Completely Positive Factorization in Orthogonality Optimization via Smoothing Method

Zhijian Lai; Akiko Yoshise

University of Tsukuba, Japan

July 22, 2021
SIAM Conference on Optimization

CP Factorization via Orthogonality Constrained Problem

Background: Basic Concepts

Definition 1.1 [Abraham and Naomi, 2003]

1. A matrix $A \in \mathbb{S}_n$ is called completely positive (CP matrix) if there exists an *entrywise nonnegative* matrix $B \in \mathbb{R}^{n \times r}$ such that $A = BB^\top$. Such B is called a CP factorization of A.

Background: Basic Concepts

Definition 1.1 [Abraham and Naomi, 2003]

1. A matrix $A \in \mathbb{S}_n$ is called completely positive (CP matrix) if there exists an *entrywise nonnegative* matrix $B \in \mathbb{R}^{n\times r}$ such that $A = BB^\top$. Such B is called a CP factorization of A.
2. $\mathcal{CP}_n := \left\{BB^\top \in \mathbb{S}_n \mid B \text{ is a nonnegative matrix}\right\}$ denotes the completely positive cone (CP cone).

Background: Basic Concepts

Definition 1.1 [Abraham and Naomi, 2003]

1. A matrix $A \in \mathbb{S}_n$ is called completely positive (CP matrix) if there exists an *entrywise nonnegative* matrix $B \in \mathbb{R}^{n \times r}$ such that $A = BB^\top$. Such B is called a CP factorization of A.
2. $\mathcal{CP}_n := \left\{ BB^\top \in \mathbb{S}_n \mid B \text{ is a nonnegative matrix} \right\}$ denotes the completely positive cone (CP cone).

For example, consider the matrix $A \in \mathcal{CP}_3$ where $A = B_1 B_1^\top = B_2 B_2^\top = B_3 B_3^\top$.

$$A = \begin{pmatrix} 18 & 9 & 9 \\ 9 & 18 & 9 \\ 9 & 9 & 18 \end{pmatrix}.$$

$$B_1 = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 3 & 3 & 0 \\ 3 & 0 & 3 \\ 0 & 3 & 3 \end{pmatrix}, \quad B_3 = \begin{pmatrix} 3 & 3 & 0 & 0 \\ 3 & 0 & 3 & 0 \\ 3 & 0 & 0 & 3 \end{pmatrix}.$$

Background: Basic Concepts

Definition 1.1 [Abraham and Naomi, 2003]

1. A matrix $A \in \mathbb{S}_n$ is called completely positive (CP matrix) if there exists an *entrywise nonnegative* matrix $B \in \mathbb{R}^{n \times r}$ such that $A = BB^\top$. Such B is called a CP factorization of A.
2. $\mathcal{CP}_n := \left\{ BB^\top \in \mathbb{S}_n \mid B \text{ is a nonnegative matrix} \right\}$ denotes the completely positive cone (CP cone).

For example, consider the matrix $A \in \mathcal{CP}_3$ where $A = B_1 B_1^\top = B_2 B_2^\top = B_3 B_3^\top$.

$$A = \begin{pmatrix} 18 & 9 & 9 \\ 9 & 18 & 9 \\ 9 & 9 & 18 \end{pmatrix}.$$

$$B_1 = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 3 & 3 & 0 \\ 3 & 0 & 3 \\ 0 & 3 & 3 \end{pmatrix}, \quad B_3 = \begin{pmatrix} 3 & 3 & 0 & 0 \\ 3 & 0 & 3 & 0 \\ 3 & 0 & 0 & 3 \end{pmatrix}.$$

Definition 1.2 [Abraham and Naomi, 2003]

The *minimum of the number of columns* among CP factorizations of $A \in \mathcal{CP}_n$ is called cp-rank of A, written as $\text{cp}(A)$.

Background: Application of $\mathcal{CP}_n$

Many nonconvex NP-hard quadratic and combinatorial optimizations have a linear program over completely positive cone, $\mathcal{CP}_n$.

Background: Application of $\mathcal{CP}_n$

Many nonconvex NP-hard quadratic and combinatorial optimizations have a linear program over completely positive cone, $\mathcal{CP}_n$.

Example (Standard quadratic optimization via $\mathcal{CP}_n$ [Bomze et al., 2000])

For example, standard quadratic optimization:

$$\min \left\{ x^\top M x \mid e^\top x = 1, x \in \mathbb{R}^n_+ \right\},$$

can equivalently be written as

$$\min \left\{ \langle M, X \rangle \mid \left\langle ee^\top, X \right\rangle = 1, X \in \mathcal{CP}_n \right\},$$

where $M \in \mathbb{S}_n$ possibly indefinite, and e is the all ones vector.

Background: Application of $\mathcal{CP}_n$

Many nonconvex NP-hard quadratic and combinatorial optimizations have a linear program over completely positive cone, $\mathcal{CP}_n$.

Example (Standard quadratic optimization via $\mathcal{CP}_n$ [Bomze et al., 2000])

For example, standard quadratic optimization:

$$\min \left\{ x^\top M x \mid e^\top x = 1, x \in \mathbb{R}^n_+ \right\},$$

can equivalently be written as

$$\min \left\{ \langle M, X \rangle \mid \left\langle ee^\top, X \right\rangle = 1, {\color{red} X \in \mathcal{CP}_n} \right\},$$

where $M \in \mathbb{S}_n$ possibly indefinite, and e is the all ones vector.

An application of above is

Example (Independence number α of a graph G [De Klerk and Pasechnik, 2002])

$$\alpha(G) = \max \left\{ \langle ee^\top, X \rangle \mid \langle A + I, X \rangle = 1, {\color{red} X \in \mathcal{CP}_n} \right\},$$

where A is the adjacency matrix of G.

Background: Open Problems — Finding a CP factorization

There are *many* fundamental open problems in completely positive cone, $\mathcal{CP}_n$:

Background: Open Problems — Finding a CP factorization

There are *many* fundamental open problems in completely positive cone, $\mathcal{CP}_n$:

A list of open problems about $\mathcal{CP}_n$ [Berman et al., 2015]

1. Checking membership in $\mathcal{CP}_n$.
2. Determining geometry of $\mathcal{CP}_n$.
3. Finding a factorization of a matrix in $\mathcal{CP}_n$, i.e., the "CP factorization problem" (Our goal):

$$\text{Find } B \in \mathbb{R}^{n\times r} \text{ s.t. } A = BB^\top \text{ and } B \geq 0. \tag{CPfact}$$

4. Computing the cp-rank.
5. Finding cutting planes for completely positive optimization problems.

CP Factorization via Orthogonality Constrained Problem

Reformulation of CP factorization problem: Lemma 2.1

Lemma 2.1

Suppose that $A \in \mathbb{S}_n$, $r \in \mathbb{N}$. Then

$$r \geq \mathrm{cp}(A) \Longleftrightarrow A \text{ has a CP factorization } B \text{ with } r \text{ columns.}$$

Reformulation of CP factorization problem: Lemma 2.1

Lemma 2.1

Suppose that $A \in \mathbb{S}_n$, $r \in \mathbb{N}$. Then

$$r \geq \mathrm{cp}(A) \Longleftrightarrow A \text{ has a CP factorization } B \text{ with } r \text{ columns.}$$

Proof.

If we have had a CP-factorization B with r columns, then we can easily get another CP-factorization with r' columns for every positive integer $r' \geq r$. $\square$

Reformulation of CP factorization problem: Lemma 2.1

Lemma 2.1

Suppose that $A \in \mathbb{S}_n$, $r \in \mathbb{N}$. Then

$$r \geq \mathrm{cp}(A) \iff A \text{ has a CP factorization } B \text{ with } r \text{ columns.}$$

Proof.

If we have had a CP-factorization B with r columns, then we can easily get another CP-factorization with r' columns for every positive integer $r' \geq r$. □

For example, consider

$$A = \begin{pmatrix} 18 & 9 & 9 \\ 9 & 18 & 9 \\ 9 & 9 & 18 \end{pmatrix} \in \mathcal{CP}_3.$$

Given $A = B_1 B_1^\top$, we can easily construct B_2 such that $B_2 B_2^\top = A$.

$$B_1 = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix} \geq 0, \quad \longrightarrow \quad B_2 := \begin{pmatrix} 4 & 1 & 1 & 0 \\ 1 & 4 & 1 & 0 \\ 1 & 1 & 4 & 0 \end{pmatrix} \geq 0, \text{ or } \begin{pmatrix} 4 & 1 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 1 & 1 & 4 & 0 & 0 \end{pmatrix} \geq 0.$$

Reformulation of CP factorization problem: Lemma 2.2

Lemma 2.2 [Xu, 2004, Lemma 1.]

