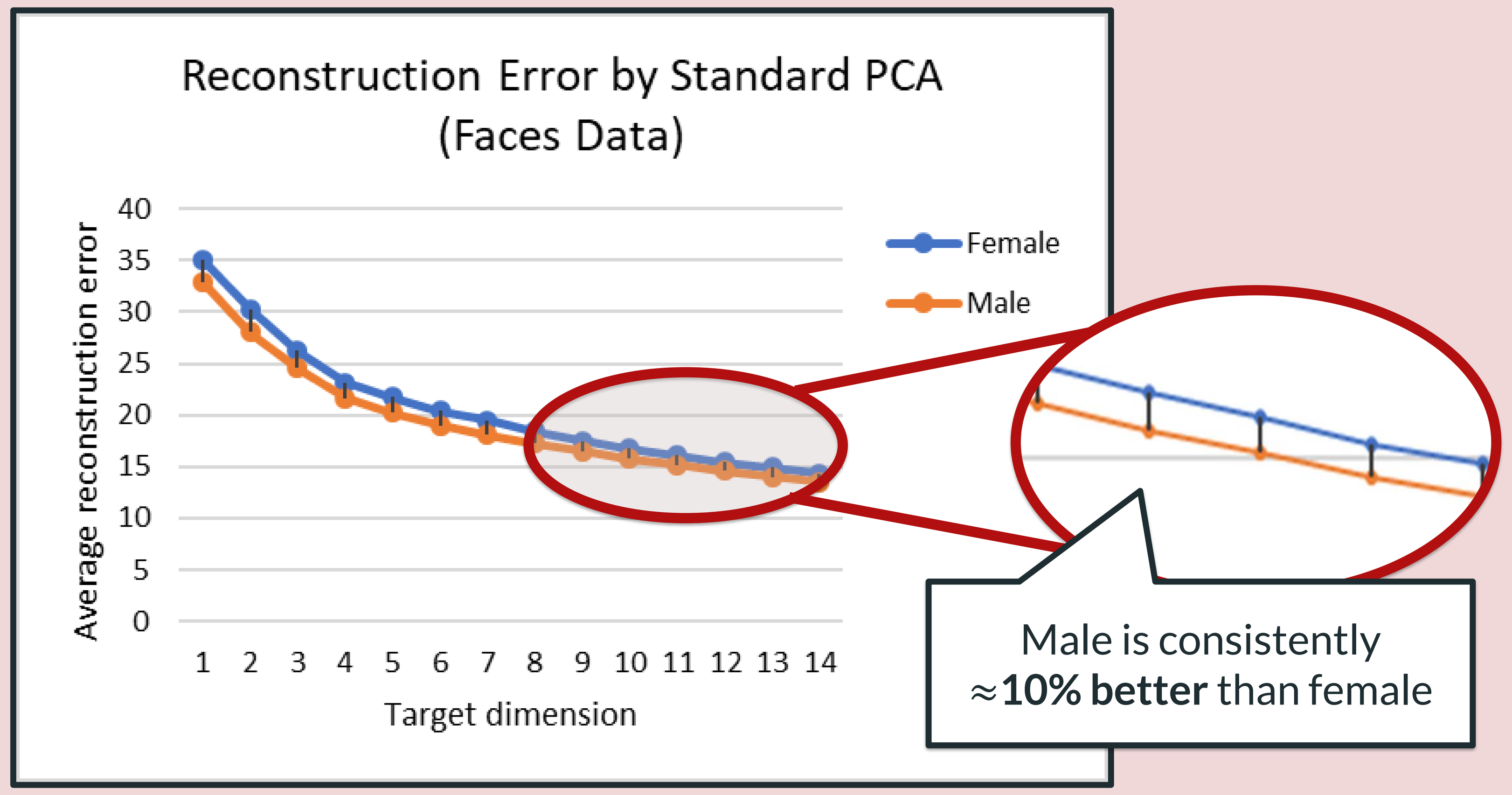
Multi-Criteria Dimensionality Reduction with Applications to Fairness

Uthaipon (Tao) Tantipongpipat, Samira Samadi, Mohit Singh, Jamie Morgenstern, and Santosh Vempala

Georgia Institute of Technology

Motivation

- Principle Component Analysis (PCA) is used in machine learning, natural sciences, and social sciences
- But PCA can result in **biased representation**



More challenges:

- Reweighting samples from each group to be equal does not fix the bias
- Two PCAs for each group are not allowed for ethical and legal reasons

Problem Formulation

- $A_1, \dots A_k$ are data of group $1, \dots, k$ (rows as entries).
- P is the orthonormal matrix for PCA projection to find.

