and showed that the natural semi-definite relaxation of the problem can be used to find a projection to dimension $d + k - 1$ whose cost is at most that of the optimal d -dimensional projection. For $k = 2$ groups, this is an increase of 1 in the dimension (as opposed to the naïve bound of $2d$, by simply taking the span of the optimal d -dimensional subspaces for the two groups). However, the computational complexity of exactly solving FAIR-PCA remained open.

1.1. Results and Techniques

Let us first focus on FAIR-PCA for ease of exposition. The problem can be reformulated as the following mathematical program where we denote PP^T by X . A natural approach to solving this problem is to consider the SDP relaxation obtained by relaxing the rank constraint to a bound on the trace.

Exact Fair-PCA	SDP Relaxation of Fair-PCA
$\begin{aligned} &\max z \\ &\langle A_i^T A_i, X \rangle \geq z \quad i \in \{1, \dots, k\} \\ &\text{rank}(X) \leq d \\ &0 \preceq X \preceq I \end{aligned}$	$\begin{aligned} &\max z \\ &\langle A_i^T A_i, X \rangle \geq z \quad i \in \{1, \dots, k\} \\ &\text{tr}(X) \leq d \\ &0 \preceq X \preceq I \end{aligned}$

Our first main result is that the SDP relaxation is exact when there are *two* groups. Thus finding an extreme point of this SDP gives an exact algorithm for FAIR-PCA for two groups. Previously, only approximation algorithms were known for this problem.

Theorem 1 *Any optimal extreme point solution to the SDP relaxation for FAIR-PCA with two groups has rank at most d . Therefore, 2-group FAIR-PCA can be solved in polynomial time.*

Our results also hold for the FAIR-DIMENSION-REDUCTION when g is monotone nondecreasing in any one coordinate and concave, and each f_i is an affine function of PP^T (and thus a special case of a quadratic function in P).

Theorem 2 *There is a polynomial time algorithm for 2-group FAIR-DIMENSION-REDUCTION problem when g is concave and monotone nondecreasing for at least one of its two arguments, and each f_i is linear in PP^T , i.e., $f_i(P) = \langle B_i, PP^T \rangle$ for some matrix $B_i(A)$.*

As indicated in the theorem, the core idea is that extreme-point solutions of the SDP in fact have rank d , not just trace equal to d .

For $k > 2$, the SDP need not recover a rank d solution. In fact, the SDP may be inexact even for $k = 3$ (see Section 7). Nonetheless, we show that we can bound the rank of a solution to the SDP and obtain the following result. We state it for FAIR-PCA, though the same bound holds for FAIR-DIMENSION-REDUCTION under the same assumptions as in Theorem 1. Note that this result generalizes Theorem 1.

Theorem 3 *For any concave g that is monotone nondecreasing in at least one of its arguments, there exists a polynomial time algorithm for FAIR-PCA with k groups that returns a $d + \left\lfloor \sqrt{2k + \frac{1}{4}} - \frac{3}{2} \right\rfloor$ -dimensional embedding whose fair objective value is at least that of the optimal d -dimensional embedding. If g is only concave, then the dimension returned increases by 1.*

This strictly improves and generalizes the bound of $d + k - 1$ for FAIR-PCA from Samadi et al. (2018a). Moreover, if the dimensionality of the solution is a hard constraint, instead of tolerating $s = O(\sqrt{k})$ extra dimension in the solution, one may solve FAIR-PCA for target dimension $d - s$ to guarantee a solution of rank at most d . Thus, we obtain an approximation algorithm for FAIR-PCA of factor $1 - \frac{O(\sqrt{k})}{d}$. This is stated formally in Section 4.

We now focus our attention to the marginal loss function. This measures the maximum over the groups of the difference between the variance of a common solution for the k groups and an optimal solution for an individual group (“the marginal cost of sharing a common subspace”). For this problem, the above scaling method could substantially harm the objective value, since the target function is nonlinear, namely $\|A_i P\|^2 - \text{OPT}_i$ for each group i . In Section 3, we develop a general iterative rounding framework for SDPs with eigenvalue upper bounds and k other linear constraints. This algorithm gives a solution of desired rank that violates each constraint by a bounded amount. The precise statement is Theorem 7. It implies that for FAIR-PCA with marginal loss as the objective the additive error is

$$\Delta(\mathcal{A}) := \max_{S \subseteq [m]} \sum_{i=1}^{\lfloor \sqrt{2|S|+1} \rfloor} \sigma_i(A_S)$$

where $A_S = \frac{1}{|S|} \sum_{i \in S} A_i$.

It is natural to ask whether FAIR-PCA is NP-hard to solve exactly. The following result implies that it is, even for target dimension $d = 1$.

Theorem 4 *The max-min FAIR-PCA problem for target dimension $d = 1$ is NP-hard when the number of groups k is part of the input.*

This raises the question of the complexity for constant $k \geq 3$ groups. An alternative view of our SDP-based algorithm for $k = 2$ is via the S-lemma (Yakubovich, 1971, 1997), which shows that for two quadratic constraints over the unit sphere, there is a polynomial-time algorithm. We refer the reader to Pólik and Terlaky (2007); Ben-Tal and Nemirovski (2001) for the various formulations of the S-lemma and applications in control theory, optimization, geometry, portfolio management and statistics. Our proof of Theorem 1 effectively shows that the S-lemma can be adapted to our setting by incorporating an upper bound on the eigenvalues and still maintaining polynomial time solvability.

For k groups, we would have k constraints, one for each group, plus the eigenvalue constraint and the trace constraint; now the tractability of the problem is far from clear. In fact, as we show in Section 7, the SDP has an integrality gap even for $k = 3, d = 1$. We therefore consider an approach beyond SDPs, to one that involves solving non-convex problems. Thanks to the powerful algorithmic theory of quadratic maps, developed by Grigoriev and Pasechnik (2005), it is polynomial-time solvable to check feasibility of a set of quadratic constraints for any fixed k . As we discuss next, their algorithm can check for zeros of a function of a set of k quadratic functions, and can be used to optimize the function. Using this result, we show that for $d = k = O(1)$, there is a polynomial-time algorithm for rather general functions g of the values of individual groups.

Theorem 5 *Let the fairness objective be $g : \mathbb{R}^k \rightarrow \mathbb{R}$ where g is a degree ℓ polynomial in some computable subring of \mathbb{R}^k and each f_i is quadratic for $1 \leq i \leq k$. Then there is an algorithm to solve the fair dimensionality reduction problem in time $(\ell d n)^{O(k+d^2)}$.*

By choosing g to be the product polynomial over the usual $(\times, +)$ ring or the min function which is degree k in the $(\min, +)$ ring, this applies to the variants of FAIR-PCA discussed above and various other problems.

