Multi-Criteria Dimensionality Reduction with Applications to Fairness

Uthaipon (Tao) Tantipongpipat[*] [†] **Samira Samadi**[*]

Mohit Singh[*] [†] **Jamie Morgenstern**[*] **Santosh Vempala**[*]

Abstract

Dimensionality reduction is a classical technique widely used for data analysis. One foundational instantiation is Principal Component Analysis (PCA), which minimizes the average reconstruction error. In this paper, we introduce the *multi-criteria dimensionality reduction* problem where we are given multiple objectives that need to be optimized simultaneously. As an application, our model captures several fairness criteria for dimensionality reduction such as the Fair-PCA problem introduced by Samadi et al. [2018] and the Nash Social Welfare (NSW) problem. In the Fair-PCA problem, 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 reconstruction error of any one group is minimized. In NSW the goal is to maximize the product of the individual variances of the groups achieved by the common low-dimensinal space.

Our main result is an exact polynomial-time algorithm for the two-criteria dimensionality reduction problem when the two criteria are increasing concave functions. As an application of this result, we obtain a polynomial time algorithm for Fair-PCA for $k = 2$ groups, resolving an open problem of Samadi et al. [2018], and a polynomial time algorithm for NSW objective for $k = 2$ groups. We also give approximation algorithms for $k > 2$. Our technical contribution in the above results is to prove new low-rank properties of extreme point solutions to semi-definite programs. We conclude with experiments indicating the effectiveness of algorithms based on extreme point solutions of semi-definite programs on several real-world datasets.

1 Introduction

Dimensionality reduction is the process of choosing a low-dimensional representation of a large, high-dimensional data set. It is a core primitive for modern machine learning and is being used in image processing, biomedical research, time series analysis, etc. Dimensionality reduction can be used during the preprocessing of the data to reduce the computational burden as well as at the final stages of data analysis to facilitate data summarization and data visualization [Raychaudhuri et al., 1999; Iezzoni and Pritts, 1991]. Among the most ubiquitous and effective of dimensionality reduction techniques in practice are Principal Component Analysis (PCA) [Pearson, 1901; Jolliffe, 1986; Hotelling, 1933], multidimensional scaling [Kruskal, 1964], Isomap [Tenenbaum et al., 2000], locally linear embedding [Roweis and Saul, 2000], and t-SNE [Maaten and Hinton, 2008].

[*]Georgia Institute of Technology. {tao,ssamadi6}@gatech.edu, mohit.singh@isye.gatech.edu, jamiemmt.cs@gatech.edu, vempala@cc.gatech.edu

[†]supported by NSF-AF:1910423 and NSF-AF:1717947

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

One of the major obstacles to dimensionality reduction tasks in practice is complex high-dimensional data structures that lie on multiple different low-dimensional subspaces. For example, Maaten and Hinton [2008] address this issue for low-dimensional visualization of images of objects from diverse classes seen from various viewpoints, or Samadi et al. [2018] study PCA on human data when different groups in the data (e.g., high-educated vs low-educated or men vs women) have an inherently different structure. Although these two contexts might seem unrelated, our work presents a general framework that addresses both issues. In both setting, a single criteria for the dimensionality reduction might not be sufficient to capture different structures in the data. This motivates our study of multi-criteria dimensionality reduction.

As an illustration, consider applying PCA on a high dimensional data to do a visualization analysis in low dimensions. Standard PCA aims to minimize the single criteria of average reconstruction error over the whole data. But the reconstruction error on different parts of data can be widely different. In particular, Samadi et al. [2018] show that on real world data sets, PCA has more reconstruction error on images of women vs images of men. A similar phenomenon is also noticed on other data sets when groups are formed based on education. Unbalanced average reconstruction error or equivalently unbalanced variance could have implications of representational harms [Crawford, 2017] in early stages of data analysis.

Multi-criteria dimensionality reduction. Multi-criteria dimensionality reduction could be used as an umbrella term with specifications changing based on the applications and the metrics that the machine learning researcher has in mind. Aiming for an output with a balanced error over different subgroups seems to be a natural choice as reflected by minimizing the maximum of average reconstruction errors studied by Samadi et al. [2018] and maximizing geometric mean of the variances of the groups, which is the well-studied Nash social welfare (NSW) objective [Kaneko and Nakamura, 1979; Nash Jr, 1950]. Motivated by these settings, the more general question that we would like to study is as following.