We eliminate unfair representation by taking the worst group's performance as an objective (**Fair PCA**):

Standard PCA: $\max_P \sum_{i=1}^{k} \|A_i P\|_F^2$

Fair PCA: $\max_P \min_{i \in \{1,\dots,k\}} \|A_i P\|_F^2$

MCDR: $\max_P g(f_1(P), \dots, f_k(P))$

MCDR (Multi-Criteria Dimensionality Reduction) allows more flexibility for each group to choose their own utility criterion f_i and the central analyst to choose the utility aggregation g.

Example:

Marginal Loss (Mar-Loss): $\min_P \max_{i \in \{1,\dots,k\}} \left(\max_Q \|A_i Q\|_F^2 - \|A_i P\|_F^2 \right)$

Nash Social Welfare (NSW): $\max_P \prod_{i=1}^{k} \|A_i P\|_F^2$

Algorithms

- One can solve **standard PCA** by Singular Value Decomposition (SVD)
- But SVD can't solve **Fair PCA** or **MCDR**.
- However, convex relaxation extends from **standard PCA** to **Fair PCA** and **MCDR**.

Convex Relaxation for Fair PCA

Input: data $A_1, \dots, A_k$ in n dimensions; target dimension $d \le n$

Output: projection matrix $X = PP^{\mathrm{T}}$

Algorithm: solve semi-definite program (SDP)

$$\max_{X \in \mathbb{R}^{n \times n}, z \in \mathbb{R}} z \text{ subject to}$$
$$z \le \langle A_1^{\mathrm{T}} A_1, X \rangle$$
$$\vdots$$
$$z \le \langle A_k^{\mathrm{T}} A_k, X \rangle$$
$$\mathrm{Tr}(X) \le d, 0 \preccurlyeq X \preccurlyeq I$$

Efficiently solvable

Output: $\hat{X} \in \mathbb{R}^{n \times n}$
$0 \preccurlyeq \hat{X} \preccurlyeq I, \mathrm{tr}(\hat{X}) \le d$

Problem: $\hat{X}$ has correct trace but higher rank than d

Output: extreme $\hat{X} \in \mathbb{R}^{n \times n}$
$0 \preccurlyeq \hat{X} \preccurlyeq I, \mathrm{rank}(\hat{X}) \le d + s$
(Rank violation $s = \sqrt{2k + 1/4} - \frac{3}{2}$)

Solution: solve SDP to an extreme point

Main Theoretical and Algorithmic Contributions

- Proof that *any* SDP extreme solution has low rank
- Algorithm to move to extreme point solutions
- Apply the result to **Fair PCA** and generalize to **MCDR**.

Our results apply to f_i's linear in PP^{T} and concave g, which cover well-studied welfare objectives including *Mar-Loss* and *NSW*.

Existence of low-rank property of SDP extreme point is known in optimization community (Barvinok'95, Pataki'98). We connect the result to ML application by generalizing the result to $X \preccurlyeq I$ constraint and developing the algorithms.

