Solving Second-order Cone Programs in Matrix Multiplication Time

Michelle Wei
Mentor: Guanghao Ye

MIT PRIMES-USA

October 14-15, 2023
MIT PRIMES Conference

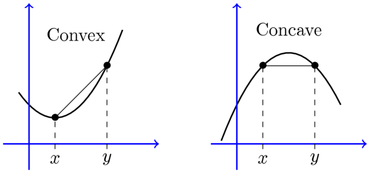
Outline

- Background: convex optimization and second-order cone programming (SOCP)
- Existing algorithms: interior point methods
- My work: developing an efficient SOCP algorithm

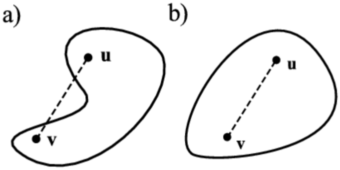
What is convex optimization?

Minimization of a convex function $f(x)$ over a convex set C.

$$\min_{x \in C} f(x)$$

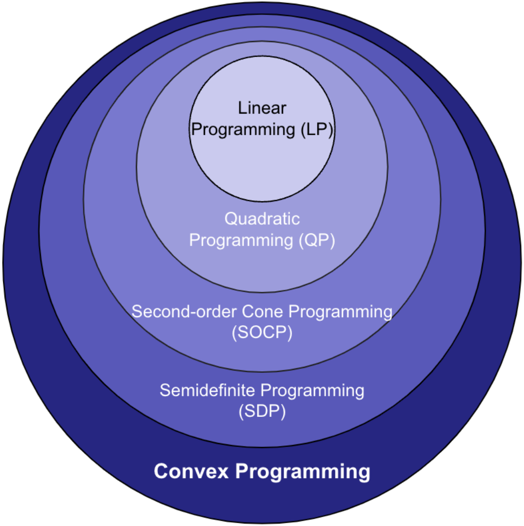


Convex function



Convex set

Subsets of Convex Optimization



Applications of Second-order Cone Programming

Second-order cone programs provide a general framework for solving a wide range of linear and quadratic programming problems with many applications in:

- Financial portfolio optimization
- Engineering and control systems
- Energy management systems
- Logistics and supply chain management
- Machine learning algorithms, such as support vector machines

Second-order Cone Definition

Definition (Second-order Cone)

A second-order cone $\mathcal{L}^k$ is defined as

$$\{(x_0, \widetilde{\boldsymbol{x}}), \widetilde{\boldsymbol{x}} \in \mathbb{R}^k : \|\widetilde{\boldsymbol{x}}\|_2 \leq x_0\}.$$

Euclidean norm: $\|x\|_2 = \sqrt{x_1^2 + \cdots + x_n^2}$.

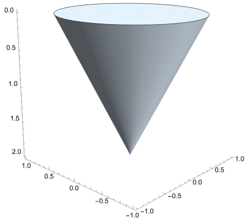
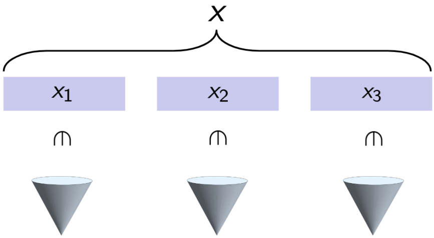


Figure: The second-order cone $\mathcal{L}^2$ is equivalent to the inequality $\sqrt{x^2 + y^2} \leq z$.

Second-order Cone Program Definition

- **Objective function:** Linear function $\boldsymbol{c}^\top \boldsymbol{x}$
- **Constraint function:** Intersection of an affine set $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ and the Cartesian product $\mathcal{L}$ of second-order cones.



$$\mathcal{L} = \mathcal{L}_1 \times \mathcal{L}_2 \times \mathcal{L}_3$$

Second-order Cone Program Formal Definition

Definition (Second-order Cone Program)

Given the constraint matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, two vectors $\boldsymbol{b} \in \mathbb{R}^m$ and $\boldsymbol{c} \in \mathbb{R}^n$, and r second-order cones $\mathcal{L}_1, \ldots, \mathcal{L}_r$. The optimization problem can be expressed as:

$$\min \boldsymbol{c}^\top \boldsymbol{x} \text{ subject to } \mathbf{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x}_i \in \mathcal{L}_i \text{ for all } i \in [r], \tag{1}$$

where $\boldsymbol{x}$ is the concatenation of $\boldsymbol{x}_i$ lying inside the domain $\mathcal{L} \stackrel{\text{def}}{=} \mathcal{L}_1 \times \cdots \times \mathcal{L}_r$ and each $\mathcal{L}_i \in \mathbb{R}^{n_i}$ is a second-order cone.