Question 1. *How might one redefine dimensionality reduction to produce projections which optimize different groups' representation in a balanced way?*

For simplicity of explanation, we first describe our framework for PCA, but the approach is general and applies to a much wider class of dimensionality reduction techniques. Consider the data points as rows of an $m \times n$ matrix A. For PCA, the objective is to find an $n \times d$ projection matrix P that maximizes the Frobenius norm, $\|AP\|_F^2$ (this is equivalent to minimizing the reconstruction error). Suppose that the rows of A belong to different *groups*, based on demographics or some other semantically meaningful clustering. 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). Multi-criteria dimensionality reduction can then be viewed as simultaneously considering objectives on the different weightings of A. 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.,

$$\max_{P:P^TP=I_d} \min_{1\le i\le k} \|A_iP\|_F^2 = \langle A_i^TA_i, PP^T\rangle. \qquad \text{(FAIR-PCA)}$$

(We note that our FAIR-PCA is different from one in Samadi et al. [2018], but equivalent by additive and multiplicative scalings.) 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 $f_i : \mathcal{P}_d \to \mathbb{R}$ that denotes the group's objective value for a particular projection. For any $g : \mathbb{R}^k \to \mathbb{R}$, we define the (f, g)-multi-criteria dimensionality reduction problem as finding a d-dimensional projection P which optimizes

$$\max_{P\in\mathcal{P}_d} g(f_1(P), f_2(P), \ldots, f_k(P)). \qquad \text{(MULTI-CRITERIA-DIMENSION-REDUCTION)}$$

In the above example of max-min Fair-PCA, g is simply the $\min$ function and $f_i(P) = \|A_iP\|^2$ 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. One could also choose the product function $g(y_1, \ldots, y_k) = \prod_i y_i$ for the accumulating function g. This is also a natural choice, famously introduced in Nash's solution to the bargaining problemNash Jr [1950]; Kaneko and Nakamura [1979]. This framework can also describe the pth power mean of the projections, e.g. $f_i(P) = \|A_iP\|^2$ and $g(y_1, \ldots, y_k) = \left(\sum_{i\in[k]} y_i^{p/2}\right)^{1/p}$.

The appropriate weighting of k objectives often depends on the context and application. The central motivating questions of this paper are the following:

◇ *What is the complexity of* FAIR-PCA *?*

◇ *More generally, what is the complexity of* MULTI-CRITERIA-DIMENSION-REDUCTION *?*

Framed another way, we ask whether these multi-criteria optimization problems force us to incur substantial computational cost compared to optimizing g over A alone. Samadi et al. [2018] introduced the problem of FAIR-PCA and showed how to use the natural semi-definite relaxation to find a rank-$(d+k-1)$ approximation whose cost is at most that of the optimal rank-d approximation. For $k=2$ groups, this is an increase of 1 in the dimension (as opposed to the naïve bound of $2d$, by taking the span of the optimal d-dimensional subspaces for the two groups). The computational complexity of finding the exact optimial solution to FAIR-PCA was left as an open question.

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**
$\max\ z$	$\max\ z$
$\langle A_i^T A_i, X\rangle \geq z \quad i \in \{1,\ldots,k\}$	$\langle A_i^T A_i, X\rangle \geq z \quad i \in \{1,\ldots,k\}$
$\text{rank}(X) \leq d$	$\text{tr}(X) \leq d$
$0 \preceq X \preceq I$	$0 \preceq X \preceq I$

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. This result also resolves the open problem posed by Samadi et al. [2018].

Theorem 1.1. *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.*

Given m datapoints partitioned into $k \leq n$ groups in n dimensions, the algorithm runs in $O(nm + n^{6.5})$ time. $O(mnk)$ is from computing $A_i^T A_i$ and $O(n^{6.5})$ is from solving an SDP over $n \times n$ PSD matrices [Ben-Tal and Nemirovski, 2001]. Our results also hold for the MULTI-CRITERIA-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 1.2. *There is a polynomial time algorithm for 2-group* MULTI-CRITERIA-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 8). 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 MULTI-CRITERIA-DIMENSION-REDUCTION under the same assumptions as in Theorem 1.1. Note that this result generalizes Theorem 1.1.

Theorem 1.3. *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 objective value is at least that of the optimal* d*-dimensional embedding. If* g *is only concave, then the solution lies in at most* $d + 1$ *dimensions.*

This strictly improves and generalizes the bound of $d + k - 1$ for FAIR-PCA . 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 $1 - \frac{O(\sqrt{k})}{d}$.

