BIT Numerical Mathematics (2026) 66:17
https://doi.org/10.1007/s10543-026-01109-9

# Fair principal component analysis via eigenvalue optimization

**Junhui Shen[1] · Aaron Davis[2] · Ding Lu[2] · Zhaojun Bai[3]**

Received: 29 August 2025 / Accepted: 28 January 2026


**Abstract**
Principal Component Analysis (PCA) is a foundational technique in machine learning for reducing the dimensionality of high-dimensional datasets. However, PCA can lead to biased representations that disadvantage certain subgroups within the data. To address this issue, a Fair PCA (FPCA) model was introduced to equalize the reconstruction loss between subgroups, but the existing semidefinite relaxation (SDR) based approach is computationally expensive even for a suboptimal solution. Although several alternative FPCA variants have been developed to improve efficiency, they often shift attention away from equalizing the reconstruction loss – the central goal of FPCA. In this paper, we identify a hidden convexity in FPCA and introduce a new algorithm that solves the resulting convex optimization via an eigenvalue optimization. Our approach achieves the desired fairness in reconstruction loss without sacrificing performance. Experiments on real-world datasets show that the proposed FPCA algorithm is approximately $8\times$ faster than the SDR-based algorithm while being at most 85% slower than standard PCA.

**Keywords** Principal component analysis · Fair machine learning · Joint numerical range · Trace minimization · Eigenvalue optimization

Dedicated to Åke Björck on the occasion of his 90th birthday.

✉ Zhaojun Bai
zbai@ucdavis.edu

Junhui Shen
junshen@ucdavis.edu

Aaron Davis
Aaron.Davis1@uky.edu

Ding Lu
Ding.Lu@uky.edu

1 Department of Mathematics, University of California, Davis, Davis, CA 95616, USA

2 Department of Mathematics, University of Kentucky, Lexington, KY 40506, USA

3 Department of Computer Science and Department of Mathematics, University of California, Davis, Davis, CA 95616, USA

Published online: 17 February 2026

**Mathematics Subject Classification** 15A18 · 65F15 · 65K10

## 1 Introduction

Machine learning has revolutionized decision-making accross many domains, yet concerns about fairness persist due to various levels of bias throughout the process. Bias can arise from skewed data, such as non-representative samples or measurement errors [9, 28], as well as from algorithms that prioritize overall accuracy at the expense of fairness [10, 17]. Addressing these biases issues is essential to prevent discriminatory outcomes and ensure the integrity and reliability of the decision-making procedure.

PCA is arguably the most prominent linear dimensionality reduction technique in machine learning and data science [16, 32]. However, standard PCA ignores disparities between subgroups of the underlying datasets and inadvertently leads to biased or discriminatory representations, which can have particularly serious consequences in socially impactful applications. This challenge has prompted the development of various fair PCA models aimed at mitigating such biases. Approaches to fair PCA generally fall into two categories: (a) methods that equalize the distributions of dimensionality-reduced data and (b) methods that equalize the approximation errors introduced by dimensionality reduction accross subgroups.

Approaches in the first category focus mostly on mitigating statistical inference of sensitive attributes in the projection. For example, [29] aims to reduce the disparities in the group means and covariances of the projected data using semidefinite programming. [26] reduces the maximum mean discrepancy between the distributions of protected groups through an exact penalty method. [20] seeks to achieve statistical independence between the projected data and its sensitive attributes by mapping the data onto the null space of a vector involving those attributes. Meanwhile, [24] introduces a “null it out” approach, which nullifies the directions in which the sensitive attribute can be inferred, using unfair directions including the mean difference and the eigenvectors of the second moment difference via a noisy power method.

In the second category of approaches, [36] introduces a Fair PCA (FPCA) model that seeks to equalize *reconstruction loss* between subgroups by minimizing the maximal reconstruction losses. The proposed algorithm relies on semidefinite relaxation and involves solving a semidefinite program followed by a linear program. This process is computationally expensive and can operate “*10 to 15 times*” slower than standard PCA [36]. Moreover, the semidefinite relaxation may introduce extra dimensions in the projection subspace in order to meet the fairness constraints; an upper bound on these extra dimensions is established in [37]. To address the computational challenges inherent in semidefinite relaxation, [21] introduces a new framework to minimize the overall reconstruction error and the group-wise reconstruction losses in a Pareto-optimal sense by adaptive gradient descent. Along the same line of multi-objective optimization, [30] introduces a strength Pareto evolutionary algorithm that uses the disparity between reconstruction errors as a fairness measure, Subsequently, [31] proposes to formulate the multi-objective problem as a single-objective optimization by optimally weighting the objective functions, which is then solved by eigenvalue decompositions. More recently, [41] introduces an alternating Riemannian projected gradient descent ascent algorithm for a generalized fair PCA problem.

The goal of this work is to revisit the FPCA approach proposed in [36] and introduce a new algorithm that computes FPCA without relying on semidefinite relaxation or linear programming. Our contributions are threefold:

- We uncover a hidden convexity in FPCA by reformulating it as a convex optimization over a joint numerical range. This reformulation provides a geometric interpretation of FPCA and enables the development of an efficient solver.
- We develop an efficient and reliable algorithm to solve the resulting convex optimization problem using a univariate eigenvalue optimization. This method yields an optimal orthogonal basis $U$ for the projection subspace.
- Through extensive experiments on human-centric datasets, we demonstrate that the proposed algorithm produces numerically accurate solutions while achieving up to $8\times$ speedup over the FPCA algorithm via semidefinite relaxation. Compared to standard PCA (without fairness constraints), the new algorithm incurs at most an 85.81% increase in runtime.

The rest of this paper is organized as follows: Section 2 reviews standard PCA and the associated fairness issue. Section 3 presents the Fair PCA model and introduces the hidden convexity. Section 4 develops an algorithm based on eigenvalue optimization for the Fair PCA and describes its implementation. Section 5 reports numerical experiments and compares the proposed methods with both the original Fair PCA method by [36] and standard PCA. Finally, Section 6 provides concluding remarks.

*Notations.* We use standard notation in matrix analysis. For a matrix $M \in \mathbb{R}^{m\times n}$, we denote its Frobenius norm by $\|M\|_F$ and its spectral norm by $\|M\|_2$. The trace of a square matrix is denoted by $\mathrm{Tr}(\cdot)$. For a symmetric matrix $S \in \mathbb{R}^{n\times n}$, its eigenvalues are ordered as $\lambda_1(S) \le \lambda_2(S) \le \cdots \le \lambda_n(S)$. For a general matrix $M \in \mathbb{R}^{m\times n}$, its singular values are ordered as $\sigma_1(M) \ge \sigma_2(M) \ge \cdots \ge \sigma_{\min(m,n)}(M)$. The set of $n \times r$ orthogonal matrices (the Stiefel manifold) is

$$\mathbb{O}^{n\times r} := \left\{ U \in \mathbb{R}^{n\times r} : U^T U = I_r \right\}. \tag{1.1}$$

Other notations will be explained as used.

## 2 PCA and fairness issue

PCA is a fundamental technique for dimensionality reduction. Let $M \in \mathbb{R}^{m\times n}$ be a data matrix, where each row represents a sample with $n$ features, and assume that $M$ is centered, i.e., $\mathbf{1}^T M = 0$. The goal of PCA is to find a projection basis $U \in \mathbb{O}^{n\times r}$ that reduces the feature dimension from $n$ to $r$ while best preserving the variance in the data. The optimal projection $U_M$ is found by minimizing the *reconstruction error*

$$\min_{U\in\mathbb{O}^{n\times r}} \left\| M - MUU^T \right\|_F^2 . \tag{2.1}$$

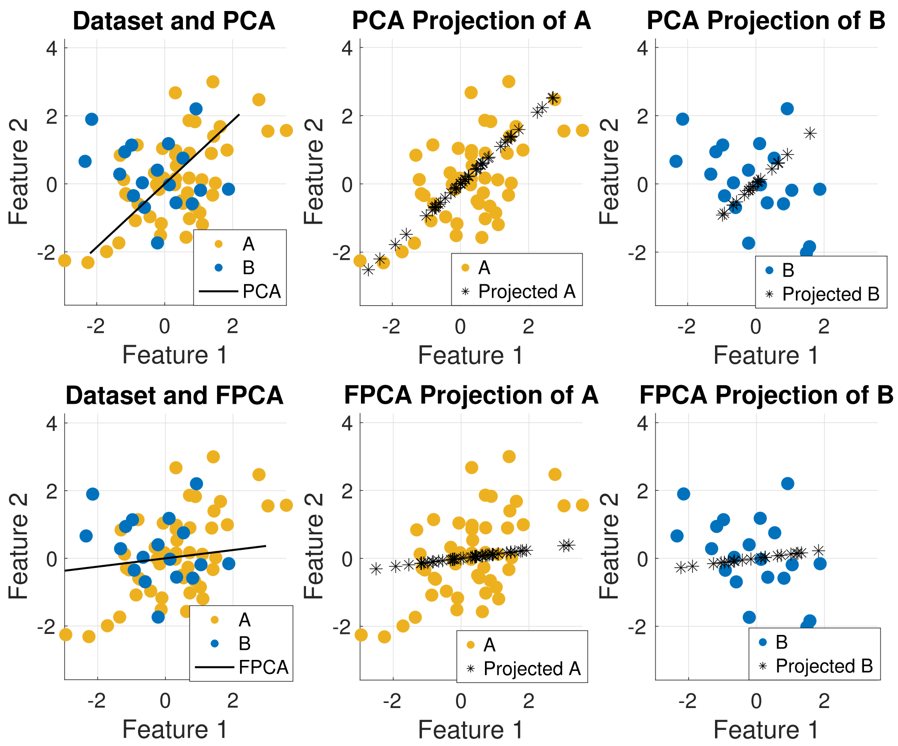


**Fig. 1** **Top panel:** the leading principal components of $M$ obtained by standard PCA captures the maximum variance of $A$ at the expense of $B$, resulting in an unfair projection. **Bottom panel:** FPCA effectively reduces this imbalance in the captured variances