SDP extreme points. For $k = 2$, the underlying structural property we show is that extreme point solutions of the SDP have rank exactly d . First, for $k = d = 1$, this is the largest eigenvalue problem, since the maximum obtained by a matrix of trace equal to 1 can also be obtained by one of the extreme points in the convex decomposition of this matrix. This extends to trace equal to any d , i.e., the optimal solution must be given by the top k eigenvectors of $A^T A$. Second, without the eigenvalue bound, for any SDP with k constraints, there is an upper bound on the rank of any extreme point, of $O(\sqrt{k})$, a seminal result of Pataki (1998) (see also Barvinok (1995)). However, we cannot apply this directly as we have the eigenvalue upper bound constraint. The complication here is that we have to take into account the constraint $X \preceq I$ without increasing the rank.

Theorem 6 *Let C and A_1, \dots, A_m be $n \times n$ real matrices, $d \leq n$, and $b_1, \dots, b_m \in \mathbb{R}$. Suppose the semi-definite program $\text{SDP}(\text{I})$:*

$$\min \langle C, X \rangle \text{ subject to} \tag{1}$$

$$\langle A_i, X \rangle \triangleleft_i b_i \quad \forall 1 \leq i \leq m \tag{2}$$

$$\text{tr}(X) \leq d \tag{3}$$

$$0 \preceq X \preceq I_n \tag{4}$$

where $\triangleleft_i \in \{\leq, \geq, =\}$, has a nonempty feasible set. Then, all extreme optimal solutions X^* to $\text{SDP}(\text{I})$ have rank at most $r^* := d + \left\lfloor \sqrt{2m + \frac{9}{4} - \frac{3}{2}} \right\rfloor$. Moreover, given a feasible optimal solution, an extreme optimal solution can be found in polynomial time.

To prove the theorem, we extend Pataki (1998)'s characterization of rank of SDP extreme points with minimal loss in the rank. We show that the constraints $0 \preceq X \preceq I$ can be interpreted as a generalization of restricting variables to lie between 0 and 1 in the case of linear programming relaxations. From a technical perspective, our results give new insights

into structural properties of extreme points of semi-definite programs and more general convex programs. Our result adds to a rather short list of algorithmic problems that utilize properties of extreme points of a semi-definite relaxation.

SDP Iterative Rounding. Using Theorem 6, we extend the iterative rounding framework for linear programs (see [Lau et al. \(2011\)](#) and references therein) to semi-definite programs, where the 0, 1 constraints are generalized to eigenvalue bounds. The algorithm has a remarkably similar flavor. In each iteration, we fix the subspaces spanned by eigenvectors with 0 and 1 eigenvalues, and argue that one of the constraints can be dropped while bounding the total violation in the constraint over the course of the algorithm. While this applies directly to the FAIR-PCA problem, in fact is a general statement for SDPs, which we give below.

Let $\mathcal{A} = \{A_1, \dots, A_m\}$ be a collection of $n \times n$ matrices. For any set $S \subseteq \{1, \dots, m\}$, let $\sigma_i(S)$ the i^{th} largest singular of the average of matrices $\frac{1}{|S|} \sum_{i \in S} A_i$. We let

$$\Delta(\mathcal{A}) := \max_{S \subseteq [m]} \sum_{i=1}^{\lfloor \sqrt{2|S|}+1 \rfloor} \sigma_i(S).$$

Theorem 7 *Let C be a $n \times n$ matrix and $\mathcal{A} = \{A_1, \dots, A_m\}$ be a collection of $n \times n$ real matrices, $d \leq n$, and $b_1, \dots, b_m \in \mathbb{R}$. Suppose the semi-definite program SDP:*

$$\begin{aligned} \min \langle C, X \rangle \quad & \text{subject to} \\ \langle A_i, X \rangle & \geq b_i \quad \forall 1 \leq i \leq m \\ \text{tr}(X) & \leq d \\ 0 \preceq X & \preceq I_n \end{aligned}$$

has a nonempty feasible set and let X^ denote an optimal solution. The Algorithm ITERATIVE-SDP (see Figure 1) returns a matrix \tilde{X} such that*

1. *rank of \tilde{X} is at most d ,*
2. *$\langle C, \tilde{X} \rangle \leq \langle C, X^* \rangle$, and*
3. *$\langle A_i, \tilde{X} \rangle \geq b_i - \Delta(\mathcal{A})$ for each $1 \leq i \leq m$.*

1.2. Related Work

With the growing use of machine learning algorithms in automated decision making, researchers have raised concerns about the bias that these algorithms might produce in the outcomes ([Angwin et al., 2018](#); [Kay et al., 2015](#); [Buolamwini and Gebru, 2018](#); [Sweeney, 2013](#)). This has resulted in a wide range of studies focusing on detecting and eliminating sources of unfairness in different stages of a decision-making process, where most of this work has focused either on biased data or on algorithms producing biased outcomes. In this regard, studying fairness for dimensionality reduction techniques focuses on a more subtle source of bias in ML applications, which may or may not be used in any particular

decision-making process. When PCA is used as a preprocessing step for decision making, it can inadvertently erase critically useful information about some populations. Even when it is used merely to visualize data, the erasure of variance for some populations raises concerns of representational bias (Crawford, 2017).

Principal Component Analysis (PCA) (Pearson, 1901; Jolliffe, 1986; Hotelling, 1933) is widely used as a preprocessing step to reduce the computational burden and/or to facilitate data summarization (Raychaudhuri et al., 1999; Iezzoni and Pritts, 1991). Samadi et al. (2018a) observed that vanilla PCA can inadvertently choose a low dimensional representation of the data which depicts different populations with different fidelities. As the result, vanilla PCA itself can be a source of unfairness in the data representation step and they suggest replacing it with the Fair PCA algorithm in applications (Samadi et al., 2018b).

As mentioned earlier, Pataki (1998) (see also Barvinok (1995)) showed low rank solutions to semi-definite programs with small number of affine constraints can be obtained efficiently. We also refer the reader to survey by Lemon et al. (2016) for more details. Closely related to low rank SDP solutions is the S-lemma (Yakubovich, 1971, 1997) and we refer the reader to the survey by Pólik and Terlaky (2007). We also remark that methods based on Johnson-Lindenstrauss lemma can also be applied to obtain bi-criteria results for FAIR-PCA problem. For example, So et al. (2008) give algorithms that give low rank solutions for SDPs with affine constraints without the upper bound on eigenvalues. Here we have focused on single criteria setting, with violation either in the number of dimensions or the objective but not both. Extreme point solutions to linear programming have played an important role in design of approximation algorithms (Lenstra et al., 1990). Iterative rounding method for linear programming, based on the extreme point solutions, has been a highly successful technique starting with the work of Jain (2001). Restricting a feasible region of certain SDPs relaxations with low-rank constraints has been shown to avoid spurious local optima (Bandeira et al., 2016) and reduce the runtime due to known heuristics and analysis (Burer and Monteiro, 2003, 2005; Boumal et al., 2016). We refer the reader to Lau et al. (2011) for details on the topic and applications.

