# Two Decades of Low-Rank Optimization

**Sam Burer** **Renato D.C. Monteiro**

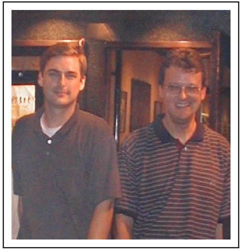
SIAM Conference on Optimization
Seattle, WA
June 2, 2023

# My Goals

- Describe an active area of research at the intersection of optimization and machine learning
- Provide a retrospective on our work from the early 2000's

# My Goals

- Describe an active area of research at the intersection of optimization and machine learning
- Provide a retrospective on our work from the early 2000's



# Outline

1. What is *low-rank*? Why do we care?
   - Examples and applications
   - Optimization paradigms
2. Our Contribution
   - The low-rank idea
   - Benign nonconvexity
3. Substantial Progress
   - Convex approaches
   - Low-rank approaches
   - Ongoing work in benign nonconvexity
4. Conclusions

# Outline

1. What is *low-rank*? Why do we care?
    - Examples and applications
    - Optimization paradigms
2. Our Contribution
    - The low-rank idea
    - Benign nonconvexity
3. Substantial Progress
    - Convex approaches
    - Low-rank approaches
    - Ongoing work in benign nonconvexity
4. Conclusions

# What is *low-rank*?

- $M \in \mathbb{R}^{m \times n}$
- *Low-rank* means $\mathrm{rank}(M) \ll \min\{m, n\}$
- A type of sparsity/interpretability
- Singular value decomposition (SVD)
  - $M = U\Sigma V^T = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^T$
- Spectral decomposition when $M = M^T$
  - $M = U\Lambda U^T = \widetilde{U}\widetilde{\Lambda}\widetilde{U}^T$

# Why do we care?

*Notions such as order, complexity, or dimensionality can often be expressed by means of the rank of an appropriate matrix.*

*—Recht, Fazel, Parrilo (2010)*

# Why do we care?

*Notions such as order, complexity, or dimensionality can often be expressed by means of the rank of an appropriate matrix.*

*For example, a low-rank matrix could correspond to a low-degree statistical model for a random process (e.g., factor analysis), a low-order realization of a linear system, a low-order controller for a plant, or a low-dimensional embedding of data in Euclidean space.*

*—Recht, Fazel, Parrilo (2010)*

# Why do we care?

SIAM J. MATH. DATA SCI.
Vol. 1, No. 1, pp. 144–160



## Why Are Big Data Matrices Approximately Low Rank?*

Madeleine Udell[†] and Alex Townsend[‡]

**Abstract.** Matrices of (approximate) low rank are pervasive in data science, appearing in movie preferences, text documents, survey data, medical records, and genomics. While there is a vast literature on how to exploit low rank structure in these datasets, there is less attention paid to explaining why the low rank structure appears in the first place. Here, we explain the effectiveness of low rank models in data science by considering a simple generative model for these matrices: we suppose that each row or column is associated to a (possibly high dimensional) bounded latent variable, and entries of the matrix are generated by applying a piecewise analytic function to these latent variables. These matrices are in general full rank. However, we show that we can approximate every entry of an $m \times n$ matrix drawn from this model to within a fixed absolute error by a low rank matrix whose rank grows as $\mathcal{O}(\log(m+n))$. Hence any sufficiently large matrix from such a latent variable model can be approximated, up to a small entrywise error, by a low rank matrix.

# Why do we care?

- Constraining the eigenvals of a symm var $X$ often yields matrices with low rank
  - E.g., the set of correlation matrices
- Generalizes the fact that optimal solutions of linear programs are generally sparse (for standard form $Ax = b, x \geq 0$)

# Why do we care?

RANK-REDUCIBILITY OF A SYMMETRIC MATRIX AND SAMPLING THEORY OF MINIMUM TRACE FACTOR ANALYSIS

ALEXANDER SHAPIRO

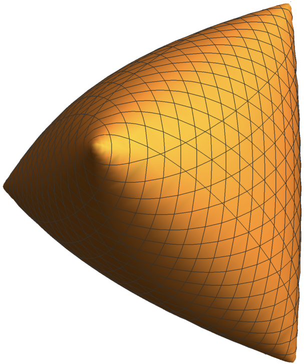
**Problems of Distance Geometry and Convex Properties of Quadratic Maps***

A. I. Barvinok

ON THE RANK OF EXTREME MATRICES IN SEMIDEFINITE PROGRAMS AND THE MULTIPLICITY OF OPTIMAL EIGENVALUES

GÁBOR PATAKI

# Why do we care?



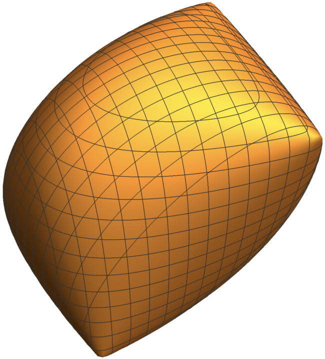


Image: Breiding, Kozhasov, Lerario (2019)

