Rank optimality for the Burer-Monteiro factorization

Irène Waldspurger

CNRS and CEREMADE (Université Paris Dauphine)
Équipe MOKAPLAN (INRIA)

Joint work with Alden Waters (Bernoulli Institute, Rijksuniversiteit Groningen)

June 19, 2020
Séminaire signal-apprentissage
Marseille, I2M / LIS

Semidefinite programming

$$
\begin{aligned}
\text{minimize } & \mathrm{Trace}(CX) \\
\text{such that } & \mathcal{A}(X) = b, \\
& X \succeq 0.
\end{aligned}
$$

Here,

- X, the unknown, is an $n \times n$ matrix ;
- C is a fixed $n \times n$ matrix (cost matrix) ;
- $\mathcal{A} : \mathrm{Sym}_n \to \mathbb{R}^m$ is linear ;
- b is a fixed vector in $\mathbb{R}^m$.

Motivations

Very diverse applications.

Main motivation for us : Many hard combinatorial optimization problems can be approximated by semidefinite programs.

Motivations

Very diverse applications.

Main motivation for us : Many hard combinatorial optimization problems can be approximated by semidefinite programs.

Principle :

$$\text{Quadratic constraints over a vector } x \in \mathbb{R}^n$$
$$\Updownarrow$$
$$\text{Linear constraints over the matrix } X = xx^T \in \mathbb{R}^{n\times n}$$

$\Rightarrow$ Problems with quadratic constraints can be turned to semidefinite programs with the change of variable "$x \to X$".

Numerical solvers

General SDPs can be solved at arbitrary precision in polynomial time.
But the order of the polynomial is large.

Numerical solvers

General SDPs can be solved at arbitrary precision in polynomial time.
But the order of the polynomial is large.

Interior point solvers : complexity $O(n^4)$ per iteration.

First-order ones : $O(n^3)$, more iterations needed.

...

Numerical solvers

General SDPs can be solved at arbitrary precision in polynomial time.
But the order of the polynomial is large.

Interior point solvers : complexity $O(n^4)$ per iteration.

First-order ones : $O(n^3)$, more iterations needed.

...

$\rightarrow$ Numerically, high dimensional SDPs are difficult to solve.

Low-rank semidefinite programming

We can speed up the solving if we know that the solution has some special structure and exploit it.

Low-rank semidefinite programming

We can speed up the solving if we know that the solution has some special structure and exploit it.

$\rightarrow$ We assume that the solution has low rank $r \ll n$.

Low-rank semidefinite programming

We can speed up the solving if we know that the solution has some special structure and exploit it.

$\rightarrow$ We assume that the solution has low rank $r \ll n$.

Intuition : When the problem has been obtained by the change of variable "$x \rightarrow X = xx^T$", the solution should be rank 1.

Exploiting the low rank

Two main strategies :

- Frank-Wolfe methods ;
  [Frank and Wolfe, 1956]
- Burer-Monteiro factorization.
  [Burer and Monteiro, 2003]

Remark : The Burer-Monteiro factorization is only a heuristic, which may not always work. The goal of the present work is precisely to help understanding when it works / does not work.

Burer-Monteiro factorization : principle

- Assume that there is a solution with rank r_{opt}.
- Choose some integer $p \geq r_{opt}$.
- Write X under the form

$$X = VV^T,$$

with V an $n \times p$ matrix.
- Minimize $\mathrm{Trace}(CVV^T)$ over V.

$$\text{minimize } \mathrm{Trace}(CX)$$
$$\text{for } X \in \mathbb{R}^{n\times n} \text{ such that } \mathcal{A}(X) = b, \quad X \succeq 0.$$

$$\Updownarrow$$

$$\text{minimize } \mathrm{Trace}(CVV^T)$$
$$\text{for } V \in \mathbb{R}^{n\times p} \text{ such that } \mathcal{A}(VV^T) = b.$$

Remark : p is the *factorization rank*. It must be chosen, and can be equal to or larger than r_{opt}.

$$\text{minimize } \mathrm{Trace}(CVV^T)$$
$$\text{for } V \in \mathbb{R}^{n \times p} \text{ such that } \mathcal{A}(VV^T) = b.$$

We assume that $\{V \in \mathbb{R}^{n \times p}, \mathcal{A}(VV^T) = b\}$ is a "nice" manifold.