A closely related area, especially to FAIR-DIMENSION-REDUCTION problem, is multi-objective optimization which has a vast literature. We refer the reader to Deb (2014) and references therein. We also remark that properties of extreme point solutions of linear programs (Ravi and Goemans, 1996; Grandoni et al., 2014) have also been utilized to obtain approximation algorithms to multi-objective problems. For semi-definite programming based methods, the closest works are on simultaneous max-cut (Bhangale et al., 2015, 2018) that utilize sum of squares hierarchy to obtain improved approximation algorithms.

2. Low-rank Solutions of Fair-Dimension-Reduction

In this section, we show that all extreme solutions of SDP relaxation of FAIR-DIMENSION-REDUCTION have low rank, proving Theorem 1-3. Before we state the results, we make following assumptions. In this section, we let $g : \mathbb{R}^k \rightarrow \mathbb{R}$ be a concave function which is monotonic in at least one coordinate, and mildly assume that g can be accessed with a polynomial-time subgradient oracle and is polynomially bounded by its input. We are explicitly given functions f_1, f_2, \dots, f_k which are affine in PP^T , i.e. we are given real $n \times n$ matrices B_1, \dots, B_k and constants $\alpha_1, \alpha_2, \dots, \alpha_k \in \mathbb{R}$ and $f_i(P) = \langle B_i, PP^T \rangle + \alpha_i$.

We assume g to be G -Lipschitz. For functions f_1, \dots, f_k, g that are L_1, \dots, L_k, G -Lipschitz, we define an ϵ -optimal solution to (f, g) -FAIR-DIMENSION-REDUCTION problem as a projection matrix $X \in \mathbb{R}^{n \times n}, 0 \preceq X \preceq I_n$ of rank d whose objective value is at most $G\epsilon \left(\sum_{i=1}^k L_i^2 \right)^{1/2}$ from the optimum. In the context where an optimization problem has affine constraints $F_i(X) \leq b_i$ where F_i is L_i Lipschitz, we also define ϵ -solution as a projection matrix $X \in \mathbb{R}^{n \times n}, 0 \preceq X \preceq I_n$ of rank d that violates i th affine constraints by at most ϵL_i . Note that the feasible region of the problem is implicitly bounded by the constraint $X \preceq I_n$.

For Section 2, the algorithm may involve solving an optimization under a matrix linear inequality, which may not give an answer representable in finite bits of computation. However, we give algorithms that return an ϵ -close solution whose running time depends polynomially on $\log \frac{1}{\epsilon}$ for any $\epsilon > 0$. This is standard for computational tractability in convex optimization (see, for example, in Ben-Tal and Nemirovski (2001)). Therefore, for ease of exposition, we omit the computational error dependent on this ϵ to obtain an ϵ -feasible and ϵ -optimal solution, and define polynomial running time as polynomial in n, k and $\log \frac{1}{\epsilon}$.

To prove Theorem 1-3, we first show that extreme point solutions in semi-definite cone under affine constraints and $X \preceq I$ have low rank. The statement builds on a result of Pataki (1998). We then apply our result to FAIR-DIMENSION-REDUCTION problem, which contains the FAIR-PCA problem. Finally, we show that existence of low-rank solution leads to an approximation algorithm to FAIR-PCA problem.

We first prove Theorem 6.

Proof [Theorem 6] Let X^* be an extreme point optimal solution to $\text{SDP}(\text{I})$. Suppose rank of X^* , say r , is more than r^* . Then we show a contradiction to the fact that X^* is extreme. Let $0 \leq l \leq r$ of the eigenvalues of X^* be equal to one. If $l \geq d$, then we have $l = r = d$ since $\text{tr}(X) \leq d$ and we are done. Thus we assume that $l \leq d - 1$. In that case, there exist matrices $Q_1 \in \mathbb{R}^{n \times r-l}, Q_2 \in \mathbb{R}^{n \times l}$ and a symmetric matrix $\Lambda \in \mathbb{R}^{(r-l) \times (r-l)}$ such that

$$X^* = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} \Lambda & 0 \\ 0 & I_l \end{pmatrix} \begin{pmatrix} Q_1 & Q_2 \end{pmatrix}^\top = Q_1 \Lambda Q_1^\top + Q_2 Q_2^\top$$

where $0 \prec \Lambda \prec I_{r-l}$, $Q_1^\top Q_1 = I_{r-l}$, $Q_2^\top Q_2 = I_l$, and that the columns of Q_1 and Q_2 are orthogonal, i.e. $Q = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix}$ has orthonormal columns.

Now, we have

$$\langle A_i, X^* \rangle = \langle A_i, Q_1 \Lambda Q_1^\top + Q_2 Q_2^\top \rangle = \langle Q_1^\top A_i Q_1, \Lambda \rangle + \langle A_i, Q_2 Q_2^\top \rangle$$

and

$$\text{tr}(X^*) = \langle Q_1^\top Q_1, \Lambda \rangle + \text{tr}(Q_2 Q_2^\top)$$

so that $\langle A_i, X^* \rangle$ and $\text{tr}(X^*)$ are linear in Λ .

Observe the set of $s \times s$ symmetric matrices forms a vector space of dimension $\frac{s(s+1)}{2}$ with the above inner product where we consider the matrices as long vectors. If $m + 1 < \frac{(r-l)(r-l+1)}{2}$ then there exists a $(r-l) \times (r-l)$ -symmetric matrix $\Delta \neq 0$ such that $\langle Q_1^\top A_i Q_1, \Delta \rangle = 0$ for each $1 \leq i \leq m$ and $\langle Q_1^\top Q_1, \Delta \rangle = 0$.

But then we claim that $Q_1(\Lambda \pm \delta \Delta)Q_1^\top + Q_2 Q_2^\top$ is feasible for small $\delta > 0$, which implies a contradiction to X^* being extreme. Indeed, it satisfies all the linear constraints

by construction of Δ . Thus it remains to check the eigenvalues of the newly constructed matrix. Observe that

$$Q_1(\Lambda \pm \delta\Delta)Q_1^\top + Q_2Q_2^\top = Q \begin{pmatrix} \Lambda \pm \delta\Delta & 0 \\ 0 & I_l \end{pmatrix} Q^\top$$

with orthonormal Q . Thus it is enough to consider the eigenvalues of $\begin{pmatrix} \Lambda \pm \delta\Delta & 0 \\ 0 & I_l \end{pmatrix}$.