# What is low-rank optimization?

*Low-rank optimization* $=$ optimizing over $X \in \mathbb{R}^{m \times n}$ with $\operatorname{rank}(X) \le r$

1. Important for modeling low-rank data
2. Useful when low-rank structures appear naturally (irrespective of the data)

# Outline

# Example: Principal Component Analysis (PCA)

Given a data matrix $M \in \mathbb{R}^{m \times n}$, find the closest matrix with rank at most $r$:

$$\begin{aligned} \min \quad & \|M - X\|_F^2 \\ \text{s.t.} \quad & \operatorname{rank}(X) \le r \\ & X \in \mathbb{R}^{m \times n} \end{aligned}$$

# Example: Robust PCA

Given a data matrix $M \in \mathbb{R}^{m \times n}$, find matrices $X, S \in \mathbb{R}^{m \times n}$ such that:

- $M = X + S$
- $X$ is low-rank
- $S$ is sparse

# Example: Robust PCA

Given a data matrix $M \in \mathbb{R}^{m \times n}$, find matrices $X, S \in \mathbb{R}^{m \times n}$ such that:

- $M = X + S$
- $X$ is low-rank
- $S$ is sparse

Image: Bouwmans, Zahzah (2014)

# Example: Matrix Completion

Given a data matrix $M \in \mathbb{R}^{n \times m}$ for which only a subset of entries are known, find the min-rank matrix $X$, which matches the known entries:

$$
\begin{array}{ll}
\min & \operatorname{rank}(X) \\
\text{s.t.} & X_{ij} = M_{ij} \ \ \forall \text{ known entries } (i,j) \\
& X \in \mathbb{R}^{m \times n}
\end{array}
$$

# Example: Matrix Completion

Given a data matrix $M \in \mathbb{R}^{n \times m}$ for which only a subset of entries are known, find the min-rank matrix $X$, which matches the known entries:

$$\begin{array}{ll} \min & \operatorname{rank}(X) \\ \text{s.t.} & X_{ij} = M_{ij} \;\; \forall \text{ known entries } (i,j) \\ & X \in \mathbb{R}^{m \times n} \end{array}$$

- Recommender systems, e.g., Netflix, Spotify (Koren-Bell-Volinksy 2009)
- Wireless sensor localization (So-Ye 2007, Candés-Recht 2009)

# Example: Nonnegative Matrix Factorization

$$
\begin{aligned}
\min \quad & \|M - UV^T\|_F^2 \\
\text{s.t.} \quad & U \geq 0, V \geq 0 \\
& U \in \mathbb{R}^{n\times r}, V \in \mathbb{R}^{m\times r}
\end{aligned}
$$

# Example: Nonnegative Matrix Factorization

$$
\begin{aligned}
\min \quad & \|M - UV^T\|_F^2 \\
\text{s.t.} \quad & U \geq 0, V \geq 0 \\
& U \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{m \times r}
\end{aligned}
$$

- Here, the rank is restricted explicitly
- Applications in data clustering and feature extraction in imaging and text analysis (Lee-Seung 1999, Hofmann 1999, Vavasis 2009)

# Example: Quadratically Constrained Quadratic Programming

$$
\begin{aligned}
\min \quad & \langle x, Fx \rangle + 2\langle f, x \rangle \\
\text{s.t.} \quad & \langle x, G_i x \rangle + 2\langle g_i, x \rangle + \gamma_i \le 0 \quad \forall\ i \\
& x \in \mathbb{R}^n
\end{aligned}
$$

# Example: Quadratically Constrained Quadratic Programming

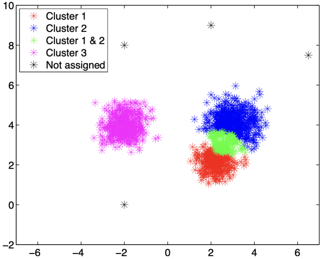
$$\begin{aligned} \min \quad & \langle x, Fx \rangle + 2\langle f, x \rangle \\ \text{s.t.} \quad & \langle x, G_i x \rangle + 2\langle g_i, x \rangle + \gamma_i \le 0 \quad \forall\ i \\ & x \in \mathbb{R}^n \end{aligned}$$

$$\downarrow$$

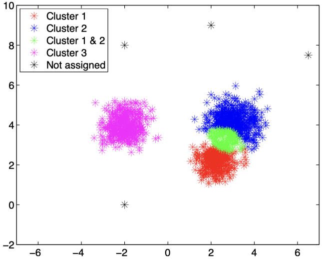
$$\begin{aligned} \min \quad & \langle F, X \rangle + 2\langle f, x \rangle \\ \text{s.t.} \quad & \langle G_i, X \rangle + 2\langle g_i, x \rangle + \gamma_i \le 0 \quad \forall\ i \\ & X = xx^T \end{aligned}$$