It is well known [18, pp.534-541] that the optimization problem (2.1) is equivalent to the trace maximization

$$\max_{U \in \mathbb{O}^{n \times r}} \operatorname{Tr}\left(U^T M^T M U\right), \tag{2.2}$$

and that the optimal solution $U_M \in \mathbb{O}^{n \times r}$ consists of the orthonormal eigenvectors corresponding to the largest $r$ eigenvalues of the matrix $M^T M$.

Real-world datasets often contain distinct subsets of data with unique characteristics, such as gender, race, or other attributes. Applying PCA directly to such datasets can raise fairness concerns. Consider a common setting where a dataset consists of two subgroups, $A$ and $B$, where subgroup $A$ has $m_1$ points and subgroup $B$ has $m_2$. The data matrix $M \in \mathbb{R}^{m \times n}$ can then be written as

$$M = \begin{bmatrix} A \\ B \end{bmatrix}, \tag{2.3}$$

where $A \in \mathbb{R}^{m_1 \times n}$ and $B \in \mathbb{R}^{m_2 \times n}$. When standard PCA is applied to the entire dataset to capture the maximum variance of $M$, it can fail to account for disparities between the subgroups $A$ and $B$, potentially producing an unfair projection in which one subgroup is disproportionately affected; see Figure 1 for an illustrative example. We note that applying standard PCA separately to each subgroup – resulting in two different projection matrices – would neglect cross-group information and introduce ethical concerns [36]. Hence, it is necessary to seek a single projection subspace for the entire dataset that also accounts for subgroup disparities.

## 3 Fair PCA and hidden convexity

In this section, we revisit the Fair PCA model from [36] and reveal a hidden convexity property within this model.

### 3.1 Fair PCA model

The goal of the Fair PCA [36] is to find an optimal projection that achieves equity in the (average) *reconstruction loss* between individual subgroups.

**Definition 3.1** ([36]) The (average) *reconstruction loss* for a given data matrix $D \in \mathbb{R}^{p\times n}$ by a projection basis matrix $U \in \mathbb{O}^{n\times r}$ is defined as

$$\mathrm{loss}_D(U) = \frac{1}{p}\Big(\|D - DUU^T\|_F^2 - \|D - DU_DU_D^T\|_F^2\Big), \tag{3.1}$$

where $U_D \in \mathbb{O}^{n\times r}$ denotes the PCA solution for $D$, that is, $U_D$ consists of orthogonal eigenvectors corresponding to the largest $r$ eigenvalues of $D^TD$.

Recall that the solution $U_D$ by PCA provides the optimal projection basis for the matrix $D$ that achieves the minimal reconstruction error $\|D - DUU^T\|_F^2$. Hence, the reconstruction loss, $\mathrm{loss}_D(U)$ in (3.1), measures how much worse a given projection basis $U$ is compared to the optimal solution $U_D$ in terms of the reconstruction error. Clearly, due to the minimality of $U_D$, we have

$$\mathrm{loss}_D(U) \geq 0 \quad \text{for all } U \in \mathbb{O}^{n\times r}, \tag{3.2}$$

where equality holds if and only if $U$ is a PCA solution of $D$ (note that the solution $U_D$ by PCA is not unique if there are repeated singular values, e.g., $\sigma_r(D) = \sigma_{r+1}(D)$).

By Definition 3.1, a projection by the basis matrix $U \in \mathbb{O}^{n\times r}$ can be viewed as a *fair projection* for two subgroups of data, $A \in \mathbb{R}^{m_1\times n}$ and $B \in \mathbb{R}^{m_2\times n}$, if it ensures equity in their reconstruction losses, i.e.,

$$\mathrm{loss}_A(U) = \mathrm{loss}_B(U). \tag{3.3}$$

In other words, the linear dimensionality reduction by the projection of $U$ should represent the two subgroups $A$ and $B$ with "equal fidelity". To find such a fair projection basis matrix $U$, the following Fair PCA (FPCA) model is introduced in [36]:

$$\min_{U \in \mathbb{O}^{n\times r}} \max \left\{ \mathrm{loss}_A(U),\ \mathrm{loss}_B(U) \right\}. \tag{3.4}$$

Intuitively, the model (3.4) minimizes the maximum reconstruction loss to prevent any subgroup from incurring a disproportionately large loss; see Figure 1 for an illustration. Later in Theorem 3.1, we will show that the solution of the minimax problem (3.4) guarantees equity in the reconstruction loss for subgroups $A$ and $B$ as given by (3.3).

### 3.2 Fair PCA as trace optimization

For analysis and computation, we express the loss function in (3.1) as a trace form and thereby recast the FPCA model (3.4) as minimizing the maximum of two trace terms.

**Lemma 3.1** *Let $D \in \mathbb{R}^{p\times n}$, $U \in \mathbb{O}^{n\times r}$, and $\mathrm{loss}_D(U)$ be as defined in (3.1). We have*

$$\mathrm{loss}_D(U) = \mathrm{Tr}(U^T H_D U), \tag{3.5}$$

*where $H_D \in \mathbb{R}^{n\times n}$ is a symmetric matrix given by*

$$H_D = \frac{1}{p}\left( \left[ \frac{1}{r}\sum_{i=1}^{r} \sigma_i^2(D) \right] \cdot I_n - D^T D \right), \tag{3.6}$$

*where $\sigma_i(D)$ denotes the $i$-th largest singular value of $D$.*

***Proof*** By a straightforward derivation, we obtain

$$\left\| D(I_n - UU^T) \right\|_F^2 = \mathrm{Tr}\Big( D(I_n - UU^T)D^T \Big) = \mathrm{Tr}(D^T D) - \mathrm{Tr}(U^T D^T D U),$$

where we used identities $\|M\|_F^2 \equiv \mathrm{Tr}(MM^T)$ and $U^T U = I_r$ in the first equation, and $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ in the second equation. Consequently,

$$\begin{aligned} \mathrm{loss}_D(U) &= \frac{1}{p}\left( \left\| D(I_n - UU^T) \right\|_F^2 - \left\| D(I_n - U_D U_D^T) \right\|_F^2 \right) \\ &= \frac{1}{p}\Big( \mathrm{Tr}(U_D^T D^T D U_D) - \mathrm{Tr}(U^T D^T D U) \Big). \end{aligned}$$

Since $U_D$ contains the eigenvectors for the $r$ largest eigenvalues of $D^T D$, or equivalently, the right singular vectors for the largest $r$ singular values of $D$, we have

$$\mathrm{Tr}(U_D^T D^T D U_D) \equiv \sum_{i=1}^{r} \sigma_i(D)^2 = \mathrm{Tr}\left( U^T \left[ \frac{1}{r}\sum_{i=1}^{r} \sigma_i(D)^2 \cdot I_n \right] U \right).$$

Combining the two equations from above, we proved (3.5). □

The trace expression (3.5) for the loss function (3.1), which did not appear in the original FPCA work [36], is a key observation for our latter development. By this expression, FPCA (3.4) can be reformulated as the following minimax problem:

$$\min_{U \in \mathbb{O}^{n\times r}} \max \left\{ \mathrm{Tr}(U^T H_A U),\ \mathrm{Tr}(U^T H_B U) \right\}, \tag{3.7}$$

where

$$H_A = \frac{1}{m_1}\left(s_A \cdot I_n - A^T A\right) \quad \text{and} \quad H_B = \frac{1}{m_2}\left(s_B \cdot I_n - B^T B\right) \tag{3.8}$$

with $s_A = \frac{1}{r}\sum_{i=1}^r \sigma_i^2(A)$ and $s_B = \frac{1}{r}\sum_{i=1}^r \sigma_i^2(B)$. An immediate benefit of the reformulation (3.7) is that it provides a simpler proof of the fairness condition (3.3) for the optimal solution $U_*$ than the one given in [36], as shown in the following theorem.