Previous Work and Our Result

Karmarkar	Linear program	1984	$O(n^{3.5})$
Nesterov, Nemirovski	Second-order cone program	1994	$O(n^{\omega+0.5})$
Lee, Song, Zhang	Constant dimension convex program	2019	$O(n^{\omega})$
Cohen, Lee, Song	Linear program	2021	$O(n^{\omega})$
Gu, Song, Zhang	Quadratic program	2023	$O(n^{\omega})$
Our result	**Second-order cone program**	**2023**	$\boldsymbol{O(n^{\omega})}$

My Approach

Developed a second-order cone programming algorithm that runs in matrix multiplication time.

- Applied approximation techniques to reduce the runtime of each iteration.
- Developed a novel approach to decompose large cone constraints into smaller ones.
- Utilized self-concordance properties to prove that the algorithm converges in matrix multiplication time.

Interior Point Methods: Duality

Definition

Given an SOCP of the form

$$\min_{\mathbf{A}\boldsymbol{x}=\boldsymbol{b},\boldsymbol{x}\in\mathcal{L}} \boldsymbol{c}^\top \boldsymbol{x}$$

the *dual* of this SOCP is the new SOCP

$$\max_{\mathbf{A}^\top\boldsymbol{y}+\boldsymbol{s}=\boldsymbol{c},\boldsymbol{s}\in\mathcal{L}} \boldsymbol{b}^\top \boldsymbol{y}.$$

We call the original SOCP a *primal* SOCP.

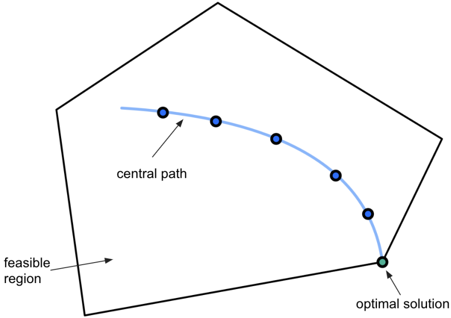
Theorem (Complementary Slackness)

Any feasible x and s are optimal if and only if $\boldsymbol{x}^\top\boldsymbol{s} = 0$.

Interior Point Methods: Central Path

In the IPM, we start with a feasible solution pair $(\boldsymbol{x}, \boldsymbol{s})$ and follow a *central path* to the solution. While the duality gap $\boldsymbol{x}^\top \boldsymbol{s} > \epsilon$, where ϵ is the error we tolerate, perform the following steps:

- Compute the next point $(\boldsymbol{x} + \delta_x, \boldsymbol{s} + \delta_s)$ to decrease the duality gap.
- Update $(\boldsymbol{x}, \boldsymbol{s})$ to the new point.



Interior Point Methods: Central Path

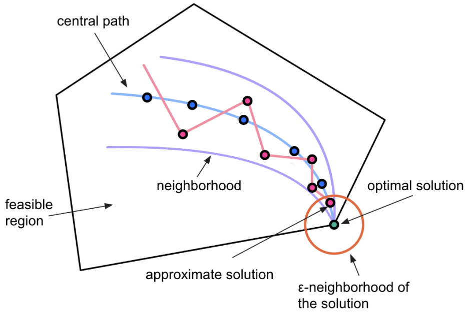
The interior point method follows the central path $\boldsymbol{x}(t)$ which starts at some interior point ($t \gg 0$) to the optimal solution ($t = 0$):

$$\boldsymbol{x}(t) = \arg\min_{\mathbf{A}\boldsymbol{x}=\boldsymbol{b}} \boldsymbol{c}^\top \boldsymbol{x} + t\phi(\boldsymbol{x}) \quad \text{with } \phi(\boldsymbol{x}) \stackrel{\text{def}}{=} \sum_{i=1}^{r} \phi_i(\boldsymbol{x}_i),$$

where $\phi_i : \mathcal{L}_i \to \mathbb{R}$ are *barrier functions*: they increase rapidly near the border of each second-order cone.