Let $\mathcal{O}_r \triangleq \{X \in \mathbb{R}^{r\times r} \mid X^\top X = I\}$. Suppose that $B, C \in \mathbb{R}^{n\times r}$. Then

$$BB^\top = CC^\top \iff \exists X \in \mathcal{O}_r \text{ such that } BX = C.$$

Reformulation of CP factorization problem: Lemma 2.2

Lemma 2.2 [Xu, 2004, Lemma 1.]

Let $\mathcal{O}_r \triangleq \{X \in \mathbb{R}^{r\times r} \mid X^\top X = I\}$. Suppose that $B, C \in \mathbb{R}^{n\times r}$. Then

$$BB^\top = CC^\top \iff \exists X \in \mathcal{O}_r \text{ such that } BX = C.$$

For example, we have known that $A = B_1 B_1^\top = B_2 B_2^\top$.

$$A = \begin{pmatrix} 18 & 9 & 9 \\ 9 & 18 & 9 \\ 9 & 9 & 18 \end{pmatrix}, \text{ and } B_1 = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix}, \quad B_2 = \begin{pmatrix} 3 & 3 & 0 \\ 3 & 0 & 3 \\ 0 & 3 & 3 \end{pmatrix}.$$

In fact, there is an orthogonal matrix X such that $B_1 X = B_2$.

$$X = \frac{1}{3}\begin{pmatrix} 2 & 2 & -1 \\ 2 & -1 & 2 \\ -1 & 2 & 2 \end{pmatrix} \in \mathcal{O}_3.$$

Reformulation of CP factorization problem

From a "bad" factorization $B \not\geq 0$. Using spectral decomposition $A = VDV^\top$, we define $B := V\sqrt{D}$, then $A = BB^T$.

$$A = \begin{pmatrix} 18 & 9 & 9 \\ 9 & 18 & 9 \\ 9 & 9 & 18 \end{pmatrix}, \quad B = \begin{pmatrix} \frac{3}{\sqrt{2}} & \frac{\sqrt{6}}{2} & 2\sqrt{3} \\ -\frac{3}{\sqrt{2}} & \frac{\sqrt{6}}{2} & 2\sqrt{3} \\ 0 & -\sqrt{6} & 2\sqrt{3} \end{pmatrix} \not\geq 0.$$

To a "good" factorization $BX \geq 0$. We find a suitable orthogonal matrix X, e.g.,

$$X = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{6}}{6} & \frac{\sqrt{6}}{6} & -\frac{\sqrt{6}}{3} \\ \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix} \in \mathcal{O}_3, \quad BX = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix} \geq 0.$$

Reformulation of CP factorization problem

From a "bad" factorization $B \not\geq 0$. Using spectral decomposition $A = VDV^\top$, we define $B := V\sqrt{D}$, then $A = BB^T$.

$$A = \begin{pmatrix} 18 & 9 & 9 \\ 9 & 18 & 9 \\ 9 & 9 & 18 \end{pmatrix}, \quad B = \begin{pmatrix} \frac{3}{\sqrt{2}} & \frac{\sqrt{6}}{2} & 2\sqrt{3} \\ -\frac{3}{\sqrt{2}} & \frac{\sqrt{6}}{2} & 2\sqrt{3} \\ 0 & -\sqrt{6} & 2\sqrt{3} \end{pmatrix} \not\geq 0.$$

To a "good" factorization $BX \geq 0$. We find a suitable orthogonal matrix X, e.g.,

$$X = \begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{6}}{6} & \frac{\sqrt{6}}{6} & -\frac{\sqrt{6}}{3} \\ \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix} \in \mathcal{O}_3, \quad BX = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix} \geq 0.$$

Reformulation of CP factorization problem[Groetzner and Dür, 2020]

$$\text{Find } X \text{ s.t. } BX \geq 0 \text{ and } X \in \mathcal{O}_r, \tag{FeasCP}$$

where $r \geq \text{cp}(A)$, $B \in \mathbb{R}^{n\times r}$ is an *arbitrary* initial factorization $A = BB^\top$ (need not nonnegative).

Reformulation of CP factorization problem

Reformulation of CP factorization problem [Groetzner and Dür, 2020]

$$\text{Find } X \text{ s.t. } BX \geq 0 \text{ and } X \in \mathcal{O}_r, \qquad \text{(FeasCP)}$$

where $r \geq \text{cp}(A)$, $B \in \mathbb{R}^{n \times r}$ is an arbitrary initial factorization $A = BB^\top$ (need not nonnegative).

Reformulation of CP factorization problem

Reformulation of CP factorization problem [Groetzner and Dür, 2020]

$$\text{Find } X \text{ s.t. } BX \geq 0 \text{ and } X \in \mathcal{O}_r, \qquad \text{(FeasCP)}$$

where $r \geq \text{cp}(A)$, $B \in \mathbb{R}^{n \times r}$ is an arbitrary initial factorization $A = BB^\top$ (need not nonnegative).

Approaches to feasibility problem (FeasCP)

[Groetzner and Dür, 2020] By $\mathcal{P} := \{X | BX \geq 0\}$, they applied the *alternating projections method* to

$$\text{Find } X \text{ s.t. } X \in \mathcal{P} \cap \mathcal{O}_r.$$

$$\text{Proj}_{\mathcal{P}}(X) \rightarrow \text{second-order cone problem; } \text{Proj}_{\mathcal{O}_r}(X) \rightarrow \text{singular value decomposition.}$$

Reformulation of CP factorization problem

Reformulation of CP factorization problem [Groetzner and Dür, 2020]

$$\text{Find } X \text{ s.t. } BX \geq 0 \text{ and } X \in \mathcal{O}_r, \tag{FeasCP}$$

where $r \geq \text{cp}(A)$, $B \in \mathbb{R}^{n \times r}$ is an arbitrary initial factorization $A = BB^\top$ (need not nonnegative).

Approaches to feasibility problem (FeasCP)

[Groetzner and Dür, 2020] By $\mathcal{P} := \{X | BX \geq 0\}$, they applied the *alternating projections method* to

$$\text{Find } X \text{ s.t. } X \in \mathcal{P} \cap \mathcal{O}_r.$$

$\text{Proj}_{\mathcal{P}}(X) \to$ second-order cone problem; $\text{Proj}_{\mathcal{O}_r}(X) \to$ singular value decomposition.

[Chen et al., 2020] Suppose that $\text{Proj}_C(x), \text{Proj}_D(x)$ can be computed efficiently for two closed sets C and D. A *difference-of-convex functions approach* for solving the *split feasibility problem*,

$$\text{Find } x \text{ s.t. } Ax \in D \text{ and } x \in C, \tag{SFP}$$

can be directly applied for (FeasCP) if $C = \mathcal{O}_r$, $D = \mathbb{R}^{n \times r}_{\geq 0}$ and $\text{Proj}_D(X) = \max\{X, 0\}$.

Reformulation of CP factorization problem

Reformulation of CP factorization problem [Groetzner and Dür, 2020]

$$\text{Find } X \text{ s.t. } BX \geq 0 \text{ and } X \in \mathcal{O}_r, \qquad \text{(FeasCP)}$$

where $r \geq \text{cp}(A)$, $B \in \mathbb{R}^{n \times r}$ is an arbitrary initial factorization $A = BB^\top$ (need not nonnegative).

Approaches to feasibility problem (FeasCP) [L. and Y. 2021]

[Our approach] We associate (FeasCP) with the following nonsmooth orthogonality optimization:

$$\max_{X \in \mathcal{O}_r} \{\min (BX)\} \equiv \min_{X \in \mathcal{O}_r} \{\max (-BX)\}. \qquad \text{(OptCP)}$$

CP Factorization via Orthogonality Constrained Problem

CP Factorization in Orthogonality Optimization via Smoothing Method

Smoothing method: original version for nonsmooth unconstrained problem

Nonsmooth unconstrained optimization

Consider

$$\min_{x \in \mathbb{R}^n} f(x), \tag{UnOpt}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is locally Lipschitz continuous on $\mathbb{R}^n$.