Theorem 1.4. *Let* $A_1, \ldots, A_k$ *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.*

That is, the algorithm returns a project $P \in \mathcal{P}_d$ of *exact* rank d with objective at least $1 - \frac{s}{d}$ of the optimal objective. More details on the approximation result are in Section 4. The runtime of Theorems 1.2 and 1.3 depends on access to first order oracle to g and standard application of the ellipsoid algorithm would take $\tilde{O}(n^2)$ oracle calls.

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. MULTI-CRITERIA-DIMENSION-REDUCTION captures the marginal loss functions by setting the utility $f_i(P) = \|A_i P\|_F^2 - \max_{Q \in \mathcal{P}_d} \|A_i Q\|_F^2$ for each group i and $g(f_1, f_2, \ldots, f_k) := \min\{f_1, f_2, \ldots, f_k\}$, giving an optimization problem

$$\min_{P \in \mathcal{P}_d} \max_{i \in [k]} \left(\max_{Q \in \mathcal{P}_d} \|A_i Q\|_F^2 - \|A_i P\|_F^2 \right) \tag{1}$$

and the marginal loss objective is indeed the objective of the problem.

In Section 5, we develop a general 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 1.8. 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 1.5. *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. 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 8, 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 1.6. *Let the fairness objective be* $g : \mathbb{R}^k \to \mathbb{R}$ *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 1.7. *Let C and $A_1, \ldots, A_m$ be $n \times n$ real matrices, $d \leq n$, and $b_1, \ldots b_m \in \mathbb{R}$. Suppose the semi-definite program $\mathbb{SDP}(\mathbb{I})$:*

$$\min \langle C, X \rangle \text{ subject to} \tag{2}$$

$$\langle A_i, X \rangle \quad \lhd_i \quad b_i \;\; \forall\, 1 \leq i \leq m \tag{3}$$

$$\text{tr}(X) \quad \leq \quad d \tag{4}$$

$$0 \preceq X \quad \preceq \quad I_n \tag{5}$$

where $\lhd_i \in \{\leq, \geq, =\}$, has a nonempty feasible set. Then, all extreme optimal solutions X^* to $\mathbb{SDP}(\mathbb{I})$ 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. Since the result of Pataki [1998] has been studied from perspective of fast algorithms Boumal et al. [2016]; Burer and Monteiro [2003, 2005] and applied in community detection and phase synchronization Bandeira et al. [2016], we expect our extension of the result to have further applications in many of these areas.

SDP iterative rounding. Using Theorem 1.7, 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 $\mathcal{A} = \{A_1, \ldots, A_m\}$ be a collection of $n \times n$ matrices. For any set $S \subseteq \{1, \ldots, m\}$, 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 1.8. *Let C be a $n \times n$ matrix and $\mathcal{A} = \{A_1, \ldots, A_m\}$ be a collection of $n \times n$ real matrices, $d \leq n$, and $b_1, \ldots b_m \in \mathbb{R}$. Suppose the semi-definite program $\mathbb{SDP}$:*

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

$$\langle A_i, X \rangle \quad \geq \quad b_i \;\; \forall\, 1 \leq i \leq m$$

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

$$0 \preceq X \quad \preceq \quad I_n$$

has a nonempty feasible set and let X^* denote an optimal solution. The Algorithm ITERATIVE-SDP *(see Figure 2 in Appendix) 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* $1 \leq i \leq m$*.*

The time complexity of Theorems 1.7 and 1.8 is analyzed in Sections 2 and 5. Both algorithms introduce the rounding procedures that do not contribute significant computational cost; rather, solving the SDPis the bottleneck for running time both in theory and practice.

1.2 Related work

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. 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 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. We also remark that extreme point solutions to linear programming have played an important role in design of approximation algorithms [Lau et al., 2011] and our result add to the comparatively small, but growing, number of applications for utilizing extreme points of semi-definite programs.

A closely related area, especially to MULTI-CRITERIA-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.

The applications of multi-criteria dimensionality reduction in fairness are closely related to studies on representational bias in machine learning [Crawford, 2017; Noble, 2018; Bolukbasi et al., 2016] and fair resource allocation in game theory [Wei et al., 2010; Fang and Bensaou, 2004]. There have been various mathematical formulations suggested for representational bias in ML [Chierichetti et al., 2017; Celis et al., 2018; Kleindessner et al., 2019; Samadi et al., 2018] among which our model covers unbalanced reconstruction error in PCA suggested by Samadi et al. [2018]. From the game theory literature, our model covers Nash social welfare objective [Kaneko and Nakamura, 1979; Nash Jr, 1950] and others [Kalai et al., 1975; Kalai, 1977].