Interior Point Methods: Approximate Solution

Because it is costly to compute $(\boldsymbol{x}, \boldsymbol{s})$ exactly at each iteration, we use an approximate solution $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{s}})$ that remains within a small neighborhood of the central path.



To ensure $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{s}})$ remains close to $(\boldsymbol{x}, \boldsymbol{s})$, we update certain blocks of $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{s}})$ at each step.

Interior Point Methods: Optimality Conditions

Theorem (Karush–Kuhn–Tucker condition)

The optimal condition of the path satisfies

$$\frac{1}{t}\boldsymbol{s} + \nabla\phi(\boldsymbol{x}) = 0.$$

We denote $\mu = \boldsymbol{s}/t + \nabla\phi(\boldsymbol{x})$, which serves as a measure of proximity to the central path.

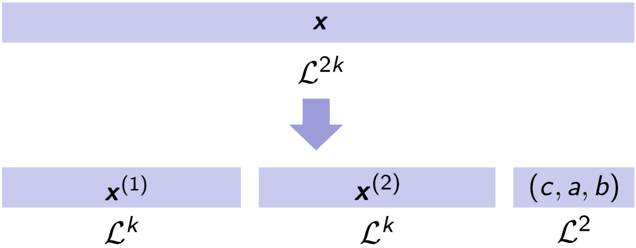
At each step, we update t by some multiplicative factor, then update $\boldsymbol{x}$ and $\boldsymbol{s}$ by solving the following system (*Newton System*):

$$\begin{pmatrix} \nabla^2\phi(\overline{\boldsymbol{x}}) & \mathbf{I}/\overline{t} & \mathbf{0} \\ \mathbf{A} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{A}^\top \end{pmatrix} \begin{pmatrix} \delta_{\boldsymbol{x}} \\ \delta_{\boldsymbol{s}} \\ \delta_{\boldsymbol{y}} \end{pmatrix} = \begin{pmatrix} \delta_\mu \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}$$

Cone Splitting

Even without recomputing from scratch at each iteration, updates for high-dimension blocks are still expensive! It takes n_i^ω time just to update one block of dimension n_i.

Instead, we can transform higher dimension cones into the intersection of smaller cones and an affine space:



where $a = \boldsymbol{x}_0^{(1)}$ and $b = \boldsymbol{x}_0^{(2)}$.

We can convert back from new SOCP to old SOCP in $O(n)$ time.

Final Algorithm for SOCP

- Cone-splitting
- Find initial feasible solution $(\boldsymbol{x}, \boldsymbol{s})$
- While $t > \epsilon$,
  - Update t to $t\left(1 - \frac{1}{\sqrt{r}}\right)$.
  - Calculate δ_x and δ_s.
  - Update $\boldsymbol{x}$ to $\boldsymbol{x} + \delta_x$ and $\boldsymbol{s}$ to $\boldsymbol{s} + \delta_s$.
  - Update $\bar{\boldsymbol{x}}$, $\bar{\boldsymbol{s}}$ as needed.
- Using the solution to the modified SOCP, reconstruct the solution to the original SOCP.

This algorithm solves a second-order cone program in $O(n^\omega + n^2 r^{1/6} + n^{2.5-\alpha/2} \log(1/\epsilon))$ time.

Acknowledgements

I would like to thank

- My mentor Guanghao Ye for his guidance and encouragement throughout this project;
- Dr. Etingof, Dr. Gerovitch, Dr. Khovanova, and all the MIT PRIMES-USA organizers for making this math research opportunity possible.

References

Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *J. ACM*, 68(1):3:1–3:39, 2021.

Yuzhou Gu, Zhao Song, and Lichen Zhang. A nearly-linear time algorithm for struc- tured support vector machines. *arXiv preprint arXiv:2307.07735*, 2023.

A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, dec 1984.

Yin Tat Lee, Zhao Song, and Qiuyi Zhang. Solving empirical risk minimization in the current matrix multiplication time. In Alina Beygelzimer and Daniel Hsu, editors, *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pages 2140–2157. PMLR, 2019.

Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming.* Society for Industrial and Applied Mathematics, 1994.