Observe that eigenvalues of the above matrix are exactly l ones and eigenvalues of $\Lambda \pm \delta\Delta$. Since eigenvalues of Λ are bounded away from 0 and 1, one can find small δ such that the eigenvalue of $\Lambda \pm \delta\Delta$ are bounded away from 0 and 1 as well, so we are done. Therefore, we must have $m + 1 \geq \frac{(r-l)(r-l+1)}{2}$ which implies $r - l \leq -\frac{1}{2} + \sqrt{2m + \frac{9}{4}}$. By $l \leq d - 1$, we have $r \leq r^*$.

For the algorithmic version, given feasible \bar{X} , we iteratively reduce $r - l$ by at least one until $m + 1 \geq \frac{(r-l)(r-l+1)}{2}$. While $m + 1 < \frac{(r-l)(r-l+1)}{2}$, we obtain Δ by using Gaussian elimination. Now we want to find the correct value of $\pm\delta$ so that $\Lambda' = \Lambda \pm \delta\Delta$ takes one of the eigenvalues to zero or one. First, determine the sign of $\langle C, \Delta \rangle$ to find the correct sign to move Λ that keeps the objective non-increasing, say it is in the positive direction. Since the set of feasible X is convex and bounded, the ray $f(t) = Q_1(\Lambda + t\Delta)Q_1^\top + Q_2Q_2^\top, t \geq 0$ intersects the boundary of feasible region at a unique $t' > 0$. Perform binary search for the correct value of t' and set $\delta = t'$ up to the desired accuracy. Since $\langle Q_1^\top A_i Q_1, \Delta \rangle = 0$ for each $1 \leq i \leq m$ and $\langle Q_1^\top Q_1, \Delta \rangle = 0$, the additional tight constraint from moving $\Lambda' \leftarrow \Lambda + \delta\Delta$ to the boundary of feasible region must be an eigenvalue constraint $0 \preceq X \preceq I_n$, i.e., at least one additional eigenvalue is now at 0 or 1, as desired. We apply eigenvalue decomposition to Λ' and update Q_1 accordingly, and repeat. \blacksquare

We also obtain the following corollary from the bound $r - l \leq -\frac{1}{2} + \sqrt{2m + \frac{9}{4}}$ in the proof of Theorem 6.

Corollary 8 *The number of fractional eigenvalues in any extreme point solution X to SDP(I) is bounded by $\sqrt{2m + \frac{9}{4}} - \frac{1}{2} \leq \lfloor \sqrt{2m} + 1 \rfloor$.*

We are now ready to state the main result of this section that we can find a low-rank solution for FAIR-DIMENSION-REDUCTION. Recall that \mathcal{P}_d is the set of all $n \times d$ projection matrices P , i.e., matrices with d orthonormal columns and the (f, g) -FAIR-DIMENSION-REDUCTION problem is to solve

$$\max_{P \in \mathcal{P}_d} g(f_1(P), f_2(P), \dots, f_k(P)) \quad (5)$$

Theorem 9 *There exists a polynomial-time algorithm to solve (f, g) -FAIR-DIMENSION-REDUCTION that returns a solution \hat{X} of rank at most $r^* := d + \left\lfloor \sqrt{2k + \frac{1}{4}} - \frac{3}{2} \right\rfloor$ whose objective value is at least that of the optimal d -dimensional embedding.*

Proof First, we write a relaxation of (5):

$$\max_{X \in \mathbb{R}^{n \times n}} g(\langle B_1, X \rangle + \alpha_1, \dots, \langle B_k, X \rangle + \alpha_k) \text{ subject to} \quad (6)$$

$$\text{tr}(X) \leq d \quad (7)$$

$$0 \preceq X \preceq I_n \quad (8)$$

Since $g(x)$ is concave in $x \in \mathbb{R}^k$ and $\langle B_i, X \rangle + \alpha_i$ is affine in $X \in \mathbb{R}^{n \times n}$, we have that g as a function of X is also concave in X . By assumptions on g , and the fact that the feasible set is convex and bounded, we can solve the convex program in polynomial time, e.g. by ellipsoid method, to obtain a (possibly high-rank) optimal solution $\bar{X} \in \mathbb{R}^{n \times n}$. (In the case that f_i is linear, the relaxation is also an SDP and may be solved faster in theory and practice). By assumptions on g , without loss of generality, we let g be nondecreasing in the first coordinate. To reduce the rank of \bar{X} , we consider an SDP(III):

$$\max_{X \in \mathbb{R}^{n \times n}} \langle B_1, X \rangle \text{ subject to} \quad (9)$$

$$\langle B_i, X \rangle = \langle B_i, \bar{X} \rangle \quad \forall 2 \leq i \leq k \quad (10)$$

$$\text{tr}(X) \leq d \quad (11)$$

$$0 \preceq X \preceq I_n \quad (12)$$

SDP(III) has a feasible solution \bar{X} of objective $\langle B_1, X \rangle$ and note that there are $k - 1$ constraints in (10). Hence, we can apply the algorithm in Theorem 6 with $m = k - 1$ to find an extreme solution X^* of SDP(III) of rank at most r^* . Since g is nondecreasing in $\langle B_1, X \rangle$, optimal solutions to SDP(III) gives objective value g at least the optimum of the relaxation (6)-(8), and hence at least the optimum of the original FAIR-DIMENSION-REDUCTION problem. \blacksquare

If the assumption that g is monotonic in at least one coordinate is dropped, Theorem 9 will hold with $r^* := d + \left\lfloor \sqrt{2k + \frac{9}{4}} - \frac{3}{2} \right\rfloor$ by indexing constraints (10) in SDP(III) for all groups instead of $k - 1$ groups.

Another way to state Theorem 9 is that the number of groups must reach $\frac{(s+1)(s+2)}{2}$ before additional s dimensions in the solution matrix P is required to achieve the optimal objective value. For $k = 2$, no additional dimension in the solution is necessary to attain the optimum. We state this fact as follows.

Corollary 10 *The (f, g) -FAIR-DIMENSION-REDUCTION problem on two groups can be solved in polynomial time.*

In particular, Corollary 10 applies to FAIR-PCA with two groups, proving Theorem 1.

3. Iterative Rounding Framework with Applications to Fair-PCA

In this section, we first prove Theorem 7.

We give an iterative rounding algorithm. The algorithm maintains three subspaces that are mutually orthogonal. Let F_0, F_1, F denote matrices whose columns form an orthonormal basis of these subspaces. We will also abuse notation and denote these matrices by sets of

vectors in their columns. We let the rank of F_0, F_1 and F be r_0, r_1 and r , respectively. We will ensure that $r_0 + r_1 + r = n$, i.e., vectors in F_0, F_1 and F span \mathbb{R}^n .