**Theorem 3.1** *The solution $U_* \in \mathbb{O}^{n\times r}$ of the minimax problem (3.7) satisfies*

$$\mathrm{Tr}(U_*^T H_A U_*) = \mathrm{Tr}(U_*^T H_B U_*). \tag{3.9}$$

***Proof*** By contradiction, assume that the equality in (3.9) does not hold. Without loss of generality, let

$$\mathrm{Tr}(U_*^T H_A U_*) > \mathrm{Tr}(U_*^T H_B U_*). \tag{3.10}$$

Since $\mathrm{Tr}(\cdot)$ is a continuous function, the inequality (3.10) implies for all $U \in \mathbb{O}^{n\times r}$ sufficiently close to $U_*$:

$$\mathrm{Tr}(U^T H_A U) > \mathrm{Tr}(U^T H_B U)$$

and consequently,

$$\begin{aligned}\mathrm{Tr}(U^T H_A U) &= \max\left\{\mathrm{Tr}(U^T H_A U), \mathrm{Tr}(U^T H_B U)\right\}\\ &\geq \max\left\{\mathrm{Tr}(U_*^T H_A U_*), \mathrm{Tr}(U_*^T H_B U_*)\right\} = \mathrm{Tr}(U_*^T H_A U_*),\end{aligned}$$

where the inequality is due to $U_*$ being a minimal solution of FPCA (3.7). Since the inequality above holds for all $U$ that is sufficiently close to $U_*$, it follows that $U_*$ is a local minimum of the trace minimization

$$\min_{U \in \mathbb{O}^{n\times r}} \mathrm{Tr}(U^T H_A U). \tag{3.11}$$

Recalling that any local minimizer of the trace minimization (3.11) must be a global minimizer [22], $U_*$ must be a global minimizer of (3.11) as well. It implies

$$\mathrm{Tr}(U_*^T H_A U_*) = \min_{U \in \mathbb{O}^{n\times r}} \mathrm{Tr}(U^T H_A U) = 0, \tag{3.12}$$

where the second equation is due to $\mathrm{Tr}(U^T H_A U) \equiv \mathrm{loss}_A(U) \geq 0$ by (3.2) with equality holding at the PCA solution $U = U_A$ for the matrix $A$. By (3.10) and (3.12), we have $0 > \mathrm{Tr}(U_*^T H_B U_*) \equiv \mathrm{loss}_B(U_*) \geq 0$, which is a contradiction. □

### 3.3 SDR-based algorithm

To solve FPCA (3.4), a semidefinite relaxation (SDR) approach was developed in [36]. The key ideas of their algorithm are summarized as follows. For consistency of notation and ease of comparison, we will present the SDR-based method using the trace expression of FPCA in (3.7), although the original method in [36] was formulated using the loss expression (3.1). The two formulations are mathematically equivalent.

By the identity $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$, the minimax problem (3.7) can be written in terms of the projection matrix $P = UU^T$ as

$$\min_{P=UU^T,\ U\in\mathbb{O}^{n\times r}} \max\left\{\mathrm{Tr}(H_A P),\ \mathrm{Tr}(H_B P)\right\}. \tag{3.13}$$

Noticing that the objective function of the minimization, $\max\left\{\mathrm{Tr}(H_A P),\ \mathrm{Tr}(H_B P)\right\}$, is convex in $P$, the problem can be transformed into a convex optimization by *relaxing* the constraint $P = UU^T$ to the larger convex set $\{P \in \mathbb{R}^{n\times n} :\ \mathrm{Tr}(P) \le r,\ 0 \preceq P \preceq I\}$, where $X \preceq Y$ denotes that $Y - X$ is positive semidefinite. This relaxation leads to a semidefinite programming (SDP):

$$\begin{aligned} &\min_{P\in\mathbb{R}^{n\times n}} \max\left\{\mathrm{Tr}(H_A P),\ \mathrm{Tr}(H_B P)\right\} \\ &\text{subject to } \mathrm{Tr}(P) \le r,\ 0 \preceq P \preceq I, \end{aligned} \tag{3.14}$$

which can be solved using existing convex optimization techniques [8]. Once $P$ is computed, a linear programming (LP) step is applied to *correct* the rank of $P$.

The formulation (3.14) is essentially the SDR problem solved in [36, Alg. 1], though we use a different expression of the objective functions through the matrices $H_A$ and $H_B$. This SDR-based approach has two major drawbacks. First, the algorithm produces an approximate projection $\widehat{P} \in \mathbb{R}^{n\times n}$, instead of a basis matrix $U \in \mathbb{O}^{n\times r}$. Due to SDR and computational error, the resulting $\widehat{P}$ may fail to recover the orthogonal projection $UU^T$ of rank $r$. Second, this approach is expensive, with a theoretical runtime of $O\left(n^3/\mathrm{tol}^2\right)$, where tol is the error tolerance of the SDP and LP. As reported in [36], the runtime is "*at most* 10 *to* 15 *times*" slower than the standard PCA.

### 3.4 Hidden convexity

In this section, we uncover a hidden convexity of FPCA (3.7) through a suitable change of variables. This allows us to visualize the minimax optimization as a two-dimensional optimization problem and leads to a reformulation of the problem as a convex optimization that facilitates the computation of the optimal solution.

Recall that the $r$-th joint numerical range of a pair of symmetric $S, T \in \mathbb{R}^{n\times n}$ is defined as

$$\mathcal{W}_r(S, T) = \left\{ \begin{bmatrix} \mathrm{Tr}(U^T SU) \\ \mathrm{Tr}(U^T TU) \end{bmatrix} \in \mathbb{R}^2 : U \in \mathbb{O}^{n\times r} \right\}.$$

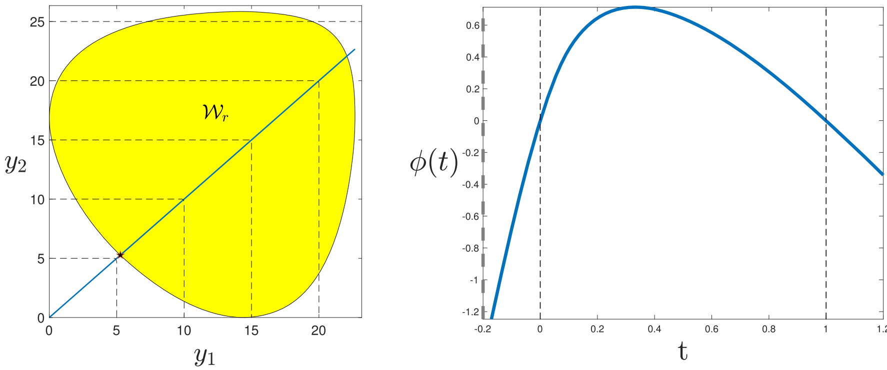


**Fig. 2** **Left:** Geometric illustration of FPCA (3.7): The yellow region represents the joint numerical range $\mathcal{W}_r(H_A, H_B)$; The dashed lines represent contours of the 'max' function, i.e., solution of $\max\{y_1, y_2\} = c$ for a given constant $c$; The solid blue line is with $y_1 = y_2$; The star marks the optimal solution $y_*$ of (3.7). **Right:** An illustration of the eigenvalue function $\phi(t)$ defined in Theorem 4.1

Since $\mathbb{O}^{n\times r}$ is a closed set and $\mathrm{Tr}(\cdot)$ is a continuous function, the $\mathcal{W}_r(S, T)$ is a bounded and closed subset of $\mathbb{R}^2$ [25]. Moreover, it is also well known that $\mathcal{W}_r(S, T)$ is a convex subset of $\mathbb{R}^2$ when the matrix size $n > 2$ [1].

Using the joint numerical range, we can rewrite FPCA (3.7) as the minimization problem

$$\min_{y\in\mathcal{W}_r(H_A, H_B)} \max\{y_1, y_2\}, \tag{3.15}$$

where

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \equiv \begin{bmatrix} \mathrm{Tr}(U^T H_A U) \\ \mathrm{Tr}(U^T H_B U) \end{bmatrix}, \quad \text{for some } U \in \mathbb{O}^{n\times r}. \tag{3.16}$$

Observe that the objective function $\max\{y_1, y_2\}$ is a convex function in $y \in \mathbb{R}^2$; see, e.g., [8, pp.72]. Moreover, the feasible set $\mathcal{W}_r(H_A, H_B)$ is a convex set under the general assumption of $n > 2$ [1]. Therefore, the optimization problem (3.15) is a convex optimization when $n > 2$.[1]

The convex optimization problem (3.15) involves only two variables, $y_1$ and $y_2$. Together with the convexity of the joint numerical range $\mathcal{W}_r(H_A, H_B)$, it allows us to conveniently visualize the solution of FPCA. For example, the left plot of Figure 2 depicts $\mathcal{W}_r(H_A, H_B)$ for a pair of randomly generated matrices $A$ and $B$, along with the contours of the objective function $\max\{y_1, y_2\}$; see Section A.2 for the details on the generation of $\mathcal{W}_r(H_A, H_B)$. We can see that the optimal solution $y_*$ of (3.15) lies at the intersection of $\mathcal{W}_r(H_A, H_B)$ with the diagonal of the Cartesian plane, where $y_1 = y_2$, achieving fairness in the reconstruction loss.