Smoothing method: original version for nonsmooth unconstrained problem

Nonsmooth unconstrained optimization

Consider

$$\min_{x \in \mathbb{R}^n} f(x), \tag{UnOpt}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is locally Lipschitz continuous on $\mathbb{R}^n$.

Recall that

1. The Clarke subdifferential is characterized by

$$\partial f(x) = \text{conv}\{v \mid \nabla f(x^k) \to v \text{ for } x^k \to x, f \text{ is differentiable at } x^k\}.$$

Smoothing method: original version for nonsmooth unconstrained problem

Nonsmooth unconstrained optimization

Consider

$$\min_{x \in \mathbb{R}^n} f(x), \qquad \text{(UnOpt)}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is locally Lipschitz continuous on $\mathbb{R}^n$.

Recall that

1. The Clarke subdifferential is characterized by

$$\partial f(x) = \text{conv}\{v \mid \nabla f(x^k) \to v \text{ for } x^k \to x, f \text{ is differentiable at } x^k\}.$$

2. The first-order optimality condition of (UnOpt) is Clarke stationary point:

$$\partial f(x) = 0.$$

Smoothing method: original version for nonsmooth unconstrained problem

Nonsmooth unconstrained optimization

Consider

$$\min_{x \in \mathbb{R}^n} f(x), \qquad \text{(UnOpt)}$$

where $f : \mathbb{R}^n \mapsto \mathbb{R}$ is locally Lipschitz continuous on $\mathbb{R}^n$.

Recall that

1. The Clarke subdifferential is characterized by

$$\partial f(x) = \text{conv}\{v \mid \nabla f(x^k) \to v \text{ for } x^k \to x, f \text{ is differentiable at } x^k\}.$$

2. The first-order optimality condition of (UnOpt) is Clarke stationary point:

$$\partial f(x) = 0.$$

3. Moreover, we call (UnOpt) smooth if $f(\cdot)$ is smooth, i.e., continuously differentiable.

Smoothing method: original version for nonsmooth unconstrained problem

Definition 3.1 [Chen, 2012]

We call $\tilde{f} : \mathbb{R}^n \times (0, \infty) \mapsto \mathbb{R}$ a smoothing function of f, if

Smoothing method: original version for nonsmooth unconstrained problem

Definition 3.1 [Chen, 2012]

We call $\tilde{f} : \mathbb{R}^n \times (0, \infty) \mapsto \mathbb{R}$ a smoothing function of f, if

1. $\tilde{f}(\cdot, \mu)$ is continuously differentiable on $\mathbb{R}^n$ for any fixed $\mu > 0$;

Smoothing method: original version for nonsmooth unconstrained problem

Definition 3.1 [Chen, 2012]

We call $\tilde{f} : \mathbb{R}^n \times (0, \infty) \mapsto \mathbb{R}$ a smoothing function of f, if

1. $\tilde{f}(\cdot, \mu)$ is continuously differentiable on $\mathbb{R}^n$ for any fixed $\mu > 0$;
2. for any $x \in \mathbb{R}^n$

$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) = f(x);$$

Smoothing method: original version for nonsmooth unconstrained problem

Definition 3.1 [Chen, 2012]

We call $\tilde{f} : \mathbb{R}^n \times (0, \infty) \mapsto \mathbb{R}$ a smoothing function of f, if

1. $\tilde{f}(\cdot, \mu)$ is continuously differentiable on $\mathbb{R}^n$ for any fixed $\mu > 0$;
2. for any $x \in \mathbb{R}^n$
$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) = f(x);$$
3. the gradient consistency holds, i.e., for any x
$$\partial f(x) = G_{\tilde{f}}(x) \triangleq \operatorname{conv}\{v \mid \nabla_x \tilde{f}(x^k, \mu_k) \to v \text{ for } x^k \to x, \mu_k \downarrow 0\},$$
where $G_{\tilde{f}}(x)$ is called subdifferential associated with $\tilde{f}$.

Smoothing method: original version for nonsmooth unconstrained problem

Definition 3.1 [Chen, 2012]

We call $\tilde{f} : \mathbb{R}^n \times (0, \infty) \mapsto \mathbb{R}$ a smoothing function of f, if

1. $\tilde{f}(\cdot, \mu)$ is continuously differentiable on $\mathbb{R}^n$ for any fixed $\mu > 0$;
2. for any $x \in \mathbb{R}^n$
$$\lim_{z \to x, \mu \downarrow 0} \tilde{f}(z, \mu) = f(x);$$
3. the gradient consistency holds, i.e., for any x
$$\partial f(x) = G_{\tilde{f}}(x) \triangleq \text{conv}\{v \mid \nabla_x \tilde{f}(x^k, \mu_k) \to v \text{ for } x^k \to x, \mu_k \downarrow 0\},$$
where $G_{\tilde{f}}(x)$ is called subdifferential associated with $\tilde{f}$.

For instance,

$$s(t, \mu) = \begin{cases} |t| & \text{if} \quad |t| > \frac{\mu}{2} \\ \frac{t^2}{\mu} + \frac{\mu}{4} & \text{if} \quad |t| \le \frac{\mu}{2} \end{cases}$$

is a smoothing function of $|t|$.

Smoothing method: original version for nonsmooth unconstrained problem

Algorithm 1: Smoothing Method for (UnOpt)

Initial step:

1. Find a smoothing function $\tilde{f}$ of f.
2. Select a **sub-algorithm** simply satisfying the weak global convergence condition,
$$\liminf_{k\to\infty} \|\nabla f(x^k)\| = 0 \tag{1}$$
for **smooth** (UnOpt).
3. Choose constants $\sigma \in (0,1), \gamma, \mu_0 > 0$ and $x^0 \in \mathbb{R}^n$. Set $k = 0$.

Smoothing method: original version for nonsmooth unconstrained problem

Algorithm 1: Smoothing Method for (UnOpt)

Initial step:

1. Find a smoothing function $\tilde{f}$ of f.
2. Select a **sub-algorithm** simply satisfying the weak global convergence condition,

$$\liminf_{k \to \infty} \|\nabla f(x^k)\| = 0 \tag{1}$$

   for **smooth** (UnOpt).
3. Choose constants $\sigma \in (0, 1), \gamma, \mu_0 > 0$ and $x^0 \in \mathbb{R}^n$. Set $k = 0$.

Inner iteration: Generate x^{k+1} from x^k by using the above algorithm to solve

$$\min_{x \in \mathbb{R}^n} \tilde{f}(x, \mu_k) \tag{2}$$

with a fixed $\mu_k > 0$.

Smoothing method: original version for nonsmooth unconstrained problem

Algorithm 1: Smoothing Method for (UnOpt)

Initial step:

1. Find a smoothing function $\tilde{f}$ of f.
2. Select a **sub-algorithm** simply satisfying the weak global convergence condition,

$$\liminf_{k\to\infty} \|\nabla f(x^k)\| = 0 \tag{1}$$

   for **smooth** (UnOpt).
3. Choose constants $\sigma \in (0,1), \gamma, \mu_0 > 0$ and $x^0 \in \mathbb{R}^n$. Set $k = 0$.

Inner iteration: Generate x^{k+1} from x^k by using the above algorithm to solve

$$\min_{x\in\mathbb{R}^n} \tilde{f}(x, \mu_k) \tag{2}$$

with a fixed $\mu_k > 0$.

Outer iteration: If

$$\|\nabla_x \tilde{f}(x^{k+1}, \mu_k)\| < \gamma\mu_k, \tag{3}$$

then set $\mu_{k+1} = \sigma\mu_k$; otherwise, set $\mu_{k+1} = \mu_k$.

Smoothing method: original version for nonsmooth unconstrained problem

Theorem 3.2 – Convergence of smoothing method for (UnOpt) [Chen, 2012, Theorem 3]

Any accumulation point generated by the smoothing method for (UnOpt) is a Clarke stationary point of (UnOpt).