$\to$ Solve with Riemannian optimization algorithms.

Main advantage of the factorized formulation

The number of variables is not $O(n^2)$ anymore, but $O(np)$, with $p \ll n$.

$\rightarrow$ Allows reducing the computational complexity.

Main advantage of the factorized formulation

The number of variables is not $O(n^2)$ anymore, but $O(np)$, with $p \ll n$.

$\rightarrow$ Allows reducing the computational complexity.

Main drawback of the factorized formulation

Contrarily to the SDP, the factorized problem is non-convex.

$\rightarrow$ Riemannian algorithms may get stuck at a critical point instead of finding a global minimizer.

Main advantage of the factorized formulation

The number of variables is not $O(n^2)$ anymore, but $O(np)$, with $p \ll n$.

$\rightarrow$ Allows reducing the computational complexity.

Main drawback of the factorized formulation

Contrarily to the SDP, the factorized problem is non-convex.

$\rightarrow$ Riemannian algorithms may get stuck at a critical point instead of finding a global minimizer.

This issue can arise or not, depending on the factorization rank p.
$\Rightarrow$ How to choose p ?

Outline

1. Literature review
   - In practice, algorithms don't get stuck if $p \gtrsim r_{opt}$.
   - In particular situations, this phenomenon is understood.
   - In a general setting, no guarantees unless $p \gtrsim \sqrt{2m}$.
   - But $r_{opt} \ll \sqrt{2m}$. Why this gap ?

Outline

1. Literature review
   - In practice, algorithms don't get stuck if $p \gtrsim r_{opt}$.
   - In particular situations, this phenomenon is understood.
   - In a general setting, no guarantees unless $p \gtrsim \sqrt{2m}$.
   - But $r_{opt} \ll \sqrt{2m}$. Why this gap ?
2. Optimal rank for the Burer-Monteiro formulation
   - A minor improvement is possible over previous general guarantees.
   - The improved result is optimal.

     → If $p \lesssim \sqrt{2m}$, Riemannian algorithms cannot be certified correct without additional assumptions.
   - Idea of proof.

Outline

1. Literature review
   - In practice, algorithms don't get stuck if $p \gtrsim r_{opt}$.
   - In particular situations, this phenomenon is understood.
   - In a general setting, no guarantees unless $p \gtrsim \sqrt{2m}$.
   - But $r_{opt} \ll \sqrt{2m}$. Why this gap ?
2. Optimal rank for the Burer-Monteiro formulation
   - A minor improvement is possible over previous general guarantees.
   - The improved result is optimal.

     $\rightarrow$ If $p \lesssim \sqrt{2m}$, Riemannian algorithms cannot be certified correct without additional assumptions.
   - Idea of proof.
3. Open questions

Numerical observations

1. [Burer and Monteiro, 2003]
   Various problems, notably MaxCut and minimum bisection.
2. [Journée, Bach, Absil, and Sepulchre, 2010]
   MaxCut (with a particular initialization scheme).
3. [Boumal, 2015]
   Orthogonal synchronization.
4. "SDP-like" problems ; see for example [Mishra, Meyer, Bonnabel, and Sepulchre, 2014].

In all these articles, it is reported that Riemannian algorithms do not get stuck as soon as p is slightly larger than r_0.

Theoretical explanations in particular cases

Strong guarantees, but in very specific situations only.

Typical result :

> "Consider a specific subclass of semidefinite programs.
> In it, choose an element at random, with a specific probability distribution.
> With high probability, Riemannian algorithms do not get stuck if $p \geq r_0$."