From the visualization, we can also clearly see why a fair solution is always attainable, as stated in Theorem 3.1. For the matrices $(H_A, H_B)$ of FPCA (3.7), the joint numerical range $\mathcal{W}_r(H_A, H_B)$ lies entirely in the first quadrant and intersects both coordinate axes; see the left plot of Figure 2. This follows from (3.5) and (3.2), which

[1] The requirement $n > 2$ is not essential, as we can instead use the joint numerical range with complex orthogonal $U \in \mathbb{C}^{n\times r}$, which is always convex; see Section A.1.

imply that $y_1 = \mathrm{Tr}(U^T H_A U) = \mathrm{loss}_A(U) \geq 0$ for all $U \in \mathbb{O}^{n\times r}$ with equality at the PCA solution $U = U_A$ for the matrix $A$, and $y_2 = \mathrm{Tr}(U^T H_B U) = \mathrm{loss}_B(U) \geq 0$ with equality at $U = U_B$. Consequently, $\mathcal{W}_r(H_A, H_B)$ must intersect the diagonal line $y_1 = y_2$, and this intersection corresponds to the solution of FPCA (3.7).

## 4 FPCA via eigenvalue optimization

In this section, we present a new algorithm for FPCA (3.7) based on convex optimization (3.15) and a univariate eigenvalue optimization. This offers a more efficient and reliable alternative to the existing SDR-based approach discussed in Section 3.3.

Motivated by the geometric illustration in the left plot of Figure 2, one can locate the optimal $y_*$ of (3.15) by searching along the boundary of the joint numerical range $\mathcal{W}_r(H_A, H_B)$. The following theorem shows that this search can be conveniently expressed as an eigenvalue optimization problem for a symmetric matrix-valued function

$$H(t) = tH_A + (1-t)H_B, \quad t \in [0,1]. \tag{4.1}$$

**Theorem 4.1** *The optimal solution of FPCA (3.15) is*

$$y_* = \begin{bmatrix} \mathrm{Tr}(U_*^T H_A U_*) \\ \mathrm{Tr}(U_*^T H_B U_*) \end{bmatrix}, \tag{4.2}$$

*where $U_* \in \mathbb{O}^{n\times r}$ is a basis matrix for the eigenspace associated with the $r$ smallest eigenvalues of $H(t_*)$, and the $t_*$ is a solution of the eigenvalue optimization*

$$\max_{t\in[0,1]} \left\{ \phi(t) := \sum_{i=1}^{r} \lambda_i(H(t)) \right\}, \tag{4.3}$$

*where $H(t)$ is defined in (4.1).*

***Proof*** We prove the theorem for the general cases with the matrix size $n > 2$; the special cases with $n \leq 2$ are addressed in Section A.1. We begin by reformulating FPCA (3.15) as

$$\begin{aligned} \min_{y\in\mathcal{W}_r(H_A,H_B)} \max\{y_1, y_2\} &= \min_{y\in\mathcal{W}_r(H_A,H_B)} \max_{t\in[0,1]} \left[t\cdot y_1 + (1-t)\cdot y_2\right] \\ &= \max_{t\in[0,1]} \min_{y\in\mathcal{W}_r(H_A,H_B)} \left[t\cdot y_1 + (1-t)\cdot y_2\right], \end{aligned} \tag{4.4}$$

where the first equality follows from a direct verification, and the second equality is an application of a generalized von Neumann's minimax theorem [34] – This theorem applies here because the objective function is affine in $y$ and $t$, respectively, and both feasible sets, $\mathcal{W}_r(H_A, H_B)$ and $[0, 1]$, are convex.

We observe that the inner minimization problem in (4.4) has a closed-form solution:

$$\min_{y\in\mathcal{W}_r(H_A,H_B)} \left[t\cdot y_1+(1-t)\cdot y_2\right] = \min_{U\in\mathbb{O}^{n\times r}} \left[t\cdot \mathrm{Tr}(U^T H_A U)+(1-t)\cdot \mathrm{Tr}(U^T H_B U)\right]$$

$$= \min_{U\in\mathbb{O}^{n\times r}} \mathrm{Tr}(U^T H(t) U) = \sum_{i=1}^{r} \lambda_i(H(t)), \tag{4.5}$$

where the first equation is by a parameterization of $y=\left[\mathrm{Tr}(U^T H_A U), \mathrm{Tr}(U^T H_B U)\right]^T$ for some $U\in\mathbb{O}^{n\times r}$, the second equation is by the definition of $H(t)$ in (4.1) and the last equation is by Ky Fan's eigenvalue minimization principle [11], which implies the minimal trace is given by the sum of the $r$ smallest eigenvalues of $H(t)$. Hence, plugging (4.5) into the inner minimization of (4.4), we write FPCA (3.15) as an eigenvalue optimization problem:

$$\min_{y\in\mathcal{W}_r(H_A,H_B)} \max\{y_1,y_2\} = \max_{t\in[0,1]} \phi(t), \tag{4.6}$$

where $\phi(t)=\sum_{i=1}^{r}\lambda_i(H(t))$.

Next, we examine the relation between the optimal solutions $y_*$ and $t_*$ of the two optimization problems in (4.6). From (4.6), we have

$$\phi(t_*) = \max\{y_{*1}, y_{*2}\} \geq t_* y_{*1} + (1-t_*)y_{*2} \geq \phi(t_*),$$

where the first inequality is due to $t_*\in[0,1]$ and the second is due to (4.5) with $t=t_*$. Since both inequalities must be equalities, we obtain

$$t_* y_{*1} + (1-t_*)y_{*2} \equiv \phi(t_*).$$

By writing $y_*=\left[\mathrm{Tr}(U_*^T H_A U_*), \mathrm{Tr}(U_*^T H_B U_*)\right]^T$ for some $U_*\in\mathbb{O}^{n\times r}$, we have

$$\mathrm{Tr}(U_*^T H(t_*) U_*) = \sum_{i=1}^{r}\lambda_i(H(t_*)),$$

which, according to Ky Fan's eigenvalue minimization principle [11], implies $U_*$ must be an eigenbasis for the $r$ smallest eigenvalues of $H(t_*)$. □

The following result shows that the objective eigenvalue function $\phi(t)$ in the problem optimization (4.3) is concave on the interval [0, 1]; see the right plot of Figure 2 for an illustration. Consequently, the maximizer of $\phi(t)$ over [0, 1] can be computed efficiently using classical methods such as the golden-section search.

**Lemma 4.1** *The function $\phi(t)$ defined in (4.3) is continuous, piecewise smooth, and concave on the interval* [0, 1].

***Proof*** The piecewise smoothness of the function $\phi(t)$ follows from a classical result in eigenvalue perturbation analysis, which shows that the $i$-th eigenvalue $\lambda_i(H)$ of a symmetric matrix $H$ is a piecewise analytic function of the entries of $H$; see, e.g., [33, Chap.1].

The concavity of $\phi(t)$ is also well known in convex analysis. In our case, it can be quickly verified by (4.5), which shows that

$$\phi(t) = \min_{y \in \mathcal{W}_r(H_A, H_B)} \Big\{ \phi_y(t) := t \cdot y_1 + (1-t) \cdot y_2 \Big\},$$

namely, $\phi(t)$ is the pointwise minimum of a set of functions $\{\phi_y(t)\}$. Since each $\phi_y(t)$ is concave in $t$, their pointwise minimum $\phi(t)$ is also concave; see, e.g., [8, Sec.3.2.3]. □

**Algorithm 1** EigOpt

**Input:** $A \in \mathbb{R}^{m_1 \times n}$, $B \in \mathbb{R}^{m_2 \times n}$, number of principal components $r$, tolerance tol.
**Output:** solution $U_* \in \mathbb{R}^{n \times r}$ of FPCA (3.7).

1: Compute the largest $r$ singular values of $A$ and $B$, respectively, as to be used to construct the matrices $H_A$ and $H_B$ in (3.8) *implicitly*.
2: Compute $t_* \in [0, 1]$ that maximizes $\phi(t)$ in (4.3), using the error tolerance tol.
3: Compute $U_*$, the eigenvectors corresponding to the $r$ smallest eigenvalues of the matrix $H(t_*) = t_* H_A + (1 - t_*) H_B$.

Algorithm 1 outlines the proposed method for FPCA (3.7) via eigenvalue optimization, referred to as EigOpt. Several remarks are given below.

(a) In steps 2 and 3, we need to compute the $r$ smallest eigenvalues of the symmetric matrix $H(t) = tH_A + (1-t)H_B$ for a given $t \in [0, 1]$. If the matrix $H(t)$ is of a moderate size, we form $H(t)$ explicitly, compute all its eigenvalues, and select the $r$ smallest ones.
For a large-scale $H(t)$, we can apply a Krylov subspace method to find its smallest $r$ eigenvalues [2, 5, 14]. In this case, we only need to supply the matrix-vector multiplications $u = H(t) \cdot v$ without explicitly formulating $H(t)$. We note that by the definitions of $H_A$ and $H_B$ in (3.8),

$$H(t) = \Big(t\gamma_A + (1-t)\gamma_B\Big) I_n - \Big(\tfrac{t}{m_1} A^T A + \tfrac{1-t}{m_2} B^T B\Big), \tag{4.7}$$

where $\gamma_A = \frac{1}{m_1 r} \sum_{i=1}^r \sigma_i^2(A)$, $\gamma_B = \frac{1}{m_2 r} \sum_{i=1}^r \sigma_i^2(B)$.
Computing $u = H(t) \cdot v$, therefore, requires four matrix-vector multiplications with the matrices $A$, $B$, $A^T$, and $B^T$, together with three scalar multiplications and two vector additions of length-$n$.