Smoothing method: original version for nonsmooth unconstrained problem

Theorem 3.2 – Convergence of smoothing method for (UnOpt) [Chen, 2012, Theorem 3]

Any accumulation point generated by the smoothing method for (UnOpt) is a Clarke stationary point of (UnOpt).

Proof.

Suppose that the sub-algorithm in the inner iteration has the convergence property (1). In combination with the update scheme (3), we eventually obtain

$$\liminf_{k\to\infty} \|\nabla_x \tilde{f}(x^{k+1}, \mu_k)\| = 0. \tag{4}$$

If $\bar{x}$ is an accumulation point of $\{x^k\}$, then by

$$\partial f(x) = G_{\tilde{f}}(x) \triangleq \mathrm{conv}\{v \mid \nabla_x \tilde{f}(x^k, \mu_k) \to v \text{ for } x^k \to x, \mu_k \downarrow 0\},$$

we have $0 \in \partial f(\bar{x})$. □

CP Factorization in Orthogonality Optimization via Smoothing Method

Smoothing method: extend to nonsmooth orthogonality optimization

Nonsmooth orthogonality optimization

Now let us consider

$$\min_{X \in \mathrm{St}(n,p)} f(X), \tag{StOpt}$$

where the feasible set

$$\mathrm{St}(n,p) = \{X \in \mathbb{R}^{n \times p} \mid X^\top X = I\}$$

is the Stiefel manifold (i.e., orthogonality constraint).

Smoothing method: extend to nonsmooth orthogonality optimization

Nonsmooth orthogonality optimization

Now let us consider

$$\min_{X \in \mathrm{St}(n,p)} f(X), \tag{StOpt}$$

where the feasible set

$$\mathrm{St}(n,p) = \{X \in \mathbb{R}^{n \times p} \mid X^\top X = I\}$$

is the Stiefel manifold (i.e., orthogonality constraint).

For convenience, we call (StOpt) smooth if $f(\cdot)$ is continuously differentiable on $\mathbb{R}^{n \times p}$.

Smoothing method: extend to nonsmooth orthogonality optimization

Nonsmooth orthogonality optimization

Now let us consider

$$\min_{X \in \mathrm{St}(n,p)} f(X), \tag{StOpt}$$

where the feasible set

$$\mathrm{St}(n,p) = \{X \in \mathbb{R}^{n \times p} \mid X^\top X = I\}$$

is the Stiefel manifold (i.e., orthogonality constraint).

For convenience, we call (StOpt) smooth if $f(\cdot)$ is continuously differentiable on $\mathbb{R}^{n \times p}$.

Lemma 3.3 – An optimality condition of (StOpt) [L. and Y. 2021]

Suppose that X is a local minimizer of (StOpt). Then X satisfies the first-order optimality condition,

$$0 \in \partial f(X) - X \partial f(X)^\top X, \tag{5}$$

and we call such X a Clarke stationary point of (StOpt).

Smoothing method: extend to nonsmooth orthogonality optimization

Nonsmooth orthogonality optimization

Now let us consider

$$\min_{X \in \mathrm{St}(n,p)} f(X), \tag{StOpt}$$

where the feasible set

$$\mathrm{St}(n,p) = \{X \in \mathbb{R}^{n \times p} \mid X^\top X = I\}$$

is the Stiefel manifold (i.e., orthogonality constraint).

For convenience, we call (StOpt) smooth if $f(\cdot)$ is continuously differentiable on $\mathbb{R}^{n \times p}$.

Lemma 3.3 – An optimality condition of (StOpt) [L. and Y. 2021]

Suppose that X is a local minimizer of (StOpt). Then X satisfies the first-order optimality condition,

$$0 \in \partial f(X) - X \partial f(X)^\top X, \tag{5}$$

and we call such X a Clarke stationary point of (StOpt). In particular, if (StOpt) is smooth, it reduces to

$$0 = \nabla F(X) \triangleq \nabla f(X) - X \nabla f(X)^\top X.$$

Smoothing method: extend to nonsmooth orthogonality optimization

Algorithm 2: Smoothing Method for (StOpt)

Initial step:

1. Find a smoothing function $\tilde{f}$ of f.
2. Select a **sub-algorithm** simply satisfying the weak global convergence condition,

$$\liminf_{k\to\infty} \|\nabla F(X^k)\| = 0 \tag{6}$$

   for **smooth** (StOpt).

3. Choose constants $\sigma \in (0,1), \gamma, \mu_0 > 0$ and $X^0 \in \mathrm{St}(n,p)$. Set $k = 0$.

Inner iteration: Generate X^{k+1} from X^k by using the above algorithm to solve

$$\min_{X\in \mathrm{St}(n,p)} \tilde{f}(X, \mu_k) \tag{7}$$

with a fixed $\mu_k > 0$.

Outer iteration: If

$$\|\nabla_X \tilde{F}(X^{k+1}, \mu_k)\| < \gamma\mu_k, \tag{8}$$

then set $\mu_{k+1} = \sigma\mu_k$; otherwise, set $\mu_{k+1} = \mu_k$.

Smoothing method: extend to nonsmooth orthogonality optimization

Theorem 3.4 – Convergence of smoothing method for (StOpt) [L. and Y. 2021]

Any accumulation point generated by the smoothing method for (StOpt) is a Clarke stationary point of (StOpt).

This theorem is proved in a similar way as Theorem 3.2.

Smoothing method: extend to nonsmooth orthogonality optimization

Theorem 3.4 – Convergence of smoothing method for (StOpt) [L. and Y. 2021]

Any accumulation point generated by the smoothing method for (StOpt) is a Clarke stationary point of (StOpt).

This theorem is proved in a similar way as Theorem 3.2.

The remaining problems

Now, return back to our approach of CP factorization problem:

$$\min_{X \in \mathcal{O}_r} \{\max(-BX)\}. \tag{OptCP}$$

Smoothing method: extend to nonsmooth orthogonality optimization

Theorem 3.4 – Convergence of smoothing method for (StOpt) [L. and Y. 2021]

Any accumulation point generated by the smoothing method for (StOpt) is a Clarke stationary point of (StOpt).

This theorem is proved in a similar way as Theorem 3.2.

The remaining problems

Now, return back to our approach of CP factorization problem:

$$\min_{X \in \mathcal{O}_r} \{\max(-BX)\}. \tag{OptCP}$$

1. How do we select a sub-algorithm for **smooth** (StOpt)?

Smoothing method: extend to nonsmooth orthogonality optimization

Theorem 3.4 – Convergence of smoothing method for (StOpt) [L. and Y. 2021]

Any accumulation point generated by the smoothing method for (StOpt) is a Clarke stationary point of (StOpt).

This theorem is proved in a similar way as Theorem 3.2.

The remaining problems

Now, return back to our approach of CP factorization problem:

$$\min_{X \in \mathcal{O}_r} \{\max(-BX)\}. \tag{OptCP}$$

1. How do we select a sub-algorithm for **smooth** (StOpt)?
2. How do we select a smoothing function of the maximum function?

CP Factorization via Orthogonality Constrained Problem

CP Factorization in Orthogonality Optimization via Smoothing Method

A Curvilinear Search Method [Wen and Yin, 2013]

Smooth orthogonality optimization

Consider **smooth** orthogonality optimization:

$$\min_{X \in \mathrm{St}(n,p)} f(X). \tag{smooth-StOpt}$$

A Curvilinear Search Method [Wen and Yin, 2013]

Smooth orthogonality optimization

Consider **smooth** orthogonality optimization:

$$\min_{X \in \mathrm{St}(n,p)} f(X). \quad \text{(smooth-StOpt)}$$

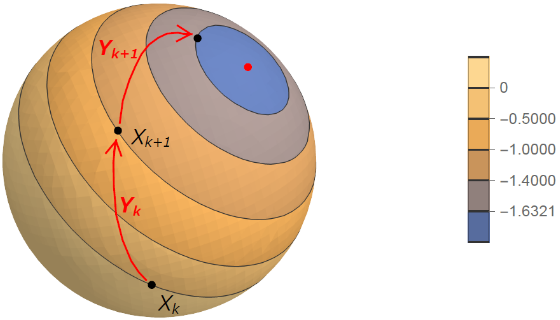

Figure 1: Illustration of unit sphere $\mathcal{M}_3^1$.

Gradient Descent (Linear Search) Method on $\mathbb{R}^n$.