# Example: Clustering with Significant Overlap and Outliers

- "Clustering problem when the data points. . . may contain both outliers and large regions of overlap" (Hou et al 2015)
- $m$ data points, $k$ clusters $\Longrightarrow$ QCQP over assignment matrices of size $m \times k$




(c) Ground-truth clusters




(d) LRSDP initialization

Image: Hou et al (2015)

# Outline

# Optimization Paradigms: Linear vs Nonlinear

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & a_i^T x \le b_i \ \ \forall\, i \\ & x \in \mathbb{R}^n \end{aligned}$$

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \le 0 \quad \forall\, i \\ & x \in \mathbb{R}^n \end{aligned}$$

# Optimization Paradigms: Convexity vs Nonconvexity

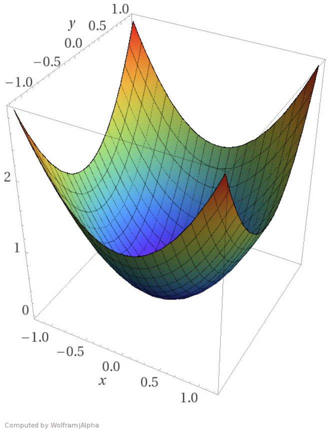
SIAM Review
Vol. 35, No. 2, pp. 183–238, June 1993


**LAGRANGE MULTIPLIERS AND OPTIMALITY***

R. TYRRELL ROCKAFELLAR†

a convex set every locally optimal solution is global. Also, first-order necessary conditions for optimality turn out to be sufficient. A variety of other properties conducive to computation and interpretation of solutions ride on convexity as well. In fact the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity. Even for problems that aren't themselves of convex type, convexity may enter, for instance, in setting up subproblems as part of an iterative numerical scheme.

# Optimization Paradigms: Convexity vs Nonconvexity



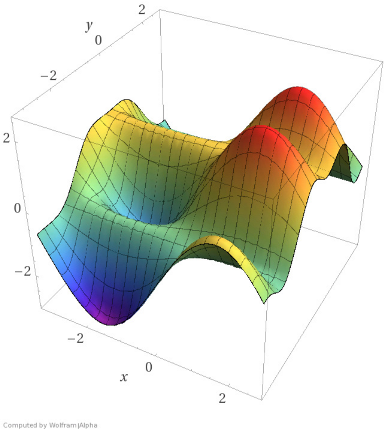




Image: Zadeh (2016)

# Low-Rank Optimization

- Unfortunately, the *rank* function is nonconvex
- So low-rank optimization is nonconvex
- In fact, it is NP-hard
- But still an important problem to solve

# Outline

1. What is *low-rank*? Why do we care?
   - Examples and applications
   - Optimization paradigms
2. Our Contribution
   - The low-rank idea
   - Benign nonconvexity
3. Substantial Progress
   - Convex approaches
   - Low-rank approaches
   - Ongoing work in benign nonconvexity
4. Conclusions

# A Brief History of Semidefinite Programming (SDP)

$$
\begin{aligned}
\min \quad & \langle F, X \rangle + 2\langle f, x \rangle \\
\text{s.t.} \quad & \langle G_i, X \rangle + 2\langle g_i, x \rangle + \gamma_i \le 0 \quad \forall\, i \\
& X \succeq xx^T
\end{aligned}
$$

# A Brief History of Semidefinite Programming (SDP)

$$\begin{array}{ll} \min & \langle F, X \rangle + 2\langle f, x \rangle \\ \text{s.t.} & \langle G_i, X \rangle + 2\langle g_i, x \rangle + \gamma_i \le 0 \quad \forall\, i \\ & X \succeq xx^T \end{array}$$

$\downarrow$ (standard form)

$$\begin{array}{ll} \min & \langle C, X \rangle \\ \text{s.t.} & \langle A_i, X \rangle = b_i \;\; \forall\, i = 1, \ldots, m \\ & X \succeq 0 \end{array}$$

# A Brief History of Semidefinite Programming (SDP)

- In the 1990's, research and interest exploded

# A Brief History of Semidefinite Programming (SDP)

- In the 1990's, research and interest exploded
  - Polynomial-time interior-point methods (delivering max-rank opt solns)
  - Goemans-Williamson approximation algorithm for MaxCut using SDP (1995)

# A Brief History of Semidefinite Programming (SDP)

- In the 1990's, research and interest exploded
  - Polynomial-time interior-point methods (delivering max-rank opt solns)
  - Goemans-Williamson approximation algorithm for MaxCut using SDP (1995)
- At the turn of the millenium, researchers were solving bigger SDPs more efficiently, often using first-order methods