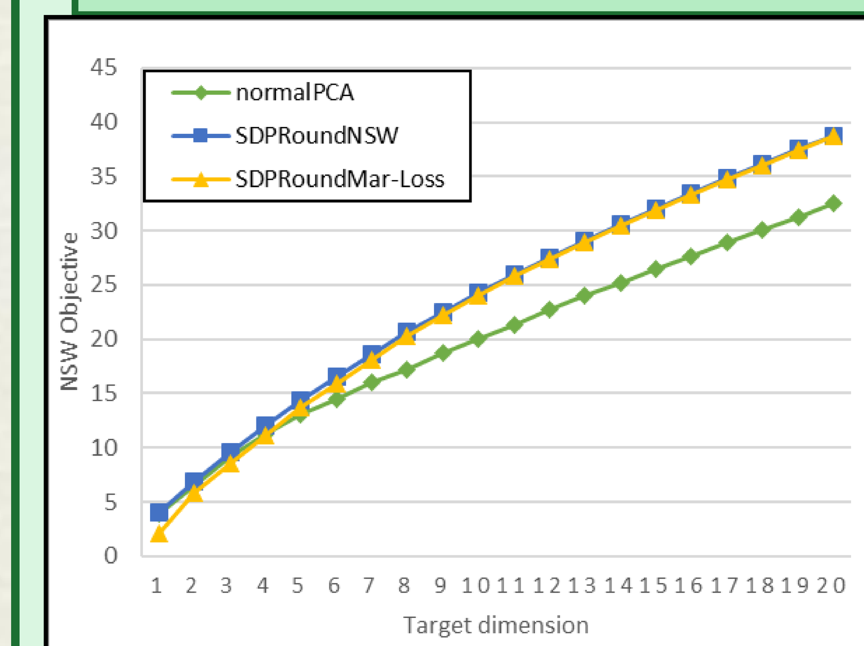
Other Theoretical Results

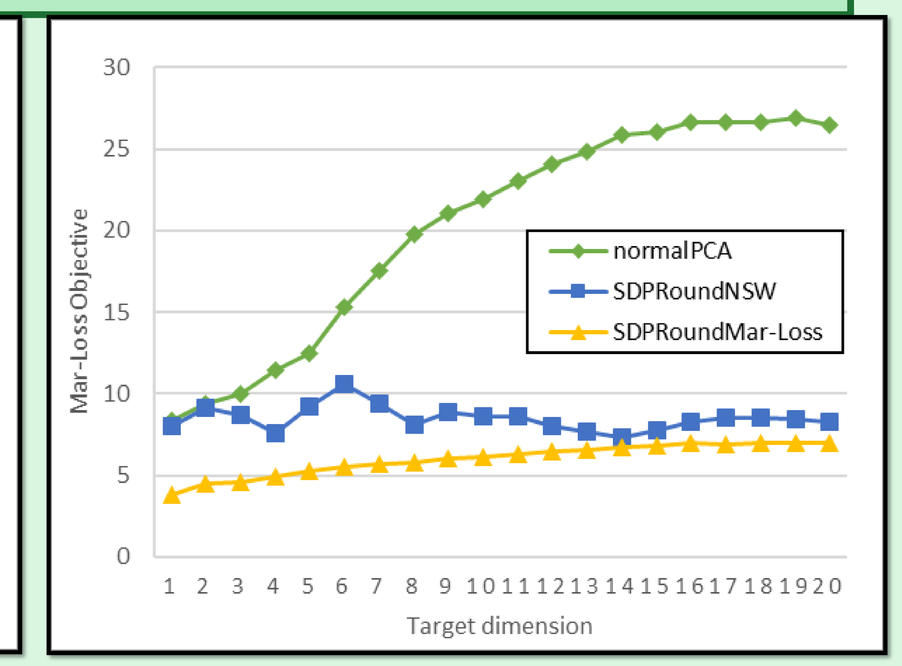
- Small rank violation leads to (multiplicative) approximation ratio performance guarantee
- Alternative iterative rounding for additive performance guarantee
- NP-hardness of **Fair PCA** (when k is general)

Experiments

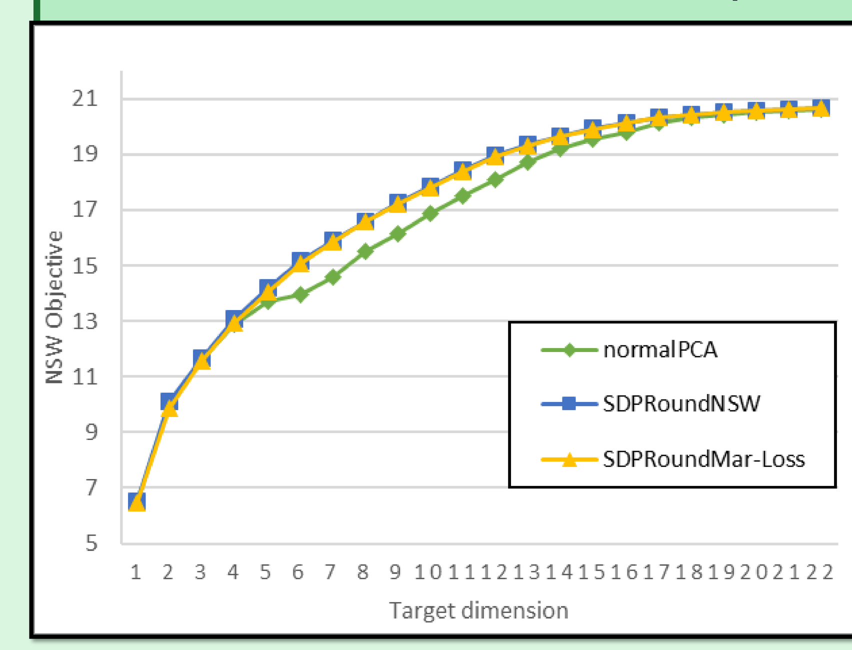
We run our algorithm **SDPRound** specified two objectives Nash Social Welfare (NSW) and Marginal Loss (Mar-Loss) and compare with standard PCA on two metrics NSW and Mar-Loss. The rank violation in practice is much smaller than the guarantee and usually non-existent.