2 Low-rank solutions of MULTI-CRITERIA-DIMENSION-REDUCTION

In this section, we show that all extreme solutions of SDP relaxation of MULTI-CRITERIA-DIMENSION-REDUCTION have low rank, proving Theorem 1.1-1.3. Before we state the results, we make following assumptions. In this section, we let $g : \mathbb{R}^k \to \mathbb{R}$ 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 $f_1, f_2, \ldots, f_k$ which are affine in PP^T, i.e. we are given real $n \times n$ matrices $B_1, \ldots, B_k$ and constants $\alpha_1, \alpha_2, \ldots, \alpha_k \in \mathbb{R}$ and $f_i(P) = \left\langle B_i, PP^T \right\rangle + \alpha_i$.

We assume g to be G-Lipschitz. For functions $f_1, \ldots, f_k, g$ that are $L_1, \ldots, L_k, G$-Lipschitz, we define an ϵ-optimal solution to (f, g)-MULTI-CRITERIA-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 ith affine constraints by at most ϵL_i. Note that the feasible region of the problem is implicitly bounded by the constraint $X \preceq I_n$.

In this section, 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 $\log \frac{1}{\epsilon}$.

We first prove Theorem 1.7 below. To prove Theorem 1.1-1.3, 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 MULTI-CRITERIA-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.

Proof of Theorem 1.7: Let X^* be an extreme point optimal solution to $\mathbb{SDP}(\mathbb{I})$. 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

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

where $0 \prec \Lambda \prec I_{r-l}$, $Q_1^T Q_1 = I_{r-l}$, $Q_2^T Q_2 = I_l$, 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 $Q_1(\Lambda \pm \delta\Delta)Q_1^\top + Q_2 Q_2^T$ 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

$$Q_1(\Lambda \pm \delta\Delta)Q_1^\top + Q_2 Q_2^T = Q \begin{pmatrix} \Lambda \pm \delta\Delta & 0 \\ 0 & I_l \end{pmatrix} Q^\top$$

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 $f(t) = Q_1(\Lambda + t\Delta)Q_1^\top + Q_2 Q_2^\top, t \geq 0$ 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.

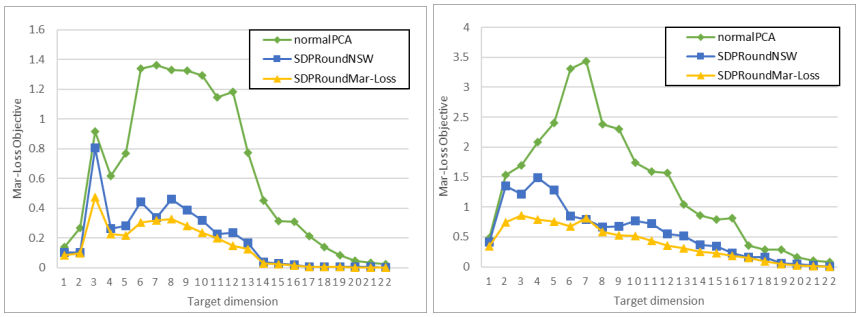

Figure 1: Marginal loss function (see (1)) of standard PCA compared to our SDP-based algorithms on Default Credit data. SDPRoundNSW and SDPRoundMar-Loss are two runs of the SDP-based algorithms maximizing NSW and minimizing marginal loss. Left: $k = 4$ groups. Right: $k = 6$.

The algorithm involves at most n rounds of reducing $r - l$, each of which involves Gaussian elimination and several iterations (from binary search) of $0 \preceq X \preceq I_n$ which can be done by eigenvalue value decomposition. Gaussian elimination and eigenvalue decomposition can be done in $O(n^3)$ time, and therefore the total runtime of SDP rounding is $\tilde{O}(n^4)$ which is polynomial. □

In practice, one may initially reduce the rank of given feasible $\bar{X}$ using an LP rounding (in $O(n^{3.5})$ time) introduced in Samadi et al. [2018] so that the number of rounds of reducing $r - l$ is further bounded by $k - 1$. The runtime complexity is then $O(n^{3.5}) + \tilde{O}(kn^3)$.

The next corollary is obtained from the bound $r - l \leq -\frac{1}{2} + \sqrt{2m + \frac{9}{4}}$ in the proof of Theorem 1.7.