# A Brief History of Semidefinite Programming (SDP)

- In the 1990's, research and interest exploded
  - Polynomial-time interior-point methods (delivering max-rank opt solns)
  - Goemans-Williamson approximation algorithm for MaxCut using SDP (1995)
- At the turn of the millenium, researchers were solving bigger SDPs more efficiently, often using first-order methods
  - Dual-scaling method (Benson-Ye-Zhang 2000)
  - Spectral bundle method (Helmberg-Rendl 2000)
  - Bundle method (Sotirov-Rendl 2001)
  - Nonlinear-programming approaches (Vanderbei-Benson 2003)
  - Dual Cholesky approach (B-Monteiro-Zhang 2002)
  - Chordal-graph approaches (Fukuda et al 2001)
  - Iterative solver for Newton system (Nakata-Fujisawa-Kojima 1998, Toh 2004)

# The Factorization Approach

- The MaxCut SDP was derived by replacing $UU^T$ with $X \succeq 0$; $U$ was square
- Peinado and Homer (1997) proposed to optimize directly with $U$ (instead of $X$)
- B and Monteiro (2001) saved time and space by taking $U$ lower triangular

# The Factorization Approach

- The MaxCut SDP was derived by replacing $UU^T$ with $X \succeq 0$; $U$ was square
- Peinado and Homer (1997) proposed to optimize directly with $U$ (instead of $X$)
- B and Monteiro (2001) saved time and space by taking $U$ lower triangular

- These papers converted the convex SDP into a nonconvex problem
- It worked in practice, but why?
- One argument: 100% equivalent to the SDP because $U$ was full rank

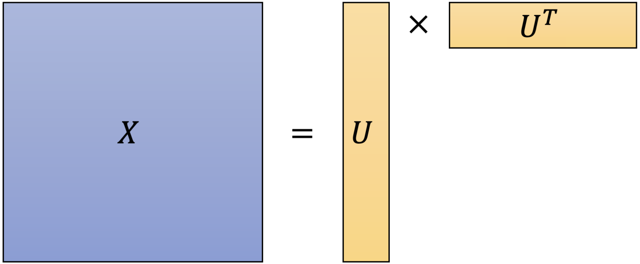
# The Low-Rank Factorization Approach

$$
\begin{aligned}
\min \quad & \langle C, X \rangle \\
\text{s.t.} \quad & \langle A_i, X \rangle = b_i \;\; \forall\, i = 1, \ldots, m \\
& X \succeq 0
\end{aligned}
$$

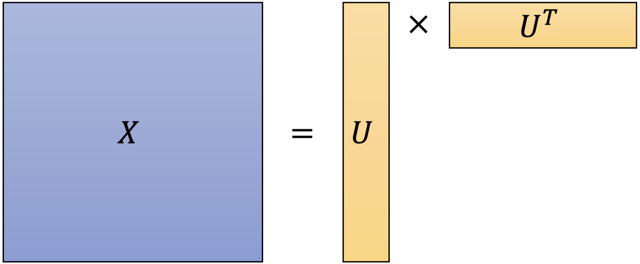
**Theorem (Shapiro 1982, Barvinok 1995, Pataki 1998)**

*There exists an optimal solution $X^*$ with rank $r^* < \sqrt{2m}$.*

# The Low-Rank Factorization Approach



# The Low-Rank Factorization Approach




- We proposed to replace $X$ by $UU^T$ with $U$ having $\lceil \sqrt{2m} \rceil$ columns
- I.e., with rank just high enough to maintain equivalence with the SDP

# The Low-Rank Factorization Approach

- Still equivalent to the SDP...but with a very different feasible set
- The potential advantage was less computation and storage
- But would it actually work in practice?

# The Low-Rank Factorization Approach

- Still equivalent to the SDP...but with a very different feasible set
- The potential advantage was less computation and storage
- But would it actually work in practice?

Can't hurt to try 🤷‍♂️

# The Advisor-Advisee Relationship

```
email — sburer@BIZ-BAIS-030:~/Desktop/email — alpine -f ./Monteiro.mbox — 78×25
 ALPINE 2.25   MESSAGE TEXT ./Monteiro.mbox  Message 1,371 of 2,501 ALL

Date: Mon, 22 May 2000 23:04:25 -0400
From: Samuel Burer <burer@math.gatech.edu>
To: Renato D.C. Monteiro <monteiro@isye.gatech.edu>
Subject: Results for low rank approach

Dr. Monteiro,

I did a quick implemenation of the low-rank idea for the primal
maxcut that we were talking about, and it looks promising. The
convergence is good, and of course, the computation and memory
requirements are less.

One example: On G1, the new approach cuts the times for both the
function and gradient evaluations by approximaately 1/2.

Sam

                                  [ALL of message]
? Help         < MsgIndex    P PrevMsg      - PrevPage D Delete     R Reply
O OTHER CMDS   > ViewAttch   N NextMsg    Spc NextPage U Undelete   F Forward
```