(b) For the eigenvalue optimization in step 2, we can apply Brent's method (see, e.g., [6, Sec.5.4]), which combines the golden-section search with parabolic interpolation. Brent's method is *derivative-free*, *globally convergent*, and *locally superlinearly convergent*. It is also available as MATLAB's built-in function `fminbd`, allowing step 2 to be implemented conveniently in just a few lines of code as shown below. The tolerance parameters for `eigs` and `fminbnd` maybe adjusted if needed:

```
H = @(t) t*HA + (1-t)*HB;
phi = @(t) sum(eigs(H(t), r, 'smallestreal'));
t_star = fminbnd(@(t) -phi(t), 0, 1);
```

(c) In step 3, we assume $\lambda_{r+1}(H(t_*)) > \lambda_r(H(t_*))$, i.e., a gap exists between the two eigenvalues. In this case, the eigenspace corresponding to the $r$ smallest eigenvalues of $H(t_*)$ is unique, and then any orthogonal basis matrix $U_* \in \mathbb{O}^{n\times r}$ can be used to define the $y_*$ by (4.2) – The choice of the orthogonal basis $U_*$ is irrelevant, since $\mathrm{Tr}(U^T H_A U)$ and $\mathrm{Tr}(U^T H_B U)$ are invariant under a right multiplication of $U$ by any $Q \in \mathbb{O}^{n\times r}$. However, if $\lambda_{r+1}(H(t_*)) = \lambda_r(H(t_*))$, the eigenspace of the $r$ smallest eigenvalues is not unique, and a particular basis $U_*$ must be selected (see Section A.3 for details). We note that such cases of repeated eigenvalues for $\lambda_r$ are expected to be rare in practice and were not observed in our experiments.

(d) Using dense linear algebra solvers, the overall complexity of Algorithm 1 is

$$O(n^3(\log_2(\mathrm{tol}))^2 + mn^2), \tag{4.8}$$

where tol is the absolute error tolerance for computing $t_*$ in the eigenvalue optimization of step 2 using Brent's method. This complexity consists of the following major components:

- In step 1, the SVD of $A$ and $B$ requires $O(m_1n^2) + O(m_2n^2)$ time complexity – Recall that the SVD of an $m \times n$ matrix has a complexity $O(mn^2)$ [14, pp.493]. In addition, forming $H_A$ and $H_B$ by (3.8) requires $O(m_1n^2)+O(m_2n^2)$ operations;
- In step 2, the eigenvalue optimization by Brent's method, with an absolute error tolerance tol, requires $O([\log_2(\mathrm{tol})]^2)$ evaluations of $\phi(t)$ over the interval [0, 1]; see [6, Sec.5.4]. Each evaluation of $\phi(t)$ requires the solution of a symmetric eigenvalue problem of size $n$, which has a complexity $O(n^3)$.

Recall that the SDR-based algorithm of [36] has a complexity of $O\left(n^{6.5}\log(1/\mathrm{tol})\right)$ if the SDP is solved by conventional convex optimization, and $O\left(n^3/\mathrm{tol}^2\right)$ if solved by the multiplicative weight (MW) update method, where tol is the error tolerance of the SDP and LP. Those complexities are substantially higher than that of (4.8).

## 5 Numerical experiments

In this section, we demonstrate the performance of EigOpt (Algorithm 1) on real-world datasets and compare it with standard PCA and the SDR-based FPCA.[2] We consider the following four datasets:

[2] The testing code is available at https://github.com/JunhuiShen/Fair-PCA-via-EigOpt. The implementation of the SDR-based FPCA algorithm is from https://github.com/samirasamadi/Fair-PCA.

1. Bank Marketing (BM): Introduced in [27] to analyze the success of direct marketing campaigns to promote term deposit subscriptions at a Portuguese bank.[3] The dataset is divided into two age groups: $A \in \mathbb{R}^{810\times 16}$, representing the younger individuals, and $B \in \mathbb{R}^{44401\times 16}$, representing the older. It has been widely used in fairness research, including clustering [3] and PCA [20, 31].
2. Default of Credit Card Clients (DCC): Introduced in [42] to investigate default payment behavior in Taiwan.[4] The data is divided by education level into two groups: $A \in \mathbb{R}^{10599\times 23}$ for graduate degree holders and $B \in \mathbb{R}^{19401\times 23}$ for other education levels. It has been used in fair PCA studies [21, 30, 31, 36].
3. Crop Mapping (CM):
   From the UCI Machine Learning Repository[5], collected in Manitoba, Canada for cropland classification [12, 39]. The dataset is divided by crop type into two groups: $A \in \mathbb{R}^{39162\times 173}$ for corn and $B \in \mathbb{R}^{286672\times 173}$ for other crops.
4. Labeled Face in the Wild (LFW):
   Contains face photographs and is widely used to study unconstrained face recognition [15, 35, 38, 40].[6] It is often used in fairness research, particularly in fair PCA studies [30, 36, 37]. The dataset is divided by gender, with $A \in \mathbb{R}^{2962\times 1764}$ for females and $B \in \mathbb{R}^{10270\times 1764}$ for males.

All experiments were conducted on a MacBook Pro with a 12-core M2 Max processor @3.49GHz, 32GB of RAM, and 48MB of L3 cache.

***Example 5.1*** We first compare FPCA with standard PCA in terms of the *reconstruction error* and *reconstruction loss* of the computed basis $\widehat{U}_r \in \mathbb{R}^{n\times r}$ for dimension reduction. The solution of FPCA (3.7) is computed by EigOpt (Algorithm 1).

The left panel of Figure 3 reports the overall reconstruction error of the data matrix $M$, measured by $\|M - M\widehat{U}_r\widehat{U}_r^T\|_F^2$, as a function of the reduced dimension $r$. As expected, the errors decrease monotonically as $r$ increases. We observe that FPCA exhibits slightly larger reconstruction errors than the standard PCA, with increases ranging from about 0.01% to 20.44% across datasets. This reflects the trade-off between accuracy and fairness in the dimensionality reduction.

The right panel of Figure 3 depicts the reconstruction losses for groups $A$ and $B$, measured respectively by $\text{loss}_A(\widehat{U}_r) = \text{Tr}(\widehat{U}_r^T H_A \widehat{U}_r)$ and $\text{loss}_B(\widehat{U}_r) = \text{Tr}(\widehat{U}_r^T H_B \widehat{U}_r)$. Standard PCA results in significant disparities, with the group $B$ showing much higher loss than the group $A$. In contrast, FPCA computed by EigOpt consistently achieves equity in the losses between the groups. In all test cases, we observed that the FPCA solution $\widehat{U}_r$ satisfies

$$\left|\frac{\text{loss}_A(\widehat{U}_r)}{\text{loss}_B(\widehat{U}_r)} - 1\right| \leq 10^{-5}, \tag{5.1}$$

which confirms the fairness property (3.3) established in Theorem 3.1 and indicates high accuracy of the solutions computed by EigOpt.

[3] https://archive.ics.uci.edu/dataset/222/bank+marketing.

[4] https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients.

[5] https://archive.ics.uci.edu/dataset/525/crop+mapping+using+fused+optical+radar+data+set.

[6] https://vis-www.cs.umass.edu/lfw/.

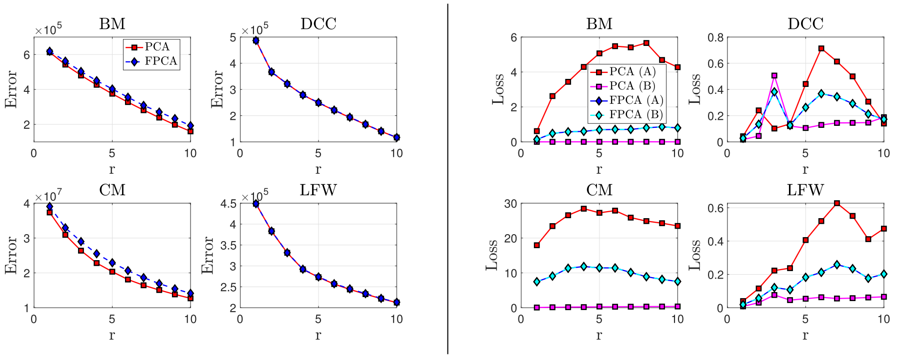


**Fig. 3** **Left:** reconstruction error $\|M - M\widehat{U}_r\widehat{U}_r^T\|_F^2$. **Right:** reconstruction loss for groups $A$ and $B$

***Example 5.2*** We now compare the runtime of the three PCA approaches: (a) standard PCA; (b) FPCA by EigOpt (Algorithm 1); and (c) FPCA by the SRD-based algorithm [36]. The results are shown in Figure 4. We observe that the runtime of EigOpt is close to that of standard PCA, with a slowdown ranging only from about 4.79% to 85.81%. In comparison, the SDR-based FPCA algorithm takes substantially more time – approximately $8\times$ slower than both standard PCA and EigOpt.