Corollary 2.1. *The number of fractional eigenvalues in any extreme point solution X to $\mathbb{SDP}(\mathbb{I})$ 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 MULTI-CRITERIA-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)-MULTI-CRITERIA-DIMENSION-REDUCTION problem is to solve

$$\max_{P \in \mathcal{P}_d} g(f_1(P), f_2(P), \ldots, f_k(P)) \tag{6}$$

Theorem 2.2. *There exists a polynomial-time algorithm to solve (f, g)-*MULTI-CRITERIA-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.*

The proof of Theorem 2.2 appears in Appendix. If the assumption that g is monotonic in at least one coordinate is dropped, Theorem 2.2 will hold with r^* by indexing constraints (11) in $\mathbb{SDP}(\mathbb{III})$ for all groups instead of $k - 1$ groups.

3 Experiments

First, we note that experiments for two groups was done in Samadi et al. [2018]. The algorithm outputs optimal solutions with exact rank, despite their weaker guarantee that the rank may be violated by at most 1. Hence, our result of Theorem 1.1 is a mathematical explanation of their missing empirical finding for two groups. We extend their experiments to more number of groups and objectives as follows (See Appendix for results on NSW objective and an additional dataset).

We perform experiments using the algorithm as outlined in Section 2 on the Default Credit data set [Yeh and Lien, 2009] for different target dimensions d. The data is partitioned into $k = 4, 6$ groups by education and gender, and preprocessed to have mean zero and same variance over features. We specified our algorithms by two objectives for MULTI-CRITERIA-DIMENSION-REDUCTION problem introduced earlier: the marginal loss function and Nash

social welfare. The code is publicly available at https://github.com/SDPforAll/multiCriteriaDimReduction. Figure 1 shows the the marginal loss by our algorithms compared to a standard PCA on the entire dataset. Our algorithms significantly reduce "unfairness" in marginal loss of PCA that the standard PCA subtly introduces.

In the experiments, extreme point solutions from SDPs enjoy lower rank violation than our worst-case guarantee. Indeed, while the guarantee is that the numbers of additional rank are at most $s = 1, 2$ for $k = 4, 6$, almost all SDP solutions have *exact* rank, and in rare cases when the solutions are not exact, the rank violation is only one. While we know that our rank violation guarantee cannot be improved in general (due to the integrality gap in Section 8), this opens a question whether the guarantee is better for instances that arise in practice.

References

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 m ax-c ut. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1407–1425. Society for Industrial and Applied Mathematics, 2018.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.

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.

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.

L Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth K Vishnoi. Fair and diverse dpp-based data summarization. *arXiv preprint arXiv:1802.04023*, 2018.

Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pages 5029–5037, 2017.

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.

Zuyuan Fang and Brahim Bensaou. Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad hoc networks. In *IEEE infocom*, volume 2, pages 1284–1295. Citeseer, 2004.

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.

Ian T Jolliffe. Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer, 1986.

Ehud Kalai. Proportional solutions to bargaining situations: interpersonal utility comparisons. *Econometrica: Journal of the Econometric Society*, pages 1623–1630, 1977.

Ehud Kalai, Meir Smorodinsky, et al. Other solutions to nash bargaining problem. *Econometrica*, 43(3):513–518, 1975.

Mamoru Kaneko and Kenjiro Nakamura. The nash social welfare function. *Econometrica: Journal of the Econometric Society*, pages 423–435, 1979.

Matthäus Kleindessner, Samira Samadi, Pranjal Awasthi, and Jamie Morgenstern. Guarantees for spectral clustering with fairness constraints. *arXiv preprint arXiv:1901.08668*, 2019.

Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

Lap Chi Lau, Ramamoorthi Ravi, and Mohit Singh. *Iterative methods in combinatorial optimization*, volume 46. Cambridge University Press, 2011.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

John F Nash Jr. The bargaining problem. *Econometrica: Journal of the Econometric Society*, pages 155–162, 1950.

Safiya Umoja Noble. *Algorithms of oppression: How search engines reinforce racism*. nyu Press, 2018.

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.

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.

UCI Machine Learning Repository. Adult data set. https://archive.ics.uci.edu/ml/datasets/adult. Accessed May 2019.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

Samira Samadi, Uthaipon Tantipongpipat, Jamie H Morgenstern, Mohit Singh, and Santosh Vempala. The price of fair pca: One extra dimension. In *Advances in Neural Information Processing Systems*, pages 10976–10987, 2018.