# The Advisor-Advisee Relationship

```
email — sburer@BIZ-BAIS-030:~/Desktop/email — alpine -f ./Monteiro.mbox — 78×25
  ALPINE 2.25   MESSAGE TEXT ./Monteiro.mbox  Message 1,371 of 2,501 ALL

Date: Mon, 22 May 2000 2[illegible]:04:25 -0400
From: Samuel Burer <burer@math.gatech.edu>
To: Renato D.C. Monteiro <monteiro@isye.gatech.edu>
Subject: Results for low rank approach

Dr. Monteiro,

I did a quick implemenation of the low-rank idea for the primal
maxcut that we were talking about, and it looks promising. The
convergence is good, and of course, the computation and memory
requirements are less.

One example: On G1, the new approach cuts the times for both the
function and gradient evaluations by approximaately 1/2.

Sam

                                  [ALL of message]
? Help         < MsgIndex   P PrevMsg      - PrevPage D Delete     R Reply
O OTHER CMDS   > ViewAttch  N NextMsg    Spc NextPage U Undelete   F Forward
```

# The Advisor-Advisee Relationship

```
email — sburer@BIZ-BAIS-030:~/Desktop/email — alpine -f ./Monteiro.mbox — 78×25

  ALPINE 2.25   MESSAGE TEXT ./Monteiro.mbox  Message 1,372 of 2,501 ALL

Date: Mon, 22 May 2000 23:54:49 -0400 (EDT)
From: monteiro@isye.gatech.edu
To: burer@math.gatech.edu
Cc: monteiro@isye.gatech.edu
Subject: Re: Results for low rank approach

Sam:

This is really good news. Let us hope that this approach works
well on all problems. If so, we may have some chance to compete
with the bundle method on some maxcut problems.

-- R.M.

                                              [ALL of message]
? Help        < MsgIndex   P PrevMsg      - PrevPage D Delete      R Reply
O OTHER CMDS  > ViewAttch  N NextMsg    Spc NextPage U Undelete    F Forward
```

# The Advisor-Advisee Relationship

email — sburer@BIZ-BAIS-030:~/Desktop/email — alpine -f ./Monteiro.mbox — 78×25

```
 ALPINE 2.25   MESSAGE TEXT ./Monteiro.mbox  Message 1,372 of 2,501 ALL

Date: Mon, 22 May 2000 23:54:49 -0400 (EDT)
From: monteiro@isye.gatech.edu
To: burer@math.gatech.edu
Cc: monteiro@isye.gatech.edu
Subject: Re: Results for low rank approach

Sam:

This is really good news. Let us hope that this approach works
well on all problems. If so, we may have some chance to compete
with the bundle method on some maxcut problems.

-- R.M.

                                    [ALL of message]
? Help       < MsgIndex  P PrevMsg     - PrevPage D Delete   R Reply
O OTHER CMDS > ViewAttch N NextMsg   Spc NextPage U Undelete F Forward
```

# Computational Results

- Had success on specific classes of large-scale SDPs (e.g., relaxations of MaxCut, maximum stable set, and the quadratic assignment problem)
- Developed a code for general SDPs
- Used only first-order methods (i.e., function and gradient information only)
- Also experimented with:
  - Restricting the rank smaller than required by theory
  - Dynamically updating the rank
  - Heuristically solving NP-hard problems using rank-2 approaches (with Y. Zhang)

# Outline

# Summarizing the Low-Rank Idea

Given an optimization problem with matrix variable $X$ satisfying $\mathrm{rank}(X) \leq r$

- If $X \succeq 0$, then replace $X \to UU^T$ with $U$ having $r$ columns
- If $X$ is a general matrix, then replace $X \to UV^T$ with $U, V$ each having $r$ columns

# Summarizing the Low-Rank Idea

- Why should this work at all?

# Summarizing the Low-Rank Idea

- Why should this work at all?

- In the early 2000's, this seemed challenging to answer

- One observation is the following:
  - If we apply the same orthogonal rotation to all rows of $U$ and $V$, then the product $UV^T$ does not change
  - That is, $UV^T = UQQ^TV^T = (UQ)(VQ)^T$ for all orthogonal matrices $Q$

# Some Pictures

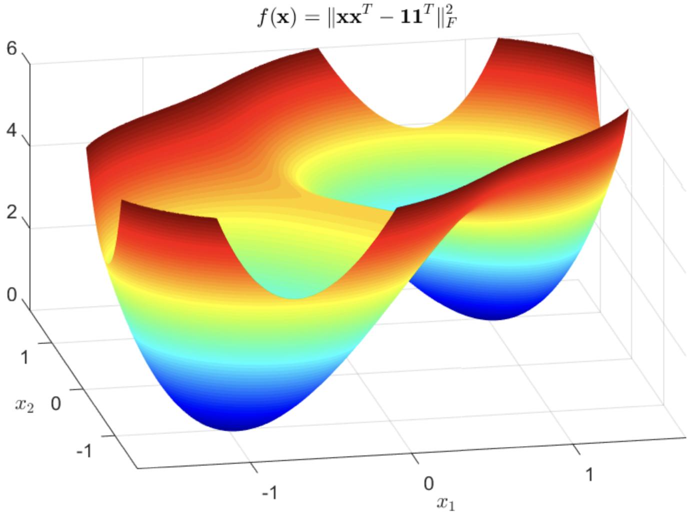



Image: Chen, Chi (2019)