$\xrightarrow{\text{extend}}$ Curvilinear Search Method on Manifold.

A Curvilinear Search Method [Wen and Yin, 2013]

Smooth orthogonality optimization

Consider **smooth** orthogonality optimization:

$$\min_{X \in \mathrm{St}(n,p)} f(X). \qquad \text{(smooth-StOpt)}$$

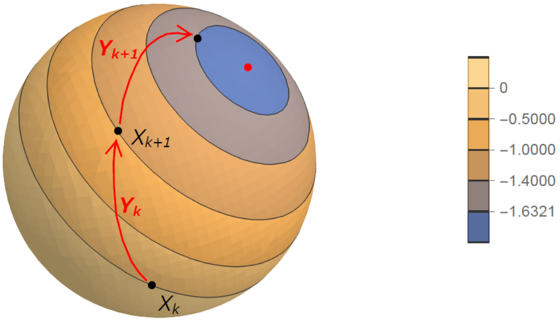

Figure 1: Illustration of unit sphere $\mathcal{M}_3^1$.

Gradient Descent (Linear Search) Method on $\mathbb{R}^n$.
$\xrightarrow{\text{extend}}$ Curvilinear Search Method on Manifold.

1. At a point X on $\mathrm{St}(n,p)$, we construct a smooth curve $Y(\tau) : \mathbb{R} \to \mathrm{St}(n,p)$ such that $Y(0) = X$.

A Curvilinear Search Method [Wen and Yin, 2013]

Smooth orthogonality optimization

Consider **smooth** orthogonality optimization:

$$\min_{X \in \mathrm{St}(n,p)} f(X). \tag{smooth-StOpt}$$

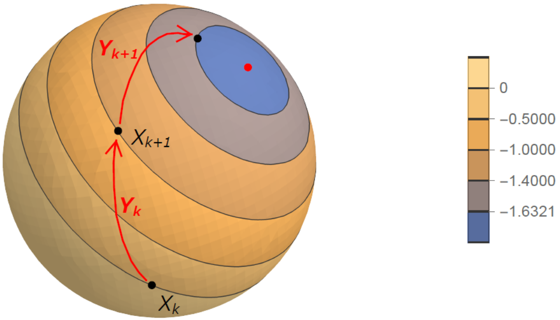

Figure 1: Illustration of unit sphere $\mathcal{M}_3^1$.

Gradient Descent (Linear Search) Method on $\mathbb{R}^n$.
$\xrightarrow{\text{extend}}$ Curvilinear Search Method on Manifold.

1. At a point X on $\mathrm{St}(n,p)$, we construct a smooth curve $Y(\tau) : \mathbb{R} \to \mathrm{St}(n,p)$ such that $Y(0) = X$.
2. If X is not a local minimizer, then $\exists \bar{\tau} \in \mathbb{R}, f(Y(\bar{\tau})) < f(Y(0))$. It is true if $\left.\frac{df(Y(\tau))}{d\tau}\right|_{\tau=0} < 0$ for $f(Y(\tau)) : \mathbb{R} \to \mathbb{R}$.

A Curvilinear Search Method [Wen and Yin, 2013]

Smooth orthogonality optimization

Consider **smooth** orthogonality optimization:

$$\min_{X \in \mathrm{St}(n,p)} f(X). \qquad \text{(smooth-StOpt)}$$

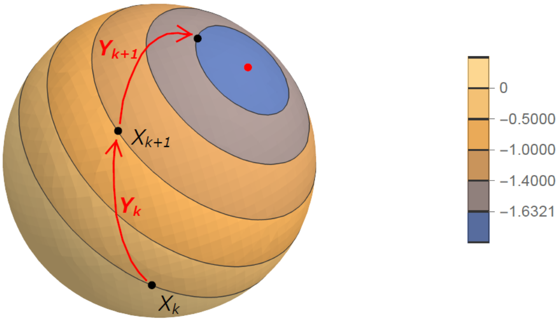

Figure 1: Illustration of unit sphere $\mathcal{M}_3^1$.

Gradient Descent (Linear Search) Method on $\mathbb{R}^n$.
$\xrightarrow{\text{extend}}$ Curvilinear Search Method on Manifold.

1. At a point X on $\mathrm{St}(n,p)$, we construct a smooth curve $Y(\tau) : \mathbb{R} \to \mathrm{St}(n,p)$ such that $Y(0) = X$.
2. If X is not a local minimizer, then $\exists \bar{\tau} \in \mathbb{R}, f(Y(\bar{\tau})) < f(Y(0))$. It is true if $\left.\frac{df(Y(\tau))}{d\tau}\right|_{\tau=0} < 0$ for $f(Y(\tau)) : \mathbb{R} \to \mathbb{R}$.
3. Get a better point $X := Y(\bar{\tau})$. Return to the first step until it is close to local minimizer.

A Curvilinear Search Method [Wen and Yin, 2013]

Lemma 3.3 has indicated that optimality condition of smooth (StOpt) is

$$0 = \nabla F(X) \triangleq \nabla f(X) - X \nabla f(X)^\top X.$$

Lemma 4.1 — Update Scheme [Wen and Yin, 2013, Lemma 3.]

1. X is a feasible point. Given any skew-symmetric matrix $W \in \mathbb{R}^{n\times n}$ (i.e. $W^\top = -W$), the matrix

$$Y(\tau) := \left(I + \frac{\tau}{2} W\right)^{-1} \left(I - \frac{\tau}{2} W\right) X$$

is a smooth curve $Y(\tau) : \mathbb{R} \to \mathrm{St}(n, p)$ such that $Y(0) = X$.

2. If set $W = A := \nabla f(X) X^\top - X \nabla f(X)^\top$. Then

$$(f \circ Y)'(0) = \left.\frac{df(Y(\tau))}{d\tau}\right|_{\tau=0} = -\frac{1}{2}\|A\|^2.$$

Note that $\nabla F(X) = AX$ and $\nabla F(X) = 0$ if and only if $A = 0$.

A Curvilinear Search Method

Algorithm 3: Curvilinear Search for (smooth-StOpt)

Set $0 < c_1 < c_2 < 1, \epsilon > 0, X^0 \in \mathcal{O}_r, k \leftarrow 0.$;

while $\|\nabla F(X^k)\| > \epsilon$ **do**

Generate $A_k \leftarrow \nabla f(X^k) X^{k\top} - X^k \nabla f(X^k)^\top, W_k \leftarrow A_k$;

Find a step size $\tau_k > 0$ that satisfies the Armijo-Wolfe conditions:

$$(f \circ Y_k)(\tau_k) \leq (f \circ Y_k)(0) + c_1 \tau_k (f \circ Y_k)'(0), \tag{9a}$$

$$(f \circ Y_k)'(\tau_k) \geq c_2 (f \circ Y_k)'(0); \tag{9b}$$

Set $X_{k+1} \leftarrow Y_k(\tau_k)$, $k \leftarrow k+1$ and continue;

end

Theorem 4.2 – Convergence of Algorithm 5 [Wen and Yin, 2013, Theorem 2]

$$\lim_{k \to \infty} \|\nabla F(X_k)\| = 0.$$

CP Factorization in Orthogonality Optimization via Smoothing Method

LogSumExp: Smooth Approximation to Max Function

Example (LogSumExp is a smoothing function of maximum function)

The LogSumExp function, $\text{lse}(x, \mu) : \mathbb{R}^n \times (0, \infty) \mapsto \mathbb{R}$, is given by

$$\text{lse}(x, \mu) = \mu \log(\textstyle\sum_{i=1}^{n} \exp(x_i/\mu)).$$

LogSumExp: Smooth Approximation to Max Function

Example (LogSumExp is a smoothing function of maximum function)

The LogSumExp function, $\text{lse}(x,\mu) : \mathbb{R}^n \times (0,\infty) \mapsto \mathbb{R}$, is given by

$$\text{lse}(x,\mu) = \mu \log(\textstyle\sum_{i=1}^n \exp(x_i/\mu)).$$

Propostion 4.3 – Properties of LogSumExp 1 [L. and Y. 2021]

$\text{lse}(\cdot,\mu)$ is continuously differentiable on $\mathbb{R}^n$ for any fixed $\mu > 0$. In particular, $\nabla_x \text{lse}(x,\mu)$ is the so-called softmax function, given by $\sigma(\cdot,\mu) : \mathbb{R}^n \mapsto \Delta^{n-1}$,

$$\nabla_x \text{lse}(x,\mu) = \sigma(x,\mu) := \frac{1}{\sum_{l=1}^n \exp(x_l/\mu)} \begin{bmatrix} \exp(x_1/\mu) \\ \vdots \\ \exp(x_n/\mu) \end{bmatrix}, \tag{10}$$

where $\Delta^{n-1} := \{x \in \mathbb{R}^n \mid \sum_{i=1} x_i = 1, x_i \geq 0\}$ is the *unit simplex*.