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.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

Guiyi Wei, Athanasios V Vasilakos, Yao Zheng, and Naixue Xiong. A game-theoretic method of fair resource allocation for cloud computing services. *The journal of supercomputing*, 54(2):252–269, 2010.

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.

Appendix

Proof of Theorem 2.2: First, we write a relaxation of (6):

$$\max_{X\in\mathbb{R}^{n\times n}} g(\langle B_1, X\rangle + \alpha_1, \ldots, \langle B_k, X\rangle + \alpha_k) \text{ subject to} \tag{7}$$

$$\text{tr}(X) \le d \tag{8}$$

$$0 \preceq X \preceq I_n \tag{9}$$

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 $\mathbb{SDP}(\mathbb{III})$:

$$\max_{X\in\mathbb{R}^{n\times n}} \quad \langle B_1, X\rangle \text{ subject to} \tag{10}$$

$$\langle B_i, X\rangle = \langle B_i, \bar{X}\rangle \qquad \forall\ 2 \le i \le k \tag{11}$$

$$\text{tr}(X) \le d \tag{12}$$

$$0 \preceq X \preceq I_n \tag{13}$$

$\mathbb{SDP}(\mathbb{III})$ has a feasible solution $\bar{X}$ of objective $\langle B_1, X\rangle$ and note that there are $k-1$ constraints in (11). Hence, we can apply the algorithm in Theorem 1.7 with $m = k-1$ to find an extreme solution X^* of $\mathbb{SDP}(\mathbb{III})$ of rank at most r^*. Since g is nondecreasing in $\langle B_1, X\rangle$, optimal solutions to $\mathbb{SDP}(\mathbb{III})$ gives objective value g at least the optimum of the relaxation and hence at least the optimum of the original MULTI-CRITERIA-DIMENSION-REDUCTION problem. □

Another way to state Theorem 2.2 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. In particular, it applies to FAIR-PCA with two groups, proving Theorem 1.1.

Corollary 3.1. *The (f, g)-MULTI-CRITERIA-DIMENSION-REDUCTION problem on two groups can be solved in polynomial time.*

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 $A_1, \ldots, A_k$, FAIR-PCA problem is to solve

$$\max_{P:P^TP=I_d} \min_{1\le i\le k} \|A_iP\|_F^2 = \langle A_i^TA_i, PP^T\rangle$$

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

Corollary 4.1. *Let $A_1, \ldots, A_k$ 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 2.2, 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. □

5 Iterative rounding framework with applications to FAIR-PCA

In this section, we first prove Theorem 1.8.

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 $\{1, \ldots, m\}$.

The algorithm is specified in Figure 2. In each iteration, we formulate the following $\mathbb{SDP}(r)$ 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 $F = I_n$, $S \leftarrow \{1, \ldots, m\}$.
2. If the $\mathbb{SDP}$ is infeasible, declare infeasibility. Else,
3. While F is not the empty matrix.
   (a) Solve $\mathbb{SDP}(r)$ 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 2: Iterative Rounding Algorithm ITERATIVE-SDP.

It is easy to see that the semi-definite program remains feasible over all iterations if $\mathbb{SDP}$ 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 5.1. *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 2.1 and Claim 5.1. 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 5.2. *Consider any extreme point solution $X(r)$ of $\mathbb{SDP}(r)$ 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 2.1, it follows that number of fractional eigenvalues of $X(r)$ is at most $-\frac{1}{2} + \sqrt{2l + \frac{9}{4}} \le \sqrt{2l} + 1$. Observe that $l > 0$ since $\text{rank}(X(r)) > \text{tr}(X(r))$. Thus $\text{rank}(X_f) \le \sqrt{2l} + 1$. Moreover, $0 \preceq X_f \preceq I$, thus from Claim 5.1, we obtain that

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

where the first inequality follows from Claim 5.1 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 1.8. 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 $X = F_1F_1^T + FX_fF^T$ 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 $X = F_1F_1^T + FX_fF^T$ satisfies

$$\begin{aligned}&\langle A_i, X\rangle = \langle A_i, F_1F_1^T\rangle + \langle A_i, FX_fF^T\rangle\\ =&\langle A_i, F_1F_1^T\rangle + \langle F^TA_iF, X_f\rangle \le \langle A_i, F_1F_1^T\rangle + b_i - \langle A_i, F_1F_1^T\rangle = b_i.\end{aligned}$$

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, $\langle F^TA_iF, X_f\rangle \le \Delta$. Thus, $\langle A_i, F_1F_1^T\rangle \ge b_i - \Delta$. In the final solution this bound can only go up as F_1 might only become larger. This completes the proof of theorem.