We initialize $F_0 = F_1 = \emptyset$ and $F = I_n$. Over iterations, we increase the subspaces spanned by columns of F_0 and F_1 and decrease F while maintaining pairwise orthogonality. The vectors in columns of F_1 will be eigenvectors of our final solution with eigenvalue 1. In each iteration, we project the constraint matrices A_i orthogonal to F_1 and F_0 . We will then formulate a residual SDP using columns of F as a basis and thus the new constructed matrices will have size $r \times r$. To readers familiar with the iterative rounding framework in linear programming, this generalizes the method of fixing certain variables to 0 or 1 and then formulating the residual problem. We also maintain a subset of constraints indexed by S where S is initialized to $\{1, \dots, m\}$.

The algorithm is specified in Figure 1. In each iteration, we formulate the following $\text{SDP}(r)$ with variables $X(r)$ which will be a $r \times r$ symmetric matrix. Recall r is the number of columns in F .

$$\begin{aligned} & \max \langle F^T C F, X(r) \rangle \\ & \langle F^T A_i F, X(r) \rangle \geq b_i - F_1^T A_i F_1 \quad i \in S \\ & \text{tr}(X) \leq d - \text{rank}(F_1) \\ & 0 \preceq X(r) \preceq I_r \end{aligned}$$

1. Initialize F_0, F_1 to be empty matrices and $F = I_n, S \leftarrow \{1, \dots, m\}$.
2. If the SDP is infeasible, declare infeasibility. Else,
3. While F is not the empty matrix.
 - (a) Solve $\text{SDP}(r)$ to obtain extreme point $X^*(r) = \sum_{j=1}^r \lambda_j v_j v_j^T$ where λ_j are the eigenvalues and $v_j \in \mathbb{R}^r$ are the corresponding eigenvectors.
 - (b) For any eigenvector v of $X^*(r)$ with eigenvalue 0, let $F_0 \leftarrow F_0 \cup \{Fv\}$.
 - (c) For any eigenvector v of $X^*(r)$ with eigenvalue 1, let $F_1 \leftarrow F_1 \cup \{Fv\}$.
 - (d) Let $X_f = \sum_{j: 0 < \lambda_j < 1} \lambda_j v_j v_j^T$. If there exists a constraint $i \in S$ such that $\langle F^T A_i F, X_f \rangle < \Delta(\mathcal{A})$, then $S \leftarrow S \setminus \{i\}$.
 - (e) For every eigenvector v of $X^*(r)$ with eigenvalue not equal to 0 or 1, consider the vectors Fv and form a matrix with these columns and use it as the new F .
4. Return $\tilde{X} = F_1 F_1^T$.

Figure 1: Iterative Rounding Algorithm ITERATIVE-SDP.

It is easy to see that the semi-definite program remains feasible over all iterations if SDP is declared feasible in the first iteration. Indeed the solution X_f defined at the end of any

iteration is a feasible solution to the next iteration. We also need the following standard claim.

Claim 1 *Let Y be a positive semi-definite matrix such that $Y \preceq I$ with $\text{tr}(Y) \leq l$. Let B be real matrix of the same size as Y and let $\lambda_i(B)$ denote the i^{th} largest singular value of B . Then*

$$\langle B, Y \rangle \leq \sum_{i=1}^l \lambda_i(B).$$

The following result follows from Corollary 8 and Claim 1. Recall that

$$\Delta(\mathcal{A}) := \max_{S \subseteq [m]} \sum_{i=1}^{\lfloor \sqrt{2|S|+1} \rfloor} \sigma_i(S).$$

where $\sigma_i(S)$ is the i^{th} largest singular value of $\frac{1}{|S|} \sum_{i \in S} A_i$.

We let Δ denote $\Delta(\mathcal{A})$ for the rest of the section.

Lemma 11 *Consider any extreme point solution $X(r)$ of $\text{SDP}(r)$ such that $\text{rank}(X(r)) > \text{tr}(X(r))$. Let $X(r) = \sum_{j=1}^r \lambda_j v_j v_j^T$ be its eigenvalue decomposition and $X_f = \sum_{0 < \lambda_j < 1} \lambda_j v_j v_j^T$. Then there exists a constraint i such that $\langle F^T A_i F, X_f \rangle < \Delta$.*

Proof Let $l = |S|$. From Corollary 8, it follows that number of fractional eigenvalues of $X(r)$ is at most $-\frac{1}{2} + \sqrt{2l + \frac{9}{4}} \leq \sqrt{2l} + 1$. Observe that $l > 0$ since $\text{rank}(X(r)) > \text{tr}(X(r))$. Thus $\text{rank}(X_f) \leq \sqrt{2l} + 1$. Moreover, $0 \preceq X_f \preceq I$, thus from Claim 1, we obtain that

$$\left\langle \sum_{j \in S} F^T A_j F, X_f \right\rangle \leq \sum_{i=1}^{\lfloor \sqrt{2l}+1 \rfloor} \sigma_i \left(\sum_{j \in S} F^T A_j F \right) \leq \sum_{i=1}^{\lfloor \sqrt{2l}+1 \rfloor} \sigma_i \left(\sum_{j \in S} A_j \right) \leq l \cdot \Delta$$

where the first inequality follows from Claim 1 and second inequality follows since the sum of top l singular values reduces after projection. But then we obtain, by averaging, that there exists $j \in S$ such that

$$\langle F^T A_j F, X_f \rangle < \frac{1}{l} \cdot l \Delta = \Delta$$

as claimed. ■

Now we complete the proof of Theorem 7. Observe that the algorithm always maintains that end of each iteration, trace of X_f plus the rank of F_1 is at most d . Thus at the end of the algorithm, the returned solution has rank at most d . Next, consider the solution $X = F_1 F_1^T + F X_f F^T$ over the course of the algorithm. Again, it is easy to see that the objective value is non-increasing over the iterations. This follows since X_f defined at the end of an iteration is a feasible solution to the next iteration.

Now we argue the violation in any constraint i . While the constraint i remains in the SDP, the solution $X = F_1 F_1^T + F X_f F^T$ satisfies

$$\begin{aligned}\langle A_i, X \rangle &= \langle A_i, F_1 F_1^T \rangle + \langle A_i, F X_f F^T \rangle \\ &= \langle A_i, F_1 F_1^T \rangle + \langle F^T A_i F, X_f \rangle \leq \langle A_i, F_1 F_1^T \rangle + b_i - \langle A_i, F_1 F_1^T \rangle = b_i.\end{aligned}$$

where the inequality again follows since X_f is feasible with the updated constraints.