Main such result : [Bandeira, Boumal, and Voroninski, 2016]

Other particular SDP-like problems : [Ge, Lee, and Ma, 2016]

...

General case : one main result [Boumal, Voroninski, and Bandeira, 2018]

$$\text{minimize } \mathrm{Trace}(CVV^T)$$
$$\text{for } V \in \mathbb{R}^{n\times p} \text{ such that } \mathcal{A}(VV^T) = b.$$

The only assumption is (approximately) that

$$\mathcal{M}_p \stackrel{\text{déf}}{=} \{V \in \mathbb{R}^{n\times p}, \mathcal{A}(VV^T) = b\}$$

is a manifold.

General case : one main result [Boumal, Voroninski, and Bandeira, 2018]

$$\text{minimize } \mathrm{Trace}(CVV^T), \quad \text{for } V \in \mathcal{M}_p.$$

Riemannian optimization algorithms typically converge to second-order critical points :

A matrix $V_0 \in \mathcal{M}_p$ is a second-order critical point if

- $\nabla f_C(V_0) = 0_{n,p}$;
- $\mathrm{Hess}\, f_C(V_0) \succeq 0$,

where $f_C \overset{déf}{=} (V \in \mathcal{M}_p \to \mathrm{Trace}(CVV^T))$.

General case : one main result [Boumal, Voroninski, and Bandeira, 2018]

Theorem

For almost all matrices C, if

$$p > \left\lfloor \sqrt{2m + \frac{1}{4}} - \frac{1}{2} \right\rfloor,$$

all second-order critical points are global minimizers.
Consequently, Riemannian optimization algorithms always find a global minimizer.

General case : one main result [Boumal, Voroninski, and Bandeira, 2018]

Theorem

For almost all matrices C, if

$$p > \left\lfloor \sqrt{2m + \frac{1}{4}} - \frac{1}{2} \right\rfloor,$$

all second-order critical points are global minimizers.
Consequently, Riemannian optimization algorithms always find a global minimizer.

Remark : The value of p does not depend on r_{opt}.

Summary

- In empirical experiments, as well as in the few particular cases that have been studied, algorithms seem to always work when
$$p = O(r_{opt}).$$
- The only available general result guarantees that algorithms work when
$$p \gtrsim \sqrt{2m}.$$

Summary

- In empirical experiments, as well as in the few particular cases that have been studied, algorithms seem to always work when

$$p = O(r_{opt}).$$

- The only available general result guarantees that algorithms work when

$$p \gtrsim \sqrt{2m}.$$

As r_{opt} is often much smaller than $\sqrt{2m}$, this leaves a big gap.

$\rightarrow$ Is it possible to obtain general guarantees for $p \ll \sqrt{2m}$?

Overview of our results

- A minor improvement is possible over the result by [Boumal, Voroninski, and Bandeira, 2018], but it does not change the leading order term

$$p \gtrsim \sqrt{2m}.$$

Overview of our results

- A minor improvement is possible over the result by [Boumal, Voroninski, and Bandeira, 2018], but it does not change the leading order term

$$p \gtrsim \sqrt{2m}.$$

- With this improvement, the result is essentially optimal, even if $r_{opt} \ll \sqrt{2m}$.

[Boumal, Voroninski, and Bandeira, 2018]

Theorem

For almost all matrices C, if

$$p > \left\lfloor \sqrt{2m + \frac{1}{4}} - \frac{1}{2} \right\rfloor,$$

all second-order critical points are global minimizers.
Consequently, Riemannian optimization algorithms always find a global minimizer.

[Boumal, Voroninski, and Bandeira, 2018] improved

Theorem

For almost all matrices C, if

$$p > \left\lfloor \sqrt{2m + \frac{1}{4}} - \frac{1}{2} \right\rfloor,$$

$\left\lfloor \sqrt{2m + \frac{9}{4}} - \frac{3}{2} \right\rfloor$

all second-order critical points are global minimizers.
Consequently, Riemannian optimization algorithms always find a global minimizer.