# Some Pictures

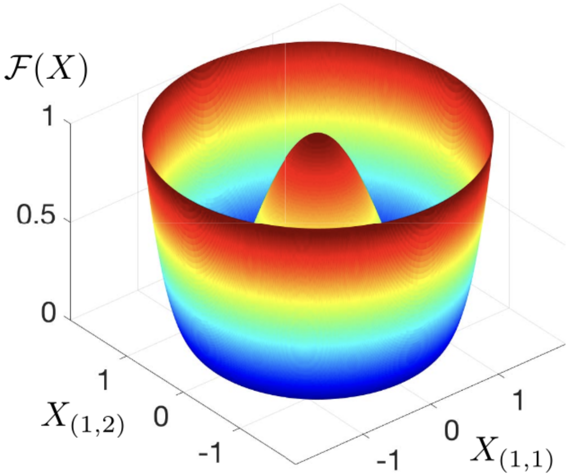



Image: Li et al (2019)

# Some Pictures

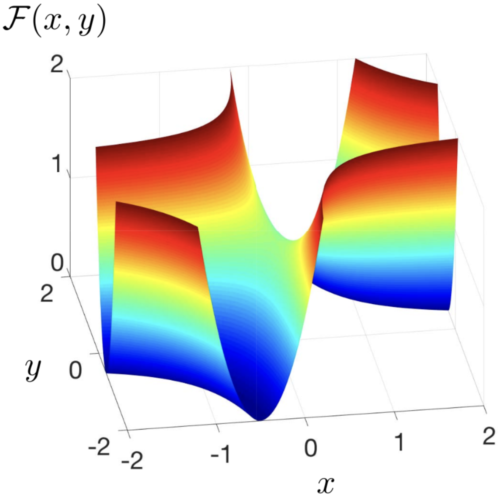



Image: Li et al (2019)

# Some Pictures

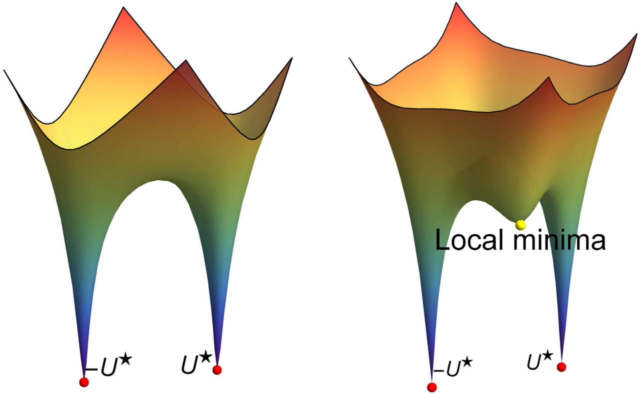



Image: Zhu et al (2021)

# A Theoretical Result

Suppose the original SDP feasible set is compact with interior.

# A Theoretical Result

Suppose the original SDP feasible set is compact with interior. Consider the rank-constrained SDP, enforcing $\text{rank}(X) \le r$ with $r \ge \lceil \sqrt{2(m+1)} \rceil$.

# A Theoretical Result

Suppose the original SDP feasible set is compact with interior. Consider the rank-constrained SDP, enforcing $\operatorname{rank}(X) \le r$ with $r \ge \lceil \sqrt{2(m+1)} \rceil$.

- Let $\bar{X}$ be a local minimum of the rank-constrained SDP. If $\operatorname{rank}(\bar{X}) < r$, then $\bar{X}$ is optimal for the original SDP.

# A Theoretical Result

Suppose the original SDP feasible set is compact with interior. Consider the rank-constrained SDP, enforcing $\text{rank}(X) \le r$ with $r \ge \lceil \sqrt{2(m+1)} \rceil$.

- Let $\bar{X}$ be a local minimum of the rank-constrained SDP. If $\text{rank}(\bar{X}) < r$, then $\bar{X}$ is optimal for the original SDP.
- Solve the factorized problem in $U$ via the standard augmented Lagrangian algorithm

# A Theoretical Result

Suppose the original SDP feasible set is compact with interior. Consider the rank-constrained SDP, enforcing rank$(X) \le r$ with $r \ge \lceil \sqrt{2(m+1)} \rceil$.

- Let $\bar{X}$ be a local minimum of the rank-constrained SDP. If rank$(\bar{X}) < r$, then $\bar{X}$ is optimal for the original SDP.

- Solve the factorized problem in $U$ via the standard augmented Lagrangian algorithm — but with added penalty term $\epsilon \det(U^T U)$.

# A Theoretical Result

Suppose the original SDP feasible set is compact with interior. Consider the rank-constrained SDP, enforcing rank$(X) \le r$ with $r \ge \lceil \sqrt{2(m+1)} \rceil$.

- Let $\bar{X}$ be a local minimum of the rank-constrained SDP. If rank$(\bar{X}) < r$, then $\bar{X}$ is optimal for the original SDP.