When constraint i is removed it might be violated by a later solution. At this iteration, $\langle F^T A_i F, X_f \rangle \leq \Delta$. Thus, $\langle A_i, F_1 F_1^T \rangle \geq b_i - \Delta$. In the final solution this bound can only go up as F_1 might only become larger. This completes the proof of theorem.

Application to Fair-PCA . For the FAIR-PCA problem, iterative rounding recovers a rank- d solution whose variance goes down from the SDP solution by at most $\Delta(\{A_1^T A_1, \dots, A_k^T A_k\})$. While this is no better than what we get by scaling (Corollary 12) for the max variance objective function, when we consider the marginal loss, i.e., the difference between the variance of the common d -dimensional solution and the best d -dimensional solution for each group, then iterative rounding can be much better. The scaling solution guarantee relies on the max-variance being a concave function and for the marginal loss, the loss for each group could go up proportional to the *largest* max variance (largest sum of top k singular values over the groups). With iterative rounding applied to the SDP solution, the loss Δ is the sum of only $O(\sqrt{k})$ singular values of the average of some subset of data matrices, so it can be better by as much as a factor of \sqrt{k} .

4. Approximation Algorithm for Fair-PCA

Recall that we require $s := \left\lfloor \sqrt{2k + \frac{1}{4}} - \frac{3}{2} \right\rfloor$ additional dimensions for the projection to achieve the optimal objective. One way to ensure that the algorithm outputs d -dimensional projection is to solve the problem in lower target dimension $d - s$, then apply the rounding described in Section 2. The relationship of objectives between problems with target dimension $d - s$ and d is at most $\frac{d-s}{d}$ factor apart for FAIR-PCA problem because the objective scales linearly with P , giving an approximation guarantee of $1 - \frac{s}{d}$. Recall that given A_1, \dots, A_k , FAIR-PCA problem is to solve

$$\max_{P: P^T P = I_d} \min_{1 \leq i \leq k} \|A_i P\|_F^2 = \langle A_i^T A_i, P P^T \rangle$$

We state the approximation guarantee and the algorithm formally as follows.

Corollary 12 *Let A_1, \dots, A_k be data sets of k groups and suppose $s := \left\lfloor \sqrt{2k + \frac{1}{4}} - \frac{3}{2} \right\rfloor < d$. Then there exists a polynomial-time approximation algorithm of factor $1 - \frac{s}{d} = 1 - \frac{O(\sqrt{k})}{d}$ to FAIR-PCA problem.*

Proof We find an extreme solution X^* of the FAIR-PCA problem of finding a projection from n to $d - s$ target dimensions. By Theorem 9, the rank of X^* is at most d .

Denote OPT_d, X_d^* the optimal value and an optimal solution to FAIR-PCA with target dimension d . Note that $\frac{d-s}{d} X_d^*$ is a feasible solution to FAIR-PCA relaxation on target dimension $d - s$ which is at least $\frac{d-s}{d} \text{OPT}_d$ because the objective scales linearly with X .

Therefore, the optimal FAIR-PCA relaxation of target dimension $d - s$ attains optimum at least $\frac{d-s}{d}\text{OPT}_d$, giving $(1 - \frac{s}{d})$ -approximation ratio. \blacksquare

5. Polynomial Time Algorithm for Fixed Number of Groups

Functions of quadratic maps. We briefly summarize the approach of Grigoriev and Pasechnik (2005). Let $f_1, \dots, f_k : \mathbb{R}^n \rightarrow \mathbb{R}$ be real-valued quadratic functions in n variables. Let $p : \mathbb{R}^k \rightarrow \mathbb{R}$ be a polynomial of degree ℓ over some subring of \mathbb{R}^k (e.g., the usual $(\times, +)$ or $(+, \min)$). The problem is to find all roots of the polynomial $p(f_1(x), f_2(x), \dots, f_k(x))$, i.e., the set

$$Z = \{x : p(f_1(x), f_2(x), \dots, f_k(x)) = 0\}.$$

First note that the set of solutions above is in general not finite and is some manifold and highly non-convex. The key idea of Grigoriev and Paleshnik (see also Barvinok Barvinok (1993) for a similar idea applied to a special case) is to show that this set of solutions can be partitioned into a relatively small number of connected components such that there is an into map from these components to roots of a univariate polynomial of degree $(\ell n)^{O(k)}$; this therefore bounds the total number of components. The proof of this mapping is based on an explicit decomposition of space with the property that if a piece of the decomposition has a solution, it must be the solution of a linear system. The number of possible such linear systems is bounded as $n^{O(k)}$, and these systems can be enumerated efficiently.

The core idea of the decomposition starts with the following simple observation that relies crucially on the maps being quadratic (and not of higher degree).

Proposition 13 *The partial derivatives of any degree d polynomial p of quadratic forms $f_i(x)$, where $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, is linear in x for any fixed value of $\{f_1(x), \dots, f_k(x)\}$.*

To see this, suppose $Y_j = f_j(x)$ and write

$$\frac{\partial p}{\partial x_i} = \sum_{j=1}^k \frac{\partial p(Y_1, \dots, Y_k)}{\partial Y_j} \frac{\partial Y_j}{\partial x_i} = \sum_{j=1}^k \frac{\partial p(Y_1, \dots, Y_k)}{\partial Y_j} \frac{\partial f_j(x)}{\partial x_i}.$$

Now the derivatives of f_j are linear in x_i as f_j is quadratic, and so for any fixed values of Y_1, \dots, Y_k , the expression is linear in x .

The next step is a nontrivial fact about connected components of analytic manifolds that holds in much greater generality. Instead of all points that correspond to zeros of p , we look at all “critical” points of p defined as the set of points x for which the partial derivatives in all but the first coordinate, i.e.,

$$Z_c = \{x : \frac{\partial p}{\partial x_i} = 0, \quad \forall 2 \leq i \leq n\}.$$

The theorem says that Z_c will intersect every connected component of Z (Grigor’ev and Vorobjov Jr, 1988).

Now the above two ideas can be combined as follows. We will cover all connected components of Z_c . To do this we consider, for each fixed value of Y_1, \dots, Y_k , the possible

solutions to the linear system obtained, alongside minimizing x_1 . The rank of this system is in general at least $n - k$ after a small perturbation (while Grigoriev and Pasechnik (2005) uses a deterministic perturbation that takes some care, we could also use a small random perturbation). So the number of possible solutions grows only as exponential in $O(k)$ (and not n), and can be effectively enumerated in time $(\ell d)^{O(k)}$. This last step is highly nontrivial, and needs the argument that over the reals, zeros from distinct components need only to be computed up to finite polynomial precision (as rationals) to keep them distinct. Thus, the perturbed version still covers all components of the original version. In this enumeration, we check for true solutions. The method actually works for any level set of p , $\{x : p(x) = t\}$ and not just its zeros. With this, we can optimize over p as well. We conclude this section by paraphrasing the main theorem from Grigoriev and Pasechnik (2005).