Theorem (Quasi-optimality of the previous result)

Let $r_0 = \min\{\mathrm{rank}(X), \mathcal{A}(X) = b, X \succeq 0\}$.
Under suitable hypotheses, if

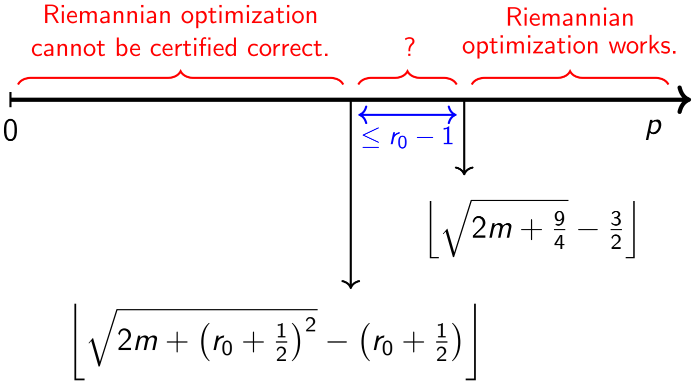
$$p \leq \left\lfloor \sqrt{2m + \left(r_0 + \frac{1}{2}\right)^2} - \left(r_0 + \frac{1}{2}\right) \right\rfloor,$$

there is a set of matrices C with non-zero Lebesgue measure for which :

1. The global minimizer has rank r_0.
2. There is a second order critical point which is not a global minimizer.

Comments

- In most applications, r_0 is small, possibly $r_0 = 1$.
- We have the following picture :



Example : MaxCut relaxations

Relaxes the *Maximum Cut* problem from graph theory. [Delorme and Poljak, 1993]

Maximum Cut : for a graph with weighted edges, split the graph in two so as to maximize the weight of cut edges.

Example : MaxCut relaxations

Relaxes the *Maximum Cut* problem from graph theory. [Delorme and Poljak, 1993]

Maximum Cut : for a graph with weighted edges, split the graph in two so as to maximize the weight of cut edges.

Most famous example of a SDP approximating a hard combinatorial problem.

Example : MaxCut relaxations

Relaxes the *Maximum Cut* problem from graph theory. [Delorme and Poljak, 1993]

Maximum Cut : for a graph with weighted edges, split the graph in two so as to maximize the weight of cut edges.

Most famous example of a SDP approximating a hard combinatorial problem.

$$\begin{aligned} \text{minimize } & \mathrm{Trace}(CX) \\ \text{such that } & \mathrm{diag}(X) = 1, \\ & X \succeq 0. \end{aligned}$$

Example : MaxCut relaxations

$$\begin{aligned} &\text{minimize } \mathrm{Trace}(CX), \\ &\text{such that } \mathrm{diag}(X) = 1, \\ &X \succeq 0. \end{aligned}$$

(Original SDP)

$$\Downarrow$$

$$\begin{aligned} &\text{minimize } \mathrm{Trace}(CVV^T), \\ &\text{such that } \mathrm{diag}(VV^T) = 1, V \in \mathbb{R}^{n\times p}. \end{aligned}$$

(Burer-Monteiro factorization)

In this case, $r_0 = 1$.

Example : MaxCut relaxations

- For almost all C, if

$$p > \left\lfloor \sqrt{2n + \frac{9}{4}} - \frac{3}{2} \right\rfloor ,$$

  no bad second-order critical point exists ; Riemannian algorithms work.

- If

$$p \leq \left\lfloor \sqrt{2n + \frac{9}{4}} - \frac{3}{2} \right\rfloor ,$$

  even when assuming a rank 1 solution, there are matrices C for which Riemannian algorithms can fail.