- Solve the factorized problem in $U$ via the standard augmented Lagrangian algorithm — but with added penalty term $\epsilon \det(U^T U)$. If an unconstrained local minimum is attained at each step,

# A Theoretical Result

Suppose the original SDP feasible set is compact with interior. Consider the rank-constrained SDP, enforcing $\text{rank}(X) \leq r$ with $r \geq \lceil \sqrt{2(m+1)} \rceil$.

- Let $\bar{X}$ be a local minimum of the rank-constrained SDP. If $\text{rank}(\bar{X}) < r$, then $\bar{X}$ is optimal for the original SDP.

- Solve the factorized problem in $U$ via the standard augmented Lagrangian algorithm — but with added penalty term $\epsilon \det(U^T U)$. If an unconstrained local minimum is attained at each step, then the overall algorithm converges to a global optimal solution $U^*$.

# Outline

1. What is *low-rank*? Why do we care?
   - Examples and applications
   - Optimization paradigms
2. Our Contribution
   - The low-rank idea
   - Benign nonconvexity
3. Substantial Progress
   - Convex approaches
   - Low-rank approaches
   - Ongoing work in benign nonconvexity
4. Conclusions

# Benign Nonconvexity

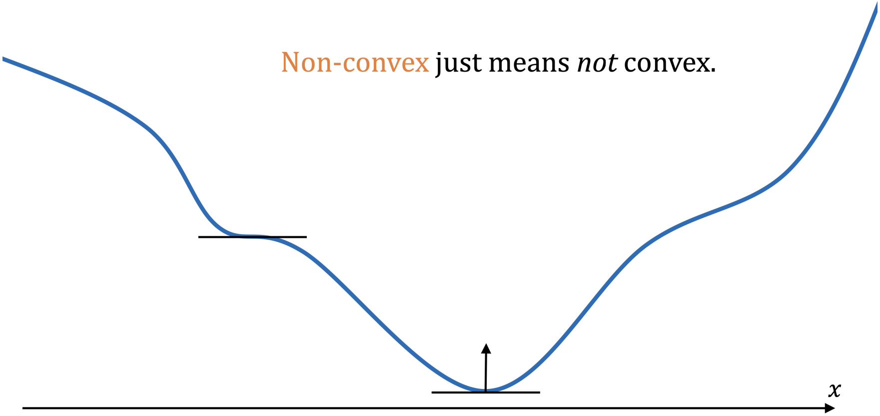



Image: Boumal (2021)

# Benign Nonconvexity

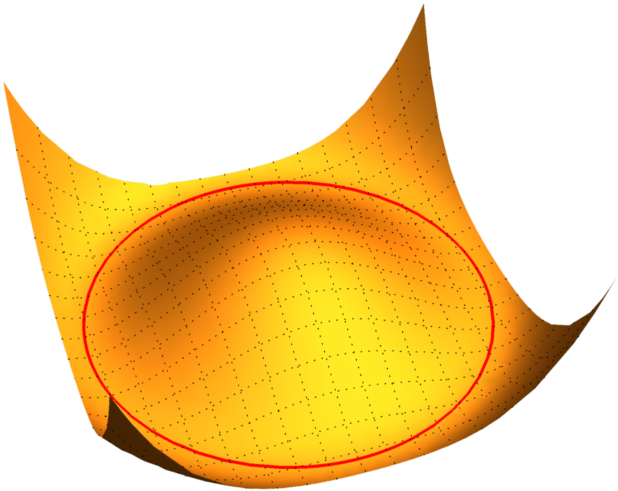


Image: Yuan (2022)

# Benign Nonconvexity


HT: Yuan (2022)

# Benign Nonconvexity


HT: Yuan (2022)

# Outline

# Outline

# Convex Approaches

$\boxed{\text{Nuclear Norm}}$

- The *nuclear norm* $\|X\|_*$ is the sum of the singular values of $X$
- $\|X\|_*$ is the convex envelope of $\operatorname{rank}(X)$ when $\|X\|_* \leq 1$ (Fazel-Hindi-Boyd 2001)
- Guarantees for calculating the min-rank solution using the nuclear norm (Candès-Recht 2009, Recht-Fazel-Parrilo 2010, Wright et al 2009, Chandrasekaran et al 2011)

# Convex Approaches

- Sparse PCA using SDP (d'Aspremont et al 2004)
- Sketching algorithms for solving SDPs directly, but with low-rank storage (Yurtsever et al 2017)
- Links between low-rank matrix completion and matrix sparsity (Madani et al 2017)

# Outline

# Low-Rank Approaches

- Manifold optimization to solve SDPs; escaping saddle points (Journée et al 2010)
- Optimization with orthogonal matrices; finding the closest low-rank correlation matrix (Wen-Yin 2013)
- Practically solving huge matrix completion problems via stochastic gradient with theoretical guarantees (Recht-Ré 2013, DeSa-Olukotun-Ré 2015)