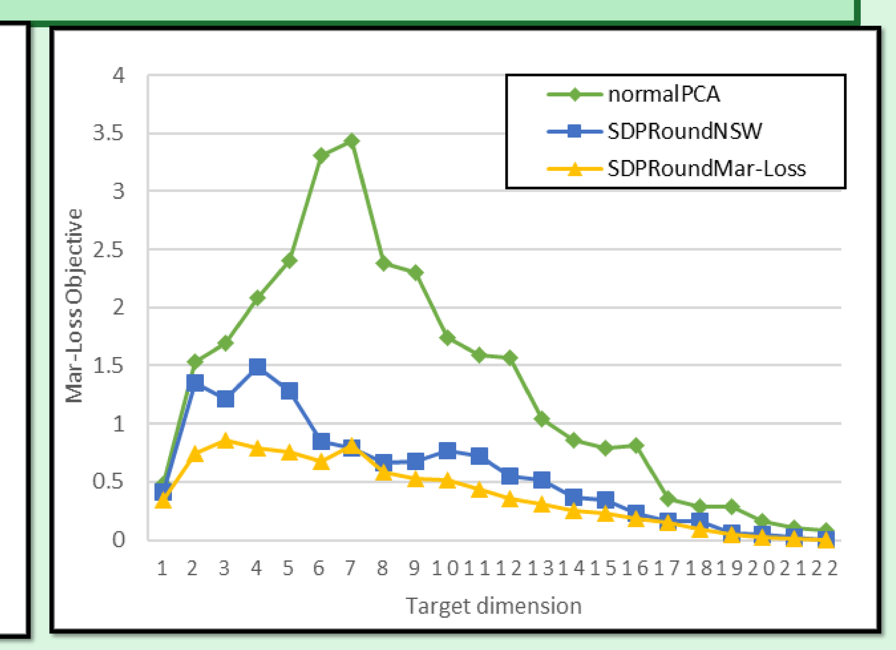
Adult Income data; 5 groups based on race
32k datapoints, 59 dimensions





Default Credit data; 6 groups based on education and gender
30k datapoints, 21 dimensions

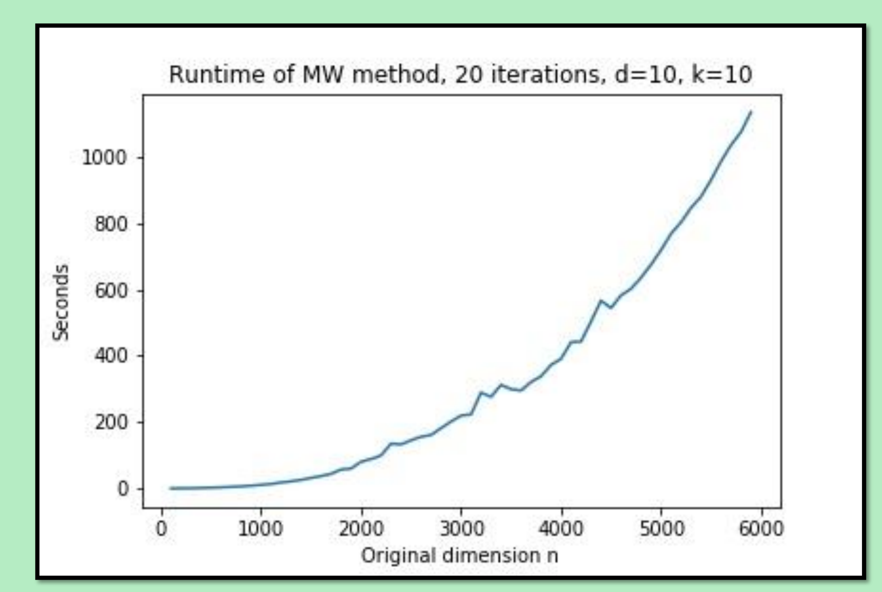




SDPRound performs better in both NSW and Mar-Loss objectives compared to standard PCA despite being optimized for only one objective.

Scalability

SDP solvers run in <1 minutes on data with original dimension $n \approx 50 - 70$. For big data and large n, we use multiplicative weight update (MW) for certain objectives of **MCDR**. In practice, MW runs on n ≈1000 dimensions in <20 minutes.



Code: github.com/SDPforAll/multiCriteriaDimReduction
Web: sites.google.com/site/ssamadi/fair-pca-homepage