We now analyze the runtime of the algorithm which contains at most k iterations. Each iteration requires solving an SDP and eigenvector decompositions over $r \times r$ matrices, and recomputing F. The SDP has runtime $O(r^{6.5})$ which exceeds eigenvector decomposition and computing X_f, F takes $O(n^2)$. However, the result in Section 2 shows that $r \le \sqrt{2k}$, and hence the total runtime of iterative rounding is $O(k^{4.25} + kn^2)$.

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 $\Delta(\{A_1^TA_1, \ldots, A_k^TA_k\})$. While this is no better than what we get by scaling (Corollary 4.1) 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 $\sqrt{k}$.

6 Polynomial time algorithm for fixed number of groups

Functions of quadratic maps. We briefly summarize the approach of Grigoriev and Pasechnik [2005]. Let $f_1, \ldots, f_k : \mathbb{R}^n \to \mathbb{R}$ be real-valued quadratic functions in n variables. Let $p : \mathbb{R}^k \to \mathbb{R}$ 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 $p(f_1(x), f_2(x), \ldots, f_k(x))$, i.e., the set

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

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

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

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

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

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

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 $Y_1, \ldots, Y_k$, 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 6.2. *[Grigoriev and Pasechnik, 2005] Given k quadratic maps $q_1, \ldots, q_k : \mathbb{R}^k \to \mathbb{R}$ and a polynomial $p : \mathbb{R}^k \to \mathbb{R}$ over some computable subring of $\mathbb{R}$ of degree at most ℓ, there is an algorithm to compute a set of points satisfying $p(q_1(x), \ldots, q_k(x)) = 0$ 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 $p, q_1, \ldots, q_k$. The minimizer, maximizer or infimum of any polynomial $r(q_1(x), \ldots, q_k(x))$ of degree at most ℓ over the zeros of p can also be computed in the same complexity.*

6.1 Proof of Theorem 1.6

We apply Theorem 6.2 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

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

We will find the maximum of the polynomial $r(f_1, \ldots f_k) = g(f_1, \ldots, f_k)$ over the set of zeros of p using the algorithm of Theorem 6.2. 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.

7 Hardness

Theorem 7.1. *The* FAIR-PCA *problem:*

$$\max_{z \in \mathbb{R}, P \in \mathbb{R}^{n \times d}} z \qquad \text{subject to} \tag{14}$$

$$\left\langle B_i, PP^T \right\rangle \ge z \qquad , \forall i \in [k] \tag{15}$$

$$P^T P = I_d \tag{16}$$

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

Proof of Theorem 7.1: 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

$$\max_{S \subseteq V} e(S, V \setminus S) \tag{17}$$

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:

$$\exists ? S \subseteq V : e(S, V \setminus S) \ge b \tag{18}$$

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

$$\exists ? v \in \{-1, 1\}^V : \frac{1}{2} \sum_{ij \in E} (1 - v_i v_j) \ge b \tag{19}$$

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

$$v_i = \begin{cases} 1 & i \in S \\ -1 & i \notin S \end{cases} \tag{20}$$

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 (14)-(16). 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 $P = [u_1; \ldots; u_n] \in \mathbb{R}^n$. The rest of the proof is to show that it is possible to construct constraints in the fair PCA form (15)-(16) 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 (16) 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 (19) to obtain an equivalent formulation of MAX-CUT:

$$\exists ? u \in \{-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\}^n : n \sum_{ij \in E} -u_i u_j \ge 2b - |E| \tag{21}$$

We are now ready to give an explicit construction of $\{B_i\}_{i=1}^k$ to solve MAX-CUT formulation (21). Let $k = 2n + 1$. For each $j = 1, \ldots, n$, 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 jth coordinate, and with all ones, respectively. It is clear that B_{2j-1}, B_{2j} are PSD. Then for each $j = 1 \ldots, n$, 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 $u_j \in \{-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\}$ 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 $B_1, \ldots, B_{2n+1}$ 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. □

8 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 8.1. *The* FAIR-PCA *SDP relaxation:*

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

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 of Lemma 8.1: 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 $v(\theta) = [\cos\theta; \sin\theta]$, $\theta \in [0, 2\pi)$). □