In terms of accuracy, all FPCA solutions computed by EigOpt satisfy (5.1), achieving fair reconstruction losses. In contrast, the SDR-based algorithm produces only suboptimal solutions. The implementation in [36] employs a Multiplicative Weight (MW) update method for the semidefinite programming, using a fixed number $T$ of iterations (a tunable parameter). For all test cases, the corresponding error in the loss ratio, measured by $\left|\text{loss}_A/\text{loss}_B - 1\right|$, is shown in the right panel of Figure 4. We observe that, despite its higher computational cost, the SDR-based algorithm exhibits significantly larger deviations from the fairness criterion than EigOpt. We also note that the SDR-based algorithm produces an approximate projection matrix $P = UU^T$ rather than the basis matrix $U$ directly. In several experiments, the computed $\widehat{P}$ had a numerical rank exceeding $r$.

These results demonstrate that EigOpt provides a reliable way to compute the FPCA solution at a cost comparable to standard PCA. This is particularly significant as it allows solving the FPCA model (3.4) directly – without resorting to alternative models for computational convenience; see Section B for a comparison with several alternative eigenvalue-based fair PCA approaches.

## 6 Concluding remarks

We presented a novel eigenvalue optimization approach for solving the FPCA problem (3.4) by revealing a hidden convexity through its reformulation as an optimization over the joint numerical range. The experiments demonstrated that the proposed method is efficient, reliable, easy to implement, and reduces the computational cost of FPCA to a level comparable to standard PCA.

| Dataset ($r$) | PCA | EigOpt | SDR |
|---|---|---|---|
| BM (2) | 0.01 | 0.01 | 0.10 |
| BM (4) | 0.01 | 0.01 | 0.10 |
| BM (8) | 0.01 | 0.01 | 0.11 |
| DCC (5) | 0.01 | 0.01 | 0.10 |
| DCC (10) | 0.01 | 0.01 | 0.10 |
| DCC (15) | 0.01 | 0.01 | 0.10 |
| CM (30) | 1.19 | 1.27 | 13.73 |
| CM (60) | 1.05 | 1.23 | 12.97 |
| CM (120) | 1.14 | 1.26 | 13.12 |
| LFW (50) | 4.82 | 6.68 | 53.99 |
| LFW (100) | 5.79 | 6.95 | 56.86 |
| LFW (200) | 4.68 | 8.71 | 58.20 |

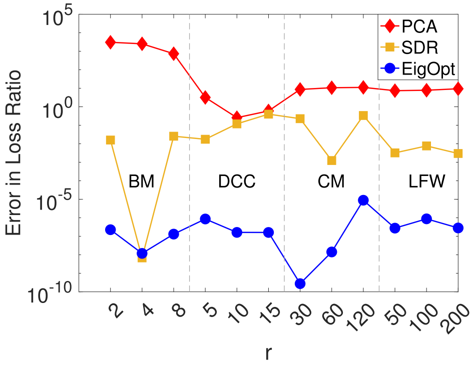


**Fig. 4** Comparison of standard PCA, FPCA via EigOpt (Algorithm 1), and FPCA via SDR-based algorithm by [36]. **Left:** runtime (in seconds); **Right:** error in the loss ratio as measured by $\left|\frac{\text{loss}_A}{\text{loss}_B} - 1\right|$

As an immediate extension, the proposed algorithm can be applied directly to the FPCA model (3.4) with three subgroups, since the the joint numerical range of three symmetric matrices remains convex when the matrix size $n > 2$ [13]. However, extending the method to four or more groups remains an open problem, as the convexity property may no longer hold.

# A Technical notes

## A.1 Fair PCA and EigOpt for matrices of size $n = 1$ or 2

The proof for Theorem 4.1 requires the matrix size $n \geq 3$. We now address the cases with $n = 1$ and 2. For the cases $n = r = 1$ and $n = r = 2$, it follows directly from the definitions of $H_A$ and $H_B$ in (3.8) that

$$\mathrm{Tr}(U^T H_A U) \equiv \mathrm{Tr}(U^T H_B U) \equiv 0,$$

for all $U \in \mathbb{O}^{n\times r}$. This implies $\mathcal{W}_r(H_A, H_B) = \{0\}$, a convex set, so the proof in Theorem 4.1 still holds.

It remains to consider the case where $n = 2$ and $r = 1$. In this case, the joint numerical range $\mathcal{W}_r(H_A, H_B)$ must form a *general ellipse* (i.e., either an ellipse, a circle, a line segment, or a point) [7]. Therefore, $\mathcal{W}_1(H_A, H_B)$ consists of the boundary points of its convex hull $\mathrm{Conv}(\mathcal{W}_1(H_A, H_B))$, the smallest convex set that contains $\mathcal{W}_1(H_A, H_B)$. Hence, the optimization (3.15) can be solved over this convex hull as

$$\min_{y\in\mathcal{W}_1(H_A,H_B)} \max\{y_1, y_2\} = \min_{y\in\mathrm{Conv}(\mathcal{W}_1(H_A,H_B))} \max\{y_1, y_2\}, \tag{A1}$$

where we used the fact that $g(y) := \max\{y_1, y_2\}$ has no local minimizer in $\mathbb{R}^2$, so the minimizer of $g$ over $\mathrm{Conv}(\mathcal{W}_1(H_A, H_B))$ must occur on its boundary, $\mathcal{W}_1(H_A, H_B)$.

It is well-known that the convex hull of $\mathcal{W}_1(H_A, H_B)$ is exactly its complex analogue:

$$\mathrm{Conv}(\mathcal{W}_1(H_A, H_B)) \equiv \mathcal{W}_1^{\mathbb{C}}(H_A, H_B), \tag{A2}$$

with

$$\mathcal{W}_1^{\mathbb{C}}(H_A, H_B) := \left\{ \begin{bmatrix} \mathrm{Tr}(U^H H_A U) \\ \mathrm{Tr}(U^H H_B U) \end{bmatrix} : U \in \mathbb{C}^{n\times 1}, U^H U = 1 \right\}, \tag{A3}$$

where $\cdot^H$ denotes conjugate transpose, and the superscript $\mathbb{C}$ in $\mathcal{W}_1^{\mathbb{C}}$ is to distinguish it from the previous definition of joint numerical range $\mathcal{W}_r$ in Section 3.4, where $U$ is a real matrix; see, e.g., [7]. Consequently, we can write (A1) as

$$\min_{y\in\mathcal{W}_1(H_A,H_B)} \max\{y_1, y_2\} = \min_{y\in\mathcal{W}_1^{\mathbb{C}}(H_A,H_B)} \max\{y_1, y_2\}. \tag{A4}$$

Since $\mathcal{W}_1^{\mathbb{C}}(H_A, H_B)$ is a convex set, the proof in Theorem 4.1 applies to the optimization in (A4), thus establishing the eigenvalue optimization in Theorem 4.1.

### A.2 Generating the joint numerical range

For visualization, the convex joint numerical range $\mathcal{W}_r(S, T)$ can be generated by sampling its boundary points, each of which can be obtained by solving a symmetric eigenvalue problem. This approach follows existing methods for computing the *numerical range of a square matrix*; see, e.g., [19]. For completeness, we derive the corresponding eigenvalue problem and outline the sampling procedure below.

Given a search direction $v := [\cos(\theta), \sin(\theta)]^T \in \mathbb{R}^2$ with an angle $\theta \in (0, 2\pi]$, the boundary point $y_\theta$ of $\mathcal{W}_r(S, T)$ with an outer normal vector $v$ can be obtained by solving the optimization problem

$$y_\theta = \operatorname*{arg\,max}_{y\in\mathcal{W}_r(S,T)} v^T y. \tag{A5}$$

By parameterizing $y \in \mathcal{W}_r(S, T)$ as $y = [\mathrm{Tr}(U^T SU), \mathrm{Tr}(U^T TU)]^T$ for some $U \in \mathbb{O}^{n\times r}$, the maximization problem (A5) leads to

$$\max_{y\in\mathcal{W}_r(S,T)} v^T y = \max_{U\in\mathbb{O}^{n\times r}} \mathrm{Tr}\Big(U^T[\cos(\theta)\cdot S + \sin(\theta)\cdot T]U\Big).$$

By Ky Fan's trace maximization principle [11], the solution $U_\theta$ to the trace maximization above is given by an orthogonal eigenbasis corresponding to the largest $r$ eigenvalues of the matrix

$$B(\theta) := \cos(\theta)S + \sin(\theta)T. \tag{A6}$$

Consequently, the boundary point along the direction $v$ is given by

$$y_\theta = [\mathrm{Tr}(U_\theta^T SU_\theta), \mathrm{Tr}(U_\theta^T TU_\theta)]^T.$$

By searching in different directions, such as using equally spaced angles $\theta_j$ in $(0, 2\pi]$, we can sample a finite number of boundary points of $\mathcal{W}_r(S, T)$. The convex hull of these boundary points generates an approximate joint numerical range, as shown in Figure 2. The overall procedure is summarized in Algorithm 2.