Idea of proof

We assume $p \lesssim \sqrt{2m}$. $\mathcal{A}$ and b are fixed.

We want to show that there exists C for which the problem

$$\begin{aligned} &\text{minimize } \mathrm{Trace}(CX) \\ &\text{for } X \in \mathbb{R}^{n\times n} \text{ such that } \mathcal{A}(X) = b, \\ &\qquad\qquad\qquad\qquad X \succeq 0. \end{aligned}$$

1. has a global minimizer of rank r_0;
2. has a “bad” second order critical point when factorized through the Burer-Monteiro heuristic.

Idea of proof

The theorem actually requires a whole non-zero Lebesgue measure set of such matrices C to exist.

With classical geometrical arguments, it more or less suffices to construct one such matrix C.

Idea of proof

The theorem actually requires a whole non-zero Lebesgue measure set of such matrices C to exist.

With classical geometrical arguments, it more or less suffices to construct one such matrix C.

$\Rightarrow$ Let us construct C.

Idea of proof : construct C

1. This problem must have a rank r_0 minimizer :

$$\begin{aligned} &\text{minimize } \mathrm{Trace}(CX) \\ &\text{for } X \in \mathbb{R}^{n\times n} \text{ such that } \mathcal{A}(X) = b, \\ &\qquad X \succeq 0. \end{aligned}$$

2. This one must have a "bad" second-order critical point :

$$\begin{aligned} &\text{minimize } \mathrm{Trace}(CVV^T) \\ &\text{for } V \in \mathbb{R}^{n\times p} \text{ such that } \mathcal{A}(VV^T) = b. \end{aligned}$$

These conditions can be written under a purely analytical form.

Idea of proof : construct C

After several simplifications, we see that the existence of C is implied by the existence of X_0, V, μ such that :

Idea of proof : construct C

After several simplifications, we see that the existence of C is implied by the existence of X_0, V, μ such that :

- X_0 is feasible for the SDP and has rank r_0 ;
- V is feasible for the factorized problem ;

Idea of proof : construct C

After several simplifications, we see that the existence of C is implied by the existence of X_0, V, μ such that :

- X_0 is feasible for the SDP and has rank r_0 ;
- V is feasible for the factorized problem ;
- $V^T \mathcal{A}^*(\mu) V \succeq 0$ and $X_0^T \mathcal{A}^*(\mu) V = 0$.

Idea of proof : construct C

The first two conditions are easy ; we focus on the third one.

$$\exists \mu, \quad V^T \mathcal{A}^*(\mu) V \succeq 0 \qquad \text{and} \qquad X_0^T \mathcal{A}^*(\mu) V = 0\,?$$

Idea of proof : construct C

The first two conditions are easy ; we focus on the third one.

$$\exists \mu, \quad V^T \mathcal{A}^*(\mu) V \succeq 0 \quad \text{and} \quad X_0^T \mathcal{A}^*(\mu) V = 0\,?$$

Fix X_0, V. Consider the map

$$\begin{array}{ccccc} \overbrace{\mathbb{R}^m}^{\text{dimension } m} & \to & \overbrace{\mathrm{Sym}^{p\times p} \quad \times \quad \mathbb{R}^{r_0 \times p}}^{\text{dimension } \frac{p(p+1)}{2} + pr_0} \\ \mu & \to & (V^T \mathcal{A}^*(\mu) V, \quad X_0^T \mathcal{A}^*(\mu) V) \end{array}$$

Idea of proof : construct C

The first two conditions are easy ; we focus on the third one.

$$\exists \mu, \quad V^T \mathcal{A}^*(\mu) V \succeq 0 \quad \text{and} \quad X_0^T \mathcal{A}^*(\mu) V = 0\,?$$

Fix X_0, V. Consider the map

$$\begin{array}{ccccc} \overbrace{\mathbb{R}^m}^{\text{dimension } m} & \to & \overbrace{\mathrm{Sym}^{p\times p} \quad \times \quad \mathbb{R}^{r_0\times p}}^{\text{dimension } \frac{p(p+1)}{2} + pr_0} \\ \mu & \to & (V^T \mathcal{A}^*(\mu) V, \quad X_0^T \mathcal{A}^*(\mu) V) \end{array}$$

If $m \geq \frac{p(p+1)}{2} + pr_0$, it is generically surjective and μ exists.