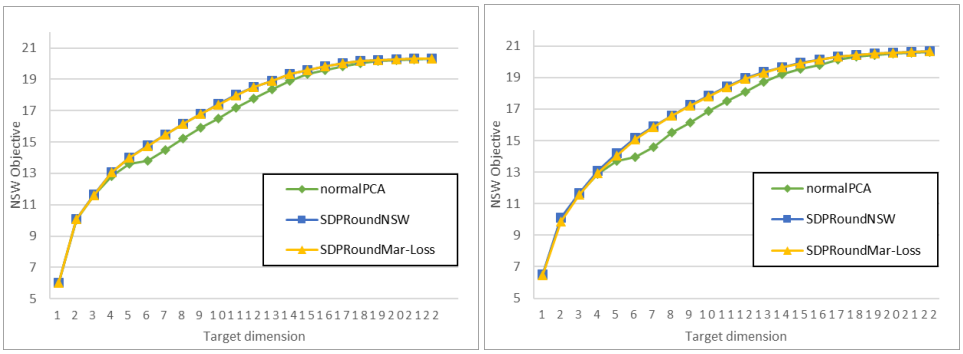

Figure 3: NSW objective of standard PCA compared to our SDP-based algorithms on Default Credit data. SDPRoundNSW and SDPRoundMar-Loss are two runs of the SDP algorithms maximizing NSW objective and minimizing maximum marginal loss. Left: $k = 4$ groups. Right: $k = 6$.

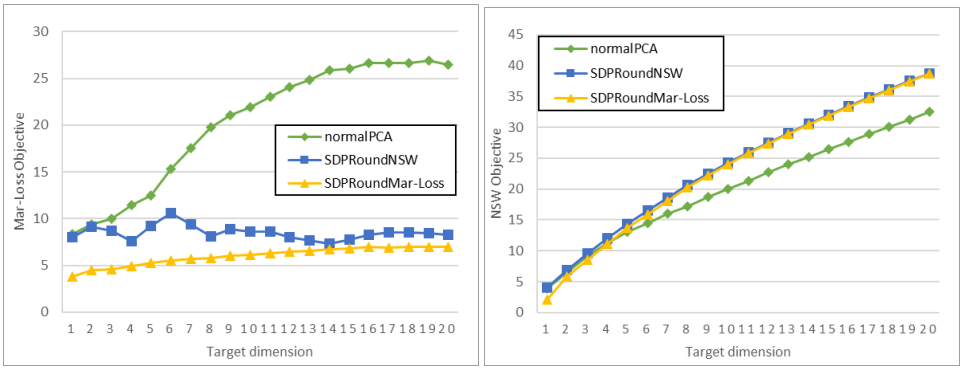

Figure 4: Marginal loss and NSW objective of standard PCA compared to our SDP-based algorithms on Adult Income data. SDPRoundNSW and SDPRoundMar-Loss are two runs of the SDP algorithms maximizing NSW objective and minimizing maximum marginal loss.

9 Extended experiments

We also assess the performance of PCA with NSW objective, summarized in Figure 3. With respect to NSW, standard PCA performs marginally worse (about 10%) compared to our algorithms. It is worth noting from Figures 1 and 3 that our algorithms that try to optimize either marginal loss function or NSW also perform well on the other fairness objective, making these PCAs promising candidates for fairness application.

Same experiments were done on the Adult Income data [Repository]. Some categorial features are preprocessed into integers vectors and some categorical features and rows with missing values are discarded. The final preprocessed data contains $m = 32560$ datapoints in $n = 59$ dimensions, partitioned into $k = 5$ groups based on race. Figure 4 shows the performance of our SDP-based algorithms compared to standard PCA on marginal loss and NSW objectives. Similar to the Credit Data, optimizing either marginal loss or NSW gives a PCA solution that also performs well in another criteria, and better than the standard PCA in both objectives. Almost all SDP solutions are exact without any rank violation.

We found that the running time of solving SDP, which depends on n, is the bottleneck in all experiments. Each run (for one value of d) of the experiments is fast (< 0.5 seconds) on Default Credit data which has $n = 23$, whereas one on Adult Income data ($n = 59$) takes between 10 and 15 seconds. However, it is worth noting that the runtime does not increase in noticeable way from the numbers of datapoints and groups: larger m only increases the data preprocessing time to obtain $n \times n$ matrices and larger k increases the number of constraints. SDP solver and rounding algorithms can handle moderate number of affine constraints efficiently. This observation is as expected from the theoretical analysis.