**Algorithm 2** Generating the joint numerical range $\mathcal{W}_r(S, T)$

**Input:** Symmetric $S, T \in \mathbb{R}^{n\times n}$, dimension $r$, and number of angle samples $\ell$.
**Output:** Approximate $\mathcal{W}_r(S, T)$ by the convex hull of boundary points $\{y_1, \ldots, y_\ell\}$.
1: Set step size for search angles in $(0, 2\pi]$ as $h = 2\pi/\ell$;
2: **for** $j = 1, 2, \ldots, \ell$ **do**
3: Set the search angle $\theta = jh$;
4: Compute orthogonal eigenbasis $U_\theta$ for the largest $r$ eigenvalues of $B(\theta)$ by (A6);
5: Compute the boundary point $y_j = [\mathrm{Tr}(U_\theta^T SU_\theta), \mathrm{Tr}(U_\theta^T TU_\theta)]^T$;
6: **end for**
7: Return the convex hull of $\{y_1, \ldots, y_\ell\}$ as an approximation of $\mathcal{W}_r(S, T)$.

### A.3 Selection of the eigenvectors with repeated eigenvalues

Recall from Theorem 4.1 that if $\lambda_r(H(t_*)) = \lambda_{r+1}(H(t_*))$, then the eigenspace for the smallest $r$ eigenvalues of $H(t_*)$ is not uniquely defined. While Theorem 4.1 guarantees the existence of an eigenspace corresponding to the Fair PCA solution, the computed matrix $\widehat{U} \in \mathbb{O}^{n\times r}$, which may correspond to an arbitrary choice within those eigenspaces, may not satisfy the fairness condition $\mathrm{Tr}(\widehat{U}^T H_A \widehat{U}) = \mathrm{Tr}(\widehat{U}^T H_B \widehat{U})$. Consequently, in the case of repeated eigenvalues, an extra postprocessing step is required to select the FPCA solution from the computed ones.

Assume that $H(t_*)$ has eigenvalues ordered as

$$\lambda_1 \le \cdots \le \lambda_p < \underbrace{\lambda_{p+1} = \ldots \lambda_r = \lambda_{r+1} = \cdots = \lambda_{p+q}}_{q \text{ times}} < \lambda_{p+1+1},$$

where the $r$-th eigenvalue has multiplicity $q$. We partition the eigenvectors accordingly:

$$[U_1, U_2] \in \mathbb{O}^{n\times(p+q)} \quad \text{with } U_1 \in \mathbb{O}^{n\times p} \text{ and } U_2 \in \mathbb{O}^{n\times q}, \tag{A7}$$

where $U_1$ corresponds to the first $p$ eigenvalues, and $U_2$ to the repeated eigenvalues.

Since the FPCA solution $U_*$ contains eigenvectors corresponding to the smallest $r$ eigenvalues of $H(t_*)$, we construct $U_*$ as

$$U_* = [U_1, U_2 V] \in \mathbb{O}^{n\times r} \qquad \text{for some } V \in \mathbb{O}^{q\times(r-p)}. \tag{A8}$$

Our goal is to find a particular $V$ such that the fairness condition is satisfied:

$$0 = \mathrm{Tr}(U_*^T H_A U_*) - \mathrm{Tr}(U_*^T H_B U_*) = \mathrm{Tr}(U_*^T [H_A - H_B] U_*) = \gamma + \mathrm{Tr}(V^T C V), \tag{A9}$$

where $\gamma = \mathrm{Tr}(U_1^T(H_A - H_B)U_1)$ is a constant and

$$C = U_2^T(H_A - H_B)U_2 \in \mathbb{R}^{q\times q}$$

is the difference matrix $H_A - H_B$ projected onto the eigenspace of repeated eigenvalues.

Note that we only need a particular $V \in \mathbb{O}^{q\times(r-p)}$ that satisfies the condition (A9). This solution can be computed conveniently by a line search. First, the maximum and minimum values of the function

$$g(V) := \gamma + \mathrm{Tr}(V^T C V)$$

are respectively achieved by the eigenvectors $V_M \in \mathbb{O}^{q\times(r-p)}$ corresponding to the $r-p$ largest eigenvalues of $C$, and $V_m \in \mathbb{O}^{q\times(r-p)}$ corresponding to the $r-p$ smallest eigenvalues of $C$. We have the inequalities

$$g(V_m) \equiv \gamma + \mathrm{Tr}(V_m^T C V_m) \le 0 \le \gamma + \mathrm{Tr}(V_M^T C V_M) \equiv g(V_M), \tag{A10}$$

where Theorem 4.1 ensures that $0 \in [g(V_m), g(V_M)]$. Since $g(V_m)$ and $g(V_M)$ have opposite signs (unless one of them is 0, in which case the solution is trivial), we can search along a smooth curve $V(t)$ that connects $V_m$ and $V_M$ over the Grassmann manifold to find the solution $V$ such that $g(V) = 0$. For example, let

$$V(t) = \mathrm{orth}(t \cdot V_M + (1-t)V_m) \quad \text{for } t \in [0, 1], \tag{A11}$$

where $\mathrm{orth}(\cdot)$ extracts an orthogonal basis for the column space of a matrix, which can be computed using the Gram-Schmidt orthogonalization process (see, e.g., [4, 23]). Then $g(V(0)) = g(V_m) < 0$ and $g(V(1)) = g(V_M) > 0$, so we can apply bisection to the root-finding problem

$$g(V(t)) = 0 \quad \text{with} \quad t \in [0, 1] \tag{A12}$$

to find $\hat{t}$ such that $g(V(\hat{t})) = 0$. Here, we assume that $V(t)$ in (A11) has a full rank $r$ for all $t \in [0, 1]$, otherwise, a more sophisticated treatment is required, which is beyond the scope of this work.

Once the root $\hat{t}$ of (A12) is found, we can construct the Fair PCA solution as $U_* = [U_1, U_2 V(\hat{t})]$ using (A8). It is straightforward to verify that $U_*$ satisfies the fairness condition via (A9) and the optimality via (4.6).

## B Comparison with alternative fair PCA approaches

### B.1 Alternative eigenvalue-based fair PCA approaches

Two alternative fair PCA formulations are introduced in [31], which aim to minimize a weighted average of the total reconstruction error and the disparity between the (average) reconstruction errors of subgroups $A$ and $B$:

$$\min_{U\in\mathbb{O}^{n\times r}} \quad \alpha\cdot\frac{\|M-MUU^T\|_F^2}{m}+(1-\alpha)\cdot\left(\frac{\|B-BUU^T\|_F^2}{m_2}-\frac{\|A-AUU^T\|_F^2}{m_1}\right), \tag{B1}$$

where $\alpha\in[0,1]$ is a weighting parameter to be determined, and we assume $\|A-AU_MU_M^T\|/m_1\le\|B-BU_MU_M^T\|/m_2$ for the standard PCA solution $U_M$ for the entire data matrix $M=[A^T,B^T]^T$. It can be shown that for a given $\alpha$, the solution of the optimization (B1), denoted by $U_\alpha$, is given by the orthogonal eigenvectors corresponding to the $r$ largest eigenvalues of the symmetric matrix

$$M_\alpha\equiv\alpha\cdot\frac{1}{m}\left(M^TM\right)+(1-\alpha)\cdot\left(\frac{A^TA}{m_1}-\frac{B^TB}{m_2}\right). \tag{B2}$$

To determine the optimal weight $\alpha$, the following two models are proposed in [31]

1. **u-FPCA** selects $\alpha$ that minimizes the disparity in reconstruction errors:

$$\min_{\alpha\in[0,1]}\left(\frac{1}{m_2}\left\|B-BU_\alpha U_\alpha^T\right\|_F^2-\frac{1}{m_1}\left\|A-AU_\alpha U_\alpha^T\right\|_F^2\right)^2, \tag{B3}$$

   where $U_\alpha$ is solution to (B1).
2. **c-FPCA** is built on u-FPCA with two additional constraints:

$$\begin{aligned}\min_{\alpha\in[0,1]} \quad & \left(\frac{1}{m_2}\left\|B-BU_\alpha U_\alpha^T\right\|_F^2-\frac{1}{m_1}\left\|A-AU_\alpha U_\alpha^T\right\|_F^2\right)^2\\ \text{s.t.} \quad & \frac{1}{m_1}\left\|A-AU_\alpha U_\alpha^T\right\|_F^2\le\frac{1}{m_2}\left\|B-BU_MU_M^T\right\|_F^2,\\ & \frac{1}{m_2}\left\|B-BU_\alpha U_\alpha^T\right\|_F^2\le\frac{1}{m_2}\left\|B-BU_MU_M^T\right\|_F^2,\end{aligned} \tag{B4}$$

where $U_\alpha$ is a solution to (B1) and $U_M$ is a PCA solution for the entire set $M$. Both univariate optimization (B3) and (B4) can be solved by the golden-section search.