LogSumExp: Smooth Approximation to Max Function

Example (LogSumExp is a smoothing function of maximum function)

The LogSumExp function, $\operatorname{lse}(x, \mu) : \mathbb{R}^n \times (0, \infty) \mapsto \mathbb{R}$, is given by

$$\operatorname{lse}(x, \mu) = \mu \log(\textstyle\sum_{i=1}^{n} \exp(x_i/\mu)).$$

Propostion 4.3 – Properties of LogSumExp 2 [L. and Y. 2021]

For all $x \in \mathbb{R}^n$ and $\mu > 0$, we have

$$\max(x) < \operatorname{lse}(x, \mu) \leq \max(x) + \mu \log(n).$$

The above inequalities imply that for any $x \in \mathbb{R}^n$, $\lim_{z \to x, \mu \downarrow 0} \operatorname{lse}(z, \mu) = \max(x)$.

LogSumExp: Smooth Approximation to Max Function

Example (LogSumExp is a smoothing function of maximum function)

The LogSumExp function, $\mathrm{lse}(x,\mu) : \mathbb{R}^n \times (0,\infty) \mapsto \mathbb{R}$, is given by

$$\mathrm{lse}(x,\mu) = \mu \log(\textstyle\sum_{i=1}^n \exp(x_i/\mu)).$$

Propostion 4.3 – Properties of LogSumExp 2 [L. and Y. 2021]

For all $x \in \mathbb{R}^n$ and $\mu > 0$, we have

$$\max(x) < \mathrm{lse}(x,\mu) \leq \max(x) + \mu \log(n).$$

The above inequalities imply that for any $x \in \mathbb{R}^n$, $\lim_{z\to x,\mu\downarrow 0} \mathrm{lse}(z,\mu) = \max(x)$.

For instance, let $x = (2, 5, -1, 3)$.

$n = 4$	$\mu = 1$	$\mu = 1/2$	$\mu = 1/4$	$\mu = 1/8$
$\mathrm{lse}(x,\mu)$	5.1719	5.0103	5.0001	5.0000

Table 1: Example of approximation effect with different parameters μ.

LogSumExp: Smooth Approximation to Max Function

Example (LogSumExp is a smoothing function of maximum function)

The LogSumExp function, $\text{lse}(x,\mu):\mathbb{R}^n\times(0,\infty)\mapsto\mathbb{R}$, is given by

$$\text{lse}(x,\mu)=\mu\log(\textstyle\sum_{i=1}^n\exp(x_i/\mu)).$$

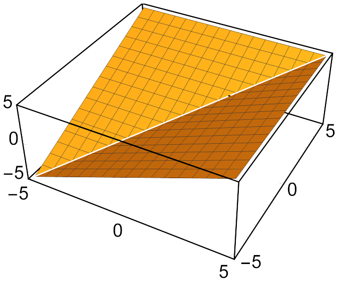

Figure 2: Graph of $\max(x,y)$.

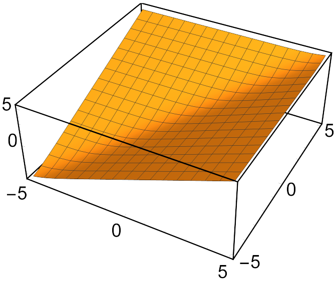

Figure 3: Graph of $\log(e^x+e^y)$.

LogSumExp: Smooth Approximation to Max Function

Example (LogSumExp is a smoothing function of maximum function)

The LogSumExp function, $\mathrm{lse}(x,\mu) : \mathbb{R}^n \times (0,\infty) \mapsto \mathbb{R}$, is given by

$$\mathrm{lse}(x,\mu) = \mu \log(\textstyle\sum_{i=1}^{n} \exp(x_i/\mu)).$$

Propostion 4.3 – Properties of LogSumExp 3 [L. and Y. 2021]

The gradient consistency

$$\partial \max(x) = G_{\mathrm{lse}}(x)$$

holds for any $x \in \mathbb{R}^n$. In other words,

$$\mathrm{conv}\{e_i \mid i \in \mathcal{I}(x)\} = \mathrm{conv}\{\lim_{x^k \to x, \mu_k \downarrow 0} \sigma(x^k, \mu_k)\},$$

where e_i is a standard unit vector and $\mathcal{I}(x) = \{i \mid i \in \{1, \cdots, n\}, x_i = \max(x)\}$.

CP Factorization via Orthogonality Constrained Problem

Numerical Results – Experiment 1

Example (Experiment 1 – Randomly generated instances)

We computed C by setting $C_{ij} := |B_{ij}|$ for all i, j, where B is a random $n \times 2n$ matrix based on the Matlab command – randn, and we took $A = CC^\top$ to be factorized.

- Intel Core i7-10700 @ 2.90GHz 2.90GHz and 16GB RAM, and MatlabR2021a.
- We take $n \in \{20, 30, 40, 100, 200, 400, 600, 800\}$ and $r = t * n$ for $t \in \{1.5, 3\}$.
- For each pair of n and r, we generated 50 instances if $n \leq 100$ and 10 instances otherwise.
- For each instance, we initialized all the algorithms at the same random starting point X^0 and initial decomposition B, except for [Boț and Nguyen, 2021].

Table 2: Experiment 1 – CP factorization of random completely positive matrices.

Method		Our method			[Chen et al., 2020]			[Boţ and Nguyen, 2021]			[Groetzner and Dür, 2020]		
n	$r = 1.5n$	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s
20	30	1.00	0.0139	60	1.00	0.0027	24	1.00	0.0081	1229	0.32	0.3502	2318
30	45	1.00	0.0425	72	1.00	0.0075	24	1.00	0.0231	1481	0.04	1.0075	2467
40	60	1.00	0.0641	75	1.00	0.0216	46	1.00	0.0574	1990	0.00	-	-
100	150	1.00	0.4087	92	1.00	0.2831	109	1.00	0.8169	4912	0.00	-	-
200	300	1.00	1.7768	116	1.00	2.2504	212	1.00	5.2908	9616	0.00	-	-
400	600	1.00	15.7512	147	1.00	36.9650	636	1.00	90.6752	17987	0.00	-	-
600	900	1.00	50.3576	177	1.00	140.0720	882	1.00	344.7035	26146	0.00	-	-
800	1200	1.00	114.5538	190	1.00	413.3798	1225	1.00	891.1210	34022	0.00	-	-
Method		Our method			[Chen et al., 2020]			[Boţ and Nguyen, 2021]			[Groetzner and Dür, 2020]		
n	$r = 3n$	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s
20	60	1.00	0.0399	64	1.00	0.0057	15	1.00	0.0105	1062	0.30	0.7267	2198
30	90	1.00	0.0853	69	1.00	0.0128	17	1.00	0.0336	1127	0.00	-	-
40	120	1.00	0.1518	74	1.00	0.0256	19	1.00	0.0822	1460	0.00	-	-
100	300	1.00	1.1297	87	1.00	0.8115	86	1.00	1.1909	4753	0.00	-	-
200	600	1.00	8.0160	110	1.00	8.1517	184	1.00	9.2248	9402	0.00	-	-
400	1200	1.00	64.1486	149	1.00	124.3410	453	1.00	156.6019	17563	0.00	-	-
600	1800	1.00	260.2481	187	1.00	981.8537	795	1.00	616.7851	25336	0.00	-	-
800	2400	1.00	574.6292	216	1.00	4027.4278	1070	1.00	1289.3736	26820	0.00	-	-

(Rate) success rate relative to the total number of instances.

(Time_s) average time in *seconds* among successful instances.

(Iter_s) average number of iterations among successful instances.