Theorem 14 (*Grigoriev and Pasechnik, 2005*) *Given k quadratic maps $q_1, \dots, q_k : \mathbb{R}^k \rightarrow \mathbb{R}$ and a polynomial $p : \mathbb{R}^k \rightarrow \mathbb{R}$ over some computable subring of \mathbb{R} of degree at most ℓ , there is an algorithm to compute a set of points satisfying $p(q_1(x), \dots, q_k(x)) = 0$ that meets each connected component of the set of zeros of p using at most $(\ell n)^{O(k)}$ operations with all intermediate representations bounded by $(\ell n)^{O(k)}$ times the bit sizes of the coefficients of p, q_1, \dots, q_k . The minimizer, maximizer or infimum of any polynomial $r(q_1(x), \dots, q_k(x))$ of degree at most ℓ over the zeros of p can also be computed in the same complexity.*

5.1. Proof of Theorem 5

We apply Theorem 14 and the corresponding algorithm as follows. Our variables will be the entries of an $n \times d$ matrix P . The quadratic maps will be $f_i(P)$ plus additional maps for $q_{ii}(P) = \|P_i\|^2 - 1$ and $q_{ij}(P) = P_i^T P_j$ for columns P_i, P_j of P . The final polynomial is

$$p(f_1, \dots, f_k, q_{11}, \dots, q_{dd}) = \sum_{i \leq j} q_{ij}(P)^2.$$

We will find the maximum of the polynomial $r(f_1, \dots, f_k) = g(f_1, \dots, f_k)$ over the set of zeros of p using the algorithm of Theorem 14. Since the total number of variables is dn and the number of quadratic maps is $k + d(d+1)/2$, we get the claimed complexity of $O(\ell dn)^{O(k+d^2)}$ operations and this times the input bit sizes as the bit complexity of the algorithm.

6. Hardness

Theorem 15 *The FAIR-PCA problem:*

$$\max_{z \in \mathbb{R}, P \in \mathbb{R}^{n \times d}} z \quad \text{subject to} \quad (13)$$

$$\langle B_i, PP^T \rangle \geq z \quad , \forall i \in [k] \quad (14)$$

$$P^T P = I_d \quad (15)$$

for arbitrary $n \times n$ symmetric real PSD matrices B_1, \dots, B_k is NP-hard for $d = 1$ and $k = O(n)$.

Proof We reduce another NP-hard problem of MAX-CUT to the stated fair PCA problem. In MAX-CUT, given a simple graph $G = (V, E)$, we optimize

$$\max_{S \subseteq V} e(S, V \setminus S) \quad (16)$$

over all subset S of vertices. Here, $e(S, V \setminus S) = |\{e_{ij} \in E : i \in S, j \in V \setminus S\}|$ is the size of the cut S in G . As common NP-hard problems, the decision version of MAX-CUT:

$$\exists? S \subseteq V : e(S, V \setminus S) \geq b \quad (17)$$

for an arbitrary $b > 0$ is also NP-hard. We may write MAX-CUT as an integer program as follows:

$$\exists? v \in \{-1, 1\}^V : \frac{1}{2} \sum_{ij \in E} (1 - v_i v_j) \geq b \quad (18)$$

Here v_i represents whether a vertex i is in the set S or not:

$$v_i = \begin{cases} 1 & i \in S \\ -1 & i \notin S \end{cases} \quad (19)$$

and it can be easily verified that the objective represents the desired cut function.

We now show that this MAX-CUT integer feasibility problem can be formulated as an instance of the fair PCA problem (13)-(15). In fact, it will be formulated as a feasibility version of the fair PCA by checking if the optimal z of an instance is at least b . We choose $d = 1$ and $n = |V|$ for this instance, and we write $P = [u_1; \dots; u_n] \in \mathbb{R}^n$. The rest of the proof is to show that it is possible to construct constraints in the fair PCA form (14)-(15) to 1) enforce a discrete condition on u_i to take only two values, behaving similarly as v_i ; and 2) check an objective value of MAX-CUT.

The reason u_i as written cannot behave exactly as v_i is that constraint (15) requires $\sum_{i=1}^n u_i^2 = 1$ but $\sum_{i=1}^n v_i^2 = n$. Hence, we scale the variables in MAX-CUT problem by writing $v_i = \sqrt{n} u_i$ and rearrange terms in (18) to obtain an equivalent formulation of MAX-CUT:

$$\exists? u \in \left\{ -\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}} \right\}^n : n \sum_{ij \in E} -u_i u_j \geq 2b - |E| \quad (20)$$

We are now ready to give an explicit construction of $\{B_i\}_{i=1}^k$ to solve MAX-CUT formulation (20). Let $k = 2n + 1$. For each $j = 1, \dots, n$, define

$$B_{2j-1} = bn \cdot \text{diag}(\mathbf{e}_j), B_{2j} = \frac{bn}{n-1} \cdot \text{diag}(\mathbf{1} - \mathbf{e}_j)$$

where \mathbf{e}_j and $\mathbf{1}$ denote vectors of length n with all zeroes except one at the j th coordinate, and with all ones, respectively. It is clear that B_{2j-1}, B_{2j} are PSD. Then for each $j = 1, \dots, n$, the constraints $\langle B_{2j-1}, PP^T \rangle \geq b$ and $\langle B_{2j}, PP^T \rangle \geq b$ are equivalent to

$$u_j^2 \geq \frac{1}{n}, \text{ and } \sum_{i \neq j} u_j^2 \geq \frac{n-1}{n}$$

respectively. Combining these two inequalities with $\sum_{i=1}^n u_i^2 = 1$ forces both inequalities to be equalities, implying that $u_j \in \left\{-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right\}$ for all $j \in [n]$, as we aim.

Next, we set

$$B_{2n+1} = \frac{bn}{2b - |E| + n^2} \cdot (nI_n - A_G)$$

where $A_G = (\mathbb{I}[ij \in E])_{i,j \in [n]}$ is the adjacency matrix of the graph G . Since the matrix $nI_n - A_G$ is diagonally dominant and real symmetric, B_{2n+1} is PSD. We have that $\langle B_{2n+1}, PP^T \rangle \geq b$ is equivalent to

$$\frac{bn}{2b - |E| + n^2} \left(n \sum_{i=1}^n u_i^2 - \sum_{ij \in E} u_i u_j \right) \geq b$$

which, by $\sum_{i=1}^n u_i^2 = 1$, is further equivalent to

$$n \sum_{ij \in E} -u_i u_j \geq 2b - |E|$$

To summarize, we constructed B_1, \dots, B_{2n+1} so that checking whether an objective of fair PCA is at least b is equivalent to checking whether a graph G has a cut of size at least b , which is NP-hard. \blacksquare

7. Integrality Gap

We showed that FAIR-PCA for $k = 2$ groups can be solved up to optimality in polynomial time using an SDP. For $k > 2$, we used a different, non-convex approach to get a polytime algorithm for any fixed k, d . Here we show that the SDP relaxation of FAIR-PCA has a gap even for $k = 3$ and $d = 1$.