# Low-Rank Approaches

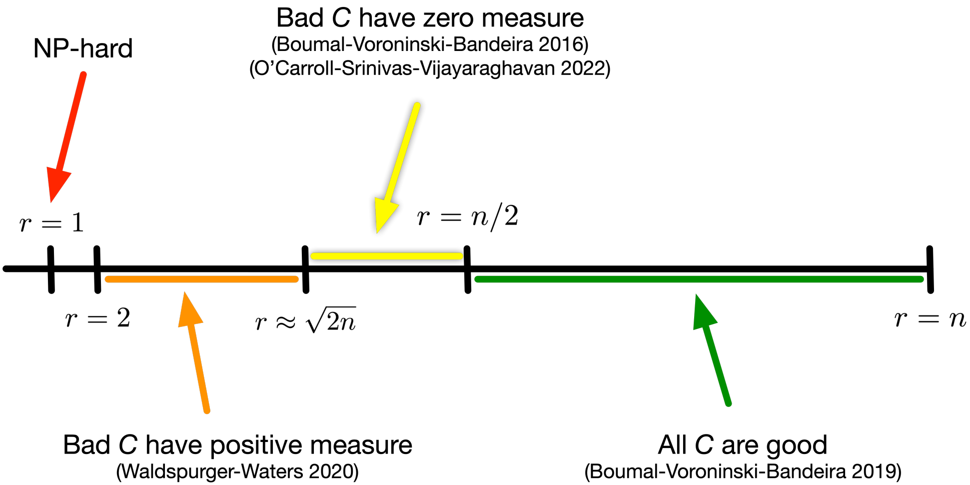
- Convergence rates for convex functions of $X$; clever intialization to achieve global optimality (Bhojanapalli-Kyrillidis-Sanghavi 2016)
- No spurious local minima in low-rank matrix completion and recovery (Ge-Lee-Ma 2016, Bhojanapalli-Neyshabur-Srebro 2016, Ge-Jin-Zheng 2017)
- Convergence rates for block-coordinate descent on the MaxCut SDP (Erdogdu et al 2018)

# Low-Rank Approaches

$\boxed{\text{Back to Standard-Form SDP...}}$

# Low-Rank Approaches

$\boxed{\text{Back to Standard-Form SDP}\ldots\text{via the MaxCut SDP with } m = n}$




HT: Boumal (2023)

# Low-Rank Approaches

$\boxed{\text{Back to Standard-Form SDP}}$

- When $r > \sqrt{2m}$, in the setting of smoothed analysis, the low-rank approach solves SDPs to any desired accuracy in polynomial time (Cifuentes-Moitra 2021)

# Outline

1. What is *low-rank*? Why do we care?
   - Examples and applications
   - Optimization paradigms
2. Our Contribution
   - The low-rank idea
   - Benign nonconvexity
3. Substantial Progress
   - Convex approaches
   - Low-rank approaches
   - Ongoing work in benign nonconvexity
4. Conclusions

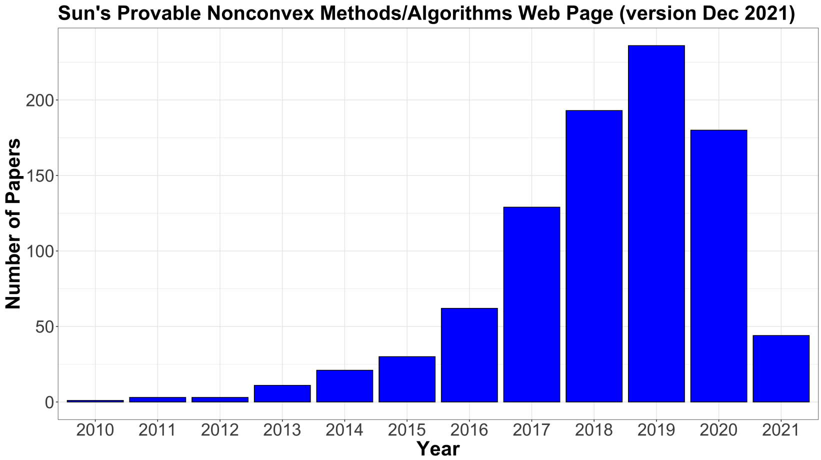
# Ongoing Work in Benign Nonconvexity

Ju Sun's Provable Nonconvex Methods/Algorithms Web Page (version Dec 2021)



Source: Sun (2021)

# Ongoing Work in Benign Nonconvexity




Source: Sun (2021)

# Outline

# Goals Redux

- The low-rank idea has turned out to be a simple, powerful, and intriguing technique. . . with many interesting applications in optimization and machine learning

# Goals Redux

- The low-rank idea has turned out to be a simple, powerful, and intriguing technique. . . with many interesting applications in optimization and machine learning

- Benign nonconvexity is an important area with lots to explore

# Goals Redux

- The low-rank idea has turned out to be a simple, powerful, and intriguing technique. . . with many interesting applications in optimization and machine learning

- Benign nonconvexity is an important area with lots to explore

- And a Big Thanks for indulging this trip down memory lane!