Numerical Results – Experiment 2

Example (Experiment 2 – A specically structured instance)

Let e_n denote the all-ones vector in $\mathbb{R}^n$ and consider the matrix,

$$A_n = \begin{pmatrix} 0 & e_{n-1}^\top \\ e_{n-1} & I_{n-1} \end{pmatrix}^\top \begin{pmatrix} 0 & e_{n-1}^\top \\ e_{n-1} & I_{n-1} \end{pmatrix} \in \mathcal{CP}_n.$$

By construction, $\mathrm{cp}(A_n) = n$.

- Intel Core i7-10700 @ 2.90GHz 2.90GHz and 16GB RAM, and MatlabR2021a.
- We tried to factorize A_n for $n \in \{10, 20, 50, 75, 100, 150\}$ using $r = \mathrm{cp}(A_n) = n$.
- For each A_n, and the same initial decomposition B, we tested all the algorithms on the same 50 randomly generated starting points, except for [Boţ and Nguyen, 2021].

Table 3: Experiment 2 – CP factorization of a family of specifically structured instances.

Method	Our method			[Chen et al., 2020]			[Boț and Nguyen, 2021]			[Groetzner and Dür, 2020]		
$n = r$	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s	Rate	Time_s	Iter_s
10	1.00	0.0052	69	1.00	0.0043	149	1.00	0.0074	2085	0.80	0.0174	616
20	1.00	0.0168	107	0.98	0.0139	201	0.74	0.0212	3478	0.90	0.0591	864
50	1.00	0.1090	125	0.98	0.3389	770	0.00	-	-	0.76	0.6948	1416
75	1.00	0.2314	139	0.98	1.0706	1186	0.00	-	-	0.64	1.4809	1510
100	1.00	0.5118	185	0.80	1.6653	1083	0.00	-	-	0.60	2.8150	1690
150	1.00	1.5551	265	0.70	3.7652	1170	0.00	-	-	0.35	9.9930	2959

(Rate) success rate relative to the total number of instances.

(Time_s) average time in *seconds* among successful instances.

(Iter_s) average number of iterations among successful instances.

Conclusions and Future work

Given $A \in \mathcal{CP}_n$, CP factorization problem is

$$\text{Find } B \text{ s.t. } A = BB^\top \text{ and } B \geq 0. \tag{CPfact}$$

1. Reformulation of [Groetzner and Dür, 2020]:

$$\text{Find } X \text{ s.t. } BX \geq 0 \text{ and } X \in \mathcal{O}_r, \tag{FeasCP}$$

   where $B \in \mathbb{R}^{n \times r}$ is an initial factorization $A = BB^\top$.

2. Treat it as nonsmooth orthogonality optimization [L. and Y. 2021]:

$$\min_{X \in \mathcal{O}_r} \{\max(-BX)\}, \tag{OptCP}$$

   and use **extended smoothing method** where **curvilinear search method** $\rightarrow$ the sub-algorithm; **lse**$(\cdot, \mu) \rightarrow$ the smoothing function of max$(\cdot)$.

3. Numerical experiments show the efficiency of our method especially for large-scale factorizations.

Conclusions and Future work

Given $A \in \mathcal{CP}_n$, CP factorization problem is

$$\text{Find } B \text{ s.t. } A = BB^\top \text{ and } B \geq 0. \tag{CPfact}$$

1. Reformulation of [Groetzner and Dür, 2020]:

$$\text{Find } X \text{ s.t. } BX \geq 0 \text{ and } X \in \mathcal{O}_r, \tag{FeasCP}$$

   where $B \in \mathbb{R}^{n \times r}$ is an initial factorization $A = BB^\top$.
2. Treat it as nonsmooth orthogonality optimization [L. and Y. 2021]:

$$\min_{X \in \mathcal{O}_r} \{\max(-BX)\}, \tag{OptCP}$$

   and use **extended smoothing method** where **curvilinear search method** $\rightarrow$ the sub-algorithm; **lse**$(\cdot, \mu) \rightarrow$ the smoothing function of $\max(\cdot)$.
3. Numerical experiments show the efficiency of our method especially for large-scale factorizations.

Future work

1. Other sub-algorithms;
2. Other techniques of nonsmooth Riemannian optimization for (OptCP).

Thank you for listening.

Reference I

[Abraham and Naomi, 2003] Abraham, B. and Naomi, S.-m. (2003).
Completely Positive Matrices.
World Scientific.

[Berman et al., 2015] Berman, A., Dür, M., and Shaked-Monderer, N. (2015).
Open problems in the theory of completely positive and copositive matrices.
Electronic Journal of Linear Algebra, 29(1):46–58.

[Bomze et al., 2015] Bomze, I. M., Dickinson, P. J., and Still, G. (2015).
The structure of completely positive matrices according to their cp-rank and cp-plus-rank.
Linear algebra and its applications, 482:191–206.

[Bomze et al., 2000] Bomze, I. M., Dür, M., De Klerk, E., Roos, C., Quist, A. J., and Terlaky, T. (2000).
On copositive programming and standard quadratic optimization problems.
Journal of Global Optimization, 18(4):301–320.

[Boţ and Nguyen, 2021] Boţ, R. I. and Nguyen, D.-K. (2021).
Factorization of completely positive matrices using iterative projected gradient steps.
Numerical Linear Algebra with Applications, page e2391.

Reference II

[Chen et al., 2020] Chen, C., Pong, T. K., Tan, L., and Zeng, L. (2020).
A difference-of-convex approach for split feasibility with applications to matrix factorizations and outlier detection.
Journal of Global Optimization, pages 1–30.

[Chen, 2012] Chen, X. (2012).
Smoothing methods for nonsmooth, nonconvex minimization.
Mathematical programming, 134(1):71–99.

[De Klerk and Pasechnik, 2002] De Klerk, E. and Pasechnik, D. V. (2002).
Approximation of the stability number of a graph via copositive programming.
SIAM Journal on Optimization, 12(4):875–892.

[Dickinson, 2010] Dickinson, P. J. (2010).
An improved characterisation of the interior of the completely positive cone.
Electronic Journal of Linear Algebra, 20(1):723–729.

[Groetzner and Dür, 2020] Groetzner, P. and Dür, M. (2020).
A factorization method for completely positive matrices.
Linear Algebra and its Applications, 591:1–24.

Reference III

[Wen and Yin, 2013] Wen, Z. and Yin, W. (2013).
A feasible method for optimization with orthogonality constraints.
Mathematical Programming, 142(1-2):397–434.

[Xu, 2004] Xu, C. (2004).
Completely positive matrices.
Linear algebra and its applications, 379:319–327.

Appendix: Copositive program (primal and dual problem)

Consider the so-called copositive program (primal problem)

$$\min \{\langle C, X\rangle \mid \langle A_i, X\rangle = b_i \ (i = 1, \ldots, m), X \in \mathcal{COP}_n\}, \quad (11)$$

where

$$\mathcal{COP}_n \triangleq \{A \in \mathcal{S}_n | x^T A x \geq 0 \text{ for all } x \in \mathbb{R}^n_+\}$$

is the cone of so-called copositive matrices. The dual problem of (11) is

$$\max \left\{\sum_{i=1}^m b_i y_i \mid C - \sum_{i=1}^m y_i A_i \in \mathcal{CP}_n, \ y_i \in \mathbb{R}\right\}, \quad (12)$$

where $\mathcal{CP}_n$ denotes the set of $n \times n$ completely positive matrices.

Dual cones, cf. [Abraham and Naomi, 2003]

$\mathcal{CP}_n$ and $\mathcal{COP}_n$ are mutually dual to each other.

Appendix: Relationship with other matrix cones

1. The cone of so-called copositive matrices: $\mathcal{COP}_n := \left\{A \in \mathcal{S}_n | x^T Ax \geq 0 \text{ for all } x \in \mathbb{R}^n_+\right\}$.
2. The cone of symmetric entrywise nonnegative matrices:
   $\mathcal{N}_n := \{A \in \mathcal{S}_n \mid A_{ij} \geq 0 \text{ for all } i,j = 1,\ldots,n\}$.
3. The cone of symmetric positive semidefinite matrices: $\mathcal{S}^+_n := \left\{A \in \mathcal{S}_n \mid x^\top Ax \geq 0 \text{ for all } x \in \mathbb{R}^n\right\}$.
4. The cone of doubly nonnegative matrices: $\mathcal{DNN}_n := \mathcal{S}^+_n \cap \mathcal{N}_n$.
5. The Minkowski sum of $\mathcal{S}^+_n$ and $\mathcal{N}_n$: $\mathcal{S}^+_n + \mathcal{N}_n := \{A + B \in \mathcal{S}_n \mid A \in \mathcal{S}^+_n, B \in \mathcal{N}_n\}$.