Lemma 16 *The FAIR-PCA SDP relaxation:*

$$\begin{aligned} & \max z \\ & \langle B_i, X \rangle \geq z \quad i \in \{1, \dots, k\} \\ & \text{tr}(X) \leq d \\ & 0 \preceq X \preceq I \end{aligned}$$

for $k = 3$, $d = 1$, and arbitrary PSD $\{B_i\}_{i=1}^k$ contains a gap, i.e. the optimum value of the SDP relaxation is different from one of exact FAIR-PCA problem.

Proof Let $B_1 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$, $B_2 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$, $B_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$. It can be checked that B_i are PSD. The optimum of the relaxation is $7/4$ (given by the optimal solution $X = \begin{bmatrix} 1/2 & 1/8 \\ 1/8 & 1/2 \end{bmatrix}$).

However, an optimal exact FAIR-PCA solution is $\hat{X} = \begin{bmatrix} 16/17 & 4/17 \\ 4/17 & 1/17 \end{bmatrix}$ which gives an optimum $26/17$ (one way to solve for optimum rank-1 solution \hat{X} is by parameterizing $\hat{X} = v(\theta)v(\theta)^T$ for $v(\theta) = [\cos \theta; \sin \theta]$, $\theta \in [0, 2\pi)$). \blacksquare

References

- Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias - propublica. <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>, 2018.
- Afonso S Bandeira, Nicolas Boumal, and Vladislav Voroninski. On the low-rank approach for semidefinite programs arising in synchronization and community detection. In *Conference on Learning Theory*, pages 361–382, 2016.
- Alexander I Barvinok. Feasibility testing for systems of real quadratic equations. *Discrete & Computational Geometry*, 10(1):1–13, 1993.
- Alexander I. Barvinok. Problems of distance geometry and convex properties of quadratic maps. *Discrete & Computational Geometry*, 13(2):189–202, 1995.
- Ahron Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*, volume 2. Siam, 2001.
- Amey Bhangale, Swastik Kopparty, and Sushant Sachdeva. Simultaneous approximation of constraint satisfaction problems. In *International Colloquium on Automata, Languages, and Programming*, pages 193–205. Springer, 2015.
- Amey Bhangale, Subhash Khot, Swastik Kopparty, Sushant Sachdeva, and Devanathan Thimvenkatachari. Near-optimal approximation algorithm for simultaneous max-cut. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1407–1425. Society for Industrial and Applied Mathematics, 2018.
- Nicolas Boumal, Vlad Voroninski, and Afonso Bandeira. The non-convex burer-monteiro approach works on smooth semidefinite programs. In *Advances in Neural Information Processing Systems*, pages 2757–2765, 2016.
- Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91, 2018.
- Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.
- Samuel Burer and Renato DC Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.
- Kate Crawford. The trouble with bias, 2017. URL <http://blog.revolutionanalytics.com/2017/12/the-trouble-with-bias-by-kate-crawford.html>. Invited Talk by Kate Crawford at NIPS 2017, Long Beach, CA.
- Kalyanmoy Deb. Multi-objective optimization. In *Search methodologies*, pages 403–449. Springer, 2014.

- Fabrizio Grandoni, R Ravi, Mohit Singh, and Rico Zenklusen. New approaches to multi-objective optimization. *Mathematical Programming*, 146(1-2):525–554, 2014.
- D Yu Grigor’ev and NN Vorobjov Jr. Solving systems of polynomial inequalities in subexponential time. *Journal of Symbolic Computation*, 5(1-2):37–64, 1988.
- Dima Grigoriev and Dmitrii V Pasechnik. Polynomial-time computing over quadratic maps i: sampling in real algebraic sets. *Computational complexity*, 14(1):20–52, 2005.
- Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.
- Amy F Iezzoni and Marvin P Pritts. Applications of principal component analysis to horticultural research. *HortScience*, 26(4):334–338, 1991.
- Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.
- Ian T Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986.
- 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*, pages 3819–3828. ACM, 2015.
- Lap Chi Lau, Ramamoorthi Ravi, and Mohit Singh. *Iterative methods in combinatorial optimization*, volume 46. Cambridge University Press, 2011.
- Alex Lemon, Anthony Man-Cho So, Yinyu Ye, et al. Low-rank semi-definite programming: Theory and applications. *Foundations and Trends® in Optimization*, 2(1-2):1–156, 2016.
- Jan Karel Lenstra, David B Shmoys, and Eva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming*, 46(1-3):259–271, 1990.
- Gábor Pataki. On the rank of extreme matrices in semi-definite programs and the multiplicity of optimal eigenvalues. *Mathematics of operations research*, 23(2):339–358, 1998.
- Karl Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.
- Imre Pólik and Tamás Terlaky. A survey of the S-lemma. *SIAM review*, 49(3):371–418, 2007.
- Ram Ravi and Michel X Goemans. The constrained minimum spanning tree problem. In *Scandinavian Workshop on Algorithm Theory*, pages 66–75. Springer, 1996.

- Soumya Raychaudhuri, Joshua M Stuart, and Russ B Altman. Principal components analysis to summarize microarray experiments: application to sporulation time series. In *Biocomputing 2000*, pages 455–466. World Scientific, 1999.
- Samira Samadi, Uthaipon Tantipongpipat, Jamie Morgenstern, Mohit Singh, and Santosh Vempala. The price of fair PCA: One extra dimension. <https://arxiv.org/pdf/1811.00103.pdf>, 2018a. To appear at 32nd Conference on Neural Information Processing Systems (NIPS).
- Samira Samadi, Uthaipon Tantipongpipat, Jamie Morgenstern, Mohit Singh, and Santosh Vempala. Fair PCA homepage. <http://www.samirasamadi.com/fair-pca-homepage>, 2018b.
- Anthony Man-Cho So, Yinyu Ye, and Jiawei Zhang. A unified theorem on sdp rank reduction. *Mathematics of Operations Research*, 33(4):910–920, 2008.
- Latanya Sweeney. Discrimination in online ad delivery. *Communications of the ACM*, 56(5):44–54, 2013.
- VA Yakubovich. S-procedure in nonlinear control theory (in russian). *Vestnick Leningrad Univ. Math.*, 1:62–77, 1971.
- VA Yakubovich. S-procedure in nonlinear control theory (english translation). *Vestnick Leningrad Univ. Math.*, 4:73–93, 1997.