Idea of proof : construct C

The first two conditions are easy ; we focus on the third one.

$$\exists \mu, \quad V^T \mathcal{A}^*(\mu) V \succeq 0 \quad \text{and} \quad X_0^T \mathcal{A}^*(\mu) V = 0\,?$$

Fix X_0, V. Consider the map

$$\begin{array}{ccccc} \overbrace{\mathbb{R}^m}^{\text{dimension } m} & \rightarrow & \overbrace{\mathrm{Sym}^{p\times p} \quad \times \quad \mathbb{R}^{r_0 \times p}}^{\text{dimension } \frac{p(p+1)}{2} + pr_0} \\ \mu & \rightarrow & (V^T \mathcal{A}^*(\mu) V, \quad X_0^T \mathcal{A}^*(\mu) V) \end{array}$$

If $m \geq \frac{p(p+1)}{2} + pr_0$, it is generically surjective and μ exists.

$$\iff p \leq \sqrt{2m + \left(r_0 + \tfrac{1}{2}\right)^2} - \left(r_0 + \tfrac{1}{2}\right)$$

Burer-Monteiro factorization : summary

- [Boumal, Voroninski, and Bandeira, 2018]

  When $p \gtrsim \sqrt{2m}$, for almost any cost matrix, all second-order critical points are minimizers.

  Numerical experiments suggest it could be true for

  $$p = O(r_{opt}) \ll \sqrt{2m}.$$

- [Our result]

  When $p \lesssim \sqrt{2m}$, it is not true.

Open questions

1. Refined understanding of the regime $p < \sqrt{2m}$;
2. Study of specific semidefinite programs for applications.

Understanding the regime $p < \sqrt{2m}$

Our result says :

> "If we want to guarantee that the Burer-Monteiro heuristic works for almost all cost matrices C, we need $p \gtrsim \sqrt{2m}$."

But in practice, we oftentimes do not need the heuristic to work for *all* cost matrices.

Understanding the regime $p < \sqrt{2m}$

Our result says :

"If we want to guarantee that the Burer-Monteiro heuristic works for almost all cost matrices C, we need $p \gtrsim \sqrt{2m}$."

But in practice, we oftentimes do not need the heuristic to work for *all* cost matrices.

$\rightarrow$ Establish more realistic guarantees, like

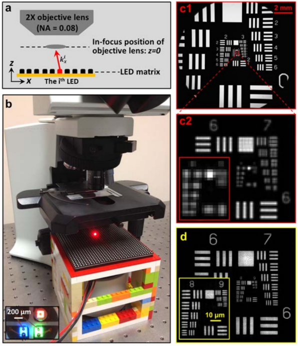
"For $p \in [r_0; \sqrt{2m}]$, the Burer-Monteiro heuristic works for most cost matrices C." ?

Application to phase retrieval problems

Reconstruct $x \in \mathbb{C}^d$ from $|\langle a_k, x\rangle|, 1 \leq k \leq m$.

Here,

- $a_1, \dots, a_m \in \mathbb{C}^d$ are known ;
- $|.|$ is the complex modulus.



Important applications in optics.

Algorithms using approximations with semidefinite programs usually offer good reconstruction quality, but are too slow.

Application to phase retrieval problems

To what extent does the Burer-Monteiro heuristic allow to speed up these algorithms ?

Thank you !

I. Waldspurger and A. Waters (2018). Rank optimality for the Burer-Monteiro factorization. arXiv preprint arXiv :1812.03046.