Relationship with other matrix cones

$$\mathcal{CP}_n \subseteq \mathcal{S}^+_n \cap \mathcal{N}_n \subseteq \mathcal{S}^+_n \subseteq \mathcal{S}^+_n + \mathcal{N}_n \subseteq \mathcal{COP}_n \tag{13}$$

Appendix: cp-plus-rank and int$(\mathcal{CP}^n)$

Definition – cp-rank and the cp-plus-rank

We define cp-rank of $A \in \mathcal{S}_n$ as the minimum of the number of columns for all CP-factor of A, that is

$$\text{cp}(A) := \min_B \left\{ r \in \mathbb{N} | \exists B \in \mathbb{R}^{n\times r}, B \geq 0, A = BB^\top \right\}.$$

Notice that $\text{cp}(A) := \infty$ if $A \notin \mathcal{CP}_n$. We also define cp-plus-rank of $A \in \mathcal{S}_n$ as

$$\text{cp}^+(A) := \min_B \left\{ r \in \mathbb{N} | \exists B \in \mathbb{R}^{n\times r}, B > 0, A = BB^\top \right\}.$$

Theorem – Interior of $\mathcal{CP}^n$ [Dickinson, 2010, Theorem 3.8]

$$\text{int}\,(\mathcal{CP}^n) = \{ A \in \mathbb{S}_n \mid \text{cp}^+(A) < \infty \text{ and } \text{rank}(A) = n \}$$

Lemma – Upper bound of cp-rank [Bomze et al., 2015, Theorem 4.1]

For all $A \in \mathcal{CP}_n$, we have $\text{cp}(A) \leq \text{cp}_n := \begin{cases} n & \text{for } n \in \{2,3,4\} \\ \frac{1}{2}n(n+1) - 4 & \text{for } n \geq 5 \end{cases}$

Bot's method [Boț and Nguyen, 2021]

Constrained minimization problem

$$\min_{x \in \mathcal{Q} \subset \mathbb{R}^n} f(x).$$

Starting from an initial point $x_0 \in \mathcal{Q}$, **Projected Gradient Method** iterates

$$x_{k+1} = P_{\mathcal{Q}}\left(x_k - t_k \nabla f\left(x_k\right)\right),$$

where $\mathrm{Pr}_{\mathcal{Q}}(.)$ is the projection operator:

$$\mathrm{Pr}_{\mathcal{Q}}\left(x_0\right) = \arg\min_{x \in \mathcal{Q}} \frac{1}{2}\left\|x - x_0\right\|_2^2.$$

Bot's method [Boț and Nguyen, 2021]

Recall the CP factorization problem:

$$\text{Find } B \in \mathbb{R}^{n\times r} \text{ s.t. } A = BB^\top \text{ and } B \geq 0. \tag{CPfact}$$

Another approach to the CP factorization problem

[Boț and Nguyen, 2021] proposed a *projected gradient method* with relaxation and inertia parameters for (CPfact), aimed at solving

$$\min_X \{\|A - XX^\top\|^2 \mid X \in \mathbb{R}_+^{n\times r} \cap \mathcal{B}(0, \sqrt{\text{trace}(A)})\},$$

where $\mathcal{B}(0,\varepsilon) := \{X \in \mathbb{R}^{n\times r} \mid \|X\| \leq \varepsilon\}$ is the closed ball centered at 0.

Indeed, if X satisfies $A = XX^T$, then

$$\|X\| \leq \sqrt{\text{trace}(A)}.$$

The authors argued that its optimal value is zero if and only if $A \in \mathcal{CP}_n$.

Bot's method [Boț and Nguyen, 2021]

For a given nonzero completely positive matrix $A \in \mathbb{R}^{n \times n}$,

$$\begin{aligned} \min_{X \in \mathbb{R}^{n \times r}} \ & \mathcal{E}(X) := \frac{1}{2} \left\| A - XX^T \right\|_F^2 \\ \text{s.t.} \ & X \in \mathcal{D} := \mathbb{R}_+^{n \times r} \cap \mathbb{B}_F(\mathbf{0}, \sqrt{\text{trace}(A)}) \end{aligned} \tag{P_{mod}}$$

Remark: $\mathcal{E}(X)$ is a nonconvex smooth function.

Proposition 1.

Let $A \in \mathcal{CP}_n$. The set $\mathcal{D}$ is nonempty convex and closed, and for any $X \in \mathbb{R}^{n \times r}$ it holds

$$\text{Pr}_D(X) = \frac{\sqrt{\text{trace}(A)}}{\max \left\{ \|[X]_+\|_{\mathrm{F}}, \sqrt{\text{trace}(A)} \right\}} [X]_+$$

where $[X]_+ := \max\{X, \mathbf{0}\}$ and the max operator is understood entrywise.

Bot's method [Boț and Nguyen, 2021]

Example 1. For every $X \in \mathbb{R}^{n\times r}$,
1. if $\mathcal{D} := \mathbb{R}_+^{n\times r}$, then it holds

$$\Pr_D(X) = [X]_+ := \max\{X, \mathbf{0}\}$$

where the max operator is understood entrywise;
2. if $\mathcal{D} := \mathbb{B}_F(\mathbf{0}; \varepsilon)$ for $\varepsilon > 0$, we have

$$\Pr_D(X) = \frac{\varepsilon}{\max\{\|X\|_{\mathrm{F}}, \varepsilon\}} X$$

Example 2. Let $\varepsilon > 0$ and K be a nonempty closed convex cone in $\mathbb{R}^{n\times r}$. Then the projection onto the intersection $K \cap \mathbb{B}_F(0, \varepsilon)$ is given by

$$\Pr_{K\cap\mathbb{B}_F(0,\varepsilon)}(X) = \Pr_{\mathbb{B}_F(0,\varepsilon)} \circ \Pr_K(X) = \frac{\varepsilon}{\max\{\|\Pr_K(X)\|_F, \varepsilon\}} \Pr_K(X) \quad \forall X \in \mathbb{R}^{n\times r}$$

Notice that in general $\Pr_{\mathbb{B}_F(0,\varepsilon)} \circ \Pr_K(X) \neq \Pr_K(X) \circ \Pr_{\mathbb{B}_F(0,\varepsilon)}$.

A projected gradient algorithm with relaxation and inertial parameters

Initialization: given starting points $X_1 := X_0 \in \mathcal{D}$ and; a sequence of *inertial parameters* $\{\alpha_k\}_{k\geq 1} \subseteq [0,1]$, for which we set $\alpha_+ := \sup_{k\geq 0} \alpha_k$ and

$$L_F(\alpha_+) := 2\left[\left(3 + 8\alpha_+ + 6\alpha_+^2\right)\mathrm{trace}(A) - \lambda_{\min}(A)\right] > 0;$$

a *relaxation parameter* $\rho \in (0,1]$ chosen such that

$$0 < \frac{\sqrt{L_F(\alpha_+) + 2\|A\|_2}}{\sqrt{L_F(\alpha_+) + 2\|A\|_2} + \sqrt{L_F(\alpha_+)}} < \rho < \frac{\sqrt{L_F(\alpha_+) + 2\|A\|_2}}{(1+\alpha_+)\sqrt{L_F(\alpha_+) + 2\|A\|_2} - \sqrt{L_F(\alpha_+)}}.$$

Main iterate: Set $k := 1$. step 1: Compute

$$Y_k := X_k + \alpha_k(X_k - X_{k-1})$$
$$Z_{k+1} := \mathrm{Pr}_D\left(Y_k - \frac{1}{L_F(\alpha_+)}\nabla\mathcal{E}(Y_k)\right),$$
$$X_{k+1} := (1-\rho)X_k + \rho Z_{k+1}$$

Step 2 : If a stopping criterion is not met, then set $k := k+1$ and go to step 1.