### B.2 Numerical experiments

The runtime of u-FPCA and c-FPCA[7] on Example 5.1 from Section 5 is reported in Figure 5 (similar to Figure 4), alongside standard PCA and FPCA via EigOpt for comparison. On average, EigOpt runs approximately 21× faster than u-FPCA and 23× faster than c-FPCA. From the corresponding loss ratios in the right panel of Figure 5, both u-FPCA and c-FPCA show larger deviations from fairness in the reconstruction loss between the subgroups. However, it should be noted that achieving fairness in the loss is not the primary objective of u-FPCA and c-FPCA.

[7] The code for u-FPCA and c-FPCA is available at https://github.com/GuilhermePelegrina/FPCA.

| Dataset ($r$) | PCA | EigOpt | u-FPCA | c-FPCA |
|---|---|---|---|---|
| BM (2) | 0.01 | 0.01 | 0.15 | 0.22 |
| BM (4) | 0.01 | 0.01 | 0.15 | 0.16 |
| BM (8) | 0.01 | 0.01 | 0.13 | 0.15 |
| DCC (5) | 0.01 | 0.01 | 0.17 | 0.20 |
| DCC (10) | 0.01 | 0.01 | 0.17 | 0.19 |
| DCC (15) | 0.01 | 0.01 | 0.16 | 0.19 |
| CM (30) | 1.05 | 1.11 | 38.97 | 39.03 |
| CM (60) | 1.12 | 1.18 | 38.78 | 38.89 |
| CM (120) | 1.04 | 1.19 | 39.64 | 38.85 |
| LFW (50) | 3.92 | 5.83 | 132.14 | 135.59 |
| LFW (100) | 4.06 | 6.60 | 125.31 | 131.54 |
| LFW (200) | 5.59 | 6.36 | 132.80 | 146.19 |

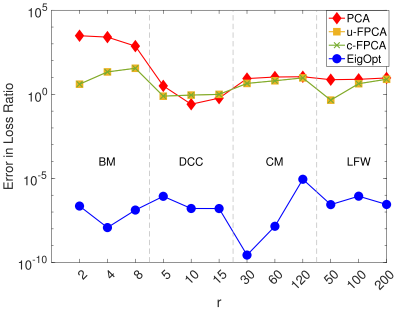


**Fig. 5** **Left:** runtime (in seconds); **Right:** error in the loss ratio as measured by $\left|\frac{\text{loss}_A}{\text{loss}_B} - 1\right|$

**Author Contributions** Conceptualization: Lu and Bai; Formal analysis and investigation: All; Writing – original draft preparation: Shen; Writing – review and editing: All; Resources: Publicly available datasets were used; Supervision: Lu and Bai.

**Data Availability** No datasets were generated or analysed during the current study.

# Declarations

**Competing interests** The authors declare no competing interests.

# References


1. Au-Yeung, Y., Tsing, N.: Some theorems on the generalized numerical ranges. Linear Multilinear Algebra **15**(1), 3–11 (1984)
2. Bai, Z., Demmel, J., Dongarra, J., Ruhe, A., Vorst (editors), H.: Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide. SIAM, Philadelphia (2000)
3. Bera, S., Deeparnab, C., Flores, N., Negahbani, M.: Fair algorithms for clustering. In: Proceedings of the Conference on Neural Information Processing Systems (2019)
4. Björck, Å.: Numerics of Gram-Schmidt orthogonalization. Linear Algebra Appl. **197**, 297–316 (1994)
5. Björck, Å.: Numerical Methods in Matrix Computations. Springer, Berlin (2015)
6. Brent, R.P.: Algorithms for Minimization Without Derivatives. Dover Publications, Mineola, NY, USA (2013)
7. Brickman, L.: On the field of values of a matrix. Proceedings of the American Mathematical Society **12**(1), 61–66 (1961)
8. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge, UK (2004)
9. Caton, S., Haas, C.: Fairness in machine learning: A survey. ACM Comput. Surv. **56**(7), 1–38 (2024)
10. Chouldechova, A., Roth, A.: A snapshot of the frontiers of fairness in machine learning. Commun. ACM **63**(5), 82–89 (2020)
11. Fan, K.: On a theorem of Weyl concerning eigenvalues of linear transformations I. Proc. Natl. Acad. Sci. U.S.A. **35**(11), 652–655 (1949). https://doi.org/10.1073/pnas.35.11.652
12. Femmam, K., Femmam, S.: Fast and efficient feature selection method using bivariate copulas. Journal of Advances in Information Technology **13**(3) (2022)
13. Gutkin, E., Jonckheere, E.A., Karow, M.: Convexity of the joint numerical range: topological and differential geometric viewpoints. Linear Algebra Appl. **376**, 143–171 (2004)
14. Golub, G., Van Loan, C.: Matrix Computations, 4th edn. Johns Hopkins University Press, Baltimore, MD, USA (2013)

15. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In: Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition (2008). https://hal.inria.fr/inria-00321923
16. Hotelling, H.: Analysis of a complex of statistical variables into principal components. J. Educ. Psychol. **24**(6), 417–441 (1933)
17. Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. In: Proceedings of the Conference on Neural Information Processing Systems (2016)
18. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer, New York, NY, USA (2009)
19. Johnson, C.R.: Numerical determination of the field of values of a general complex matrix. SIAM J. Numer. Anal. **15**(3), 595–602 (1978)
20. Kleindessner, M., Donini, M., Russell, C., Zafar, M.: Efficient fair PCA for fair representation learning. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (2023)
21. Kamani, M., Haddadpour, F., Forsati, R., Mahdavi, M.: Efficient fair principal component analysis. Mach. Learn. **111**, 3671–3702 (2022). https://doi.org/10.1007/s10994-021-06100-9
22. Kovač-Striko, J., Veselić, K.: Trace minimization and definiteness of symmetric pencils. Linear Algebra Appl. **216**, 139–158 (1995)
23. Leon, S.J., Björck, Å., Gander, W.: Gram-Schmidt orthogonalization: 100 years and more. Numerical Linear Algebra with Applications **20**(3), 492–532 (2013)
24. Lee, J., Cho, H., Yun, S., Yun, C.: Fair streaming principal component analysis: Statistical and algorithmic viewpoint. In: Proceedings of the International Conference on Neural Information Processing Systems (2023)
25. Li, C.-K.: C-numerical ranges and C-numerical radii. Linear and Multilinear Algebra **37**(1–3), 51–82 (1994)
26. Lee, J., Kim, G., Olfat, M., Hasegawa-Johnson, M., Yoo, C.: Fast and efficient MMD-based fair PCA via optimization over Stiefel manifold. In: Proceedings of the AAAI Conference on Artificial Intelligence (2022)
27. Moro, S., Cortez, P., Rita, P.: A data-driven approach to predict the success of bank telemarketing. Decis. Support Syst. **62**, 22–31 (2014)
28. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. ACM Comput. Surv. **54**(6), 1–35 (2021)
29. Olfat, M., Aswani, A.: Convex formulations for fair principal component analysis. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)
30. Pelegrina, G., Brotto, R., Duarte, L., Attux, R., Romano, J.: Analysis of trade-offs in fair principal component analysis based on multi-objective optimization. In: Proceedings of the International Joint Conference on Neural Networks, 1–8 (2022). https://doi.org/10.1109/IJCNN55064.2022.9892809
31. Pelegrina, G.D., Duarte, L.T.: A novel approach for fair principal component analysis based on eigendecomposition. IEEE Transactions on Artificial Intelligence **5**(3), 1195–1206 (2023)
32. Pearson, K.: 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)
33. Rellich, F.: Perturbation Theory of Eigenvalue Problems. CRC Press, New York, NY, USA (1969)
34. Sion, M.: On general minimax theorems. Pac. J. Math. **8**, 171–176 (1958)
35. Schlett, T., Rathgeb, C., Henniger, O., Galbally, J., Fierrez, J., Busch, C.: Face image quality assessment: A literature survey. ACM Computing Surveys **54** (2022)
36. Samadi, S., Tantipongpipat, U., Morgenstern, J.H., Singh, M., Vempala, S.: The price of Fair PCA: One extra dimension. In: Proceedings of the Conference on Neural Information Processing Systems (2018)
37. Tantipongpipat, U., Samadi, S., Singh, M., Morgenstern, J., Vempala, S.: Multi-criteria dimensionality reduction with applications to fairness. In: Proceedings of the Advances in Neural Information Processing Systems (2019)
38. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: DeepFace: closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014)
39. Vanishree, K., Halappanavar, N.H., Mukkannavar, S.R., Nishant, N.M., Nagaraja, G.S.: Comparative study of machine learning approaches on cropland mapping. In: Proceedings of the 2022 6th International Conference on Computation System and Information Technology for Sustainable Solutions, 1–5 (2022). IEEE

40. Wang, Z., Huang, B., Wang, G., Yi, P., Jiang, K.: Masked face recognition dataset and application. IEEE Transactions on Biometrics, Behavior, and Identity Science **5**(2), 298–304 (2023)
41. Xu, M., Jiang, B., Pu, W., Liu, Y.-F., So, A.M.-C.: An efficient alternating Riemannian/projected gradient descent ascent algorithm for fair principal component analysis. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (2024)
42. Yeh, I., Lien, C.: The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Syst. Appl. **36**, 2473–2480 (2009)