

CME 307 / MS&E 311 / OIT 676: Optimization

## Conic optimization

Professor Udell

Management Science and Engineering  
Stanford

November 18, 2024

# Outline

Semidefinite programming

Conic optimization

Conic form

## Semidefinite program

A **semidefinite program** (SDP) is written as

$$\begin{array}{ll} \text{minimize} & \langle C, X \rangle \\ \text{subject to} & \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m \\ & X \succeq 0 \\ \text{variable} & X \in \mathbf{S}^n \end{array}$$

where

- ▶  $C, A_i \in \mathbf{S}^n$ : symmetric matrices
- ▶  $\langle A, B \rangle = \text{tr}(A^T B) = \sum_{ij} A_{ij} B_{ij}$ : matrix inner product (linear in  $A$  and in  $B$ )

## Semidefinite program: applications

SDPs arise in various fields:

- ▶ **Control theory**: stability analysis via Lyapunov functions
- ▶ **Combinatorial optimization**: relaxations of NP-hard problems
- ▶ **Eigenvalue optimization**: maximizing or minimizing eigenvalues

## Semidefinite program: applications

SDPs arise in various fields:

- ▶ **Control theory**: stability analysis via Lyapunov functions
- ▶ **Combinatorial optimization**: relaxations of NP-hard problems
- ▶ **Eigenvalue optimization**: maximizing or minimizing eigenvalues

Advantages of SDPs:

- ▶ convex optimization: globally optimal solutions
- ▶ generalizes linear programming (LP)
- ▶ efficient algorithms (e.g., interior-point methods, first-order methods)

## Example: MaxCut

Given a graph  $G = (V, E)$  with edge weights  $w_{ij}$ , the **MaxCut** problem seeks to

- ▶ partition  $V$  into two disjoint sets  $S$  and  $V \setminus S$
- ▶ maximize the total weight of edges crossing the cut

## Example: MaxCut

Given a graph  $G = (V, E)$  with edge weights  $w_{ij}$ , the **MaxCut** problem seeks to

- ▶ partition  $V$  into two disjoint sets  $S$  and  $V \setminus S$
- ▶ maximize the total weight of edges crossing the cut

formulate as an integer quadratic program:

$$\begin{array}{ll}\text{maximize} & \frac{1}{4} \sum_{i,j} w_{ij} (1 - x_i x_j) \\ \text{subject to} & x_i \in \{-1, 1\}, \quad i = 1, \dots, n\end{array}$$

where

- ▶  $x_i$  represents assignment of node  $i$  to a partition

interpretation:

- ▶  $w_{ij}$  is value of cutting edge  $(i, j)$
- ▶ objective is to maximize total cut value

## SDP relaxation of MaxCut

Relax integer constraints by allowing  $x_i$  to be unit vectors  $v_i \in \mathbf{R}^n$ :

$$\begin{array}{ll}\text{maximize} & \frac{1}{4} \sum_{i,j} w_{ij} (1 - v_i^T v_j) \\ \text{subject to} & \|v_i\| = 1, \quad i = 1, \dots, n \\ \text{variable} & v_i \in \mathbf{R}^n\end{array}$$

## SDP relaxation of MaxCut

Relax integer constraints by allowing  $x_i$  to be unit vectors  $v_i \in \mathbf{R}^n$ :

$$\begin{array}{ll}\text{maximize} & \frac{1}{4} \sum_{i,j} w_{ij} (1 - v_i^T v_j) \\ \text{subject to} & \|v_i\| = 1, \quad i = 1, \dots, n \\ \text{variable} & v_i \in \mathbf{R}^n\end{array}$$

Equivalent SDP formulation, defining  $X_{ij} = v_i^T v_j$ :

$$\begin{array}{ll}\text{maximize} & \frac{1}{4} \sum_{i,j} w_{ij} (1 - X_{ij}) \\ \text{subject to} & X_{ii} = 1, \quad i = 1, \dots, n \\ & X \succeq 0 \\ \text{variable} & X \in \mathbf{S}^n\end{array}$$

## When is the relaxation tight?

The SDP relaxation is **tight** when  $X^*$  is rank one:  $X^* = x^*(x^*)^T$

- ▶  $(x^*(x^*)^T)_{ii} = 1 \implies x_i^* \in \{-1, 1\}$ , recovering integer solution

## When is the relaxation tight?

The SDP relaxation is **tight** when  $X^*$  is rank one:  $X^* = x^*(x^*)^T$

- ▶  $(x^*(x^*)^T)_{ii} = 1 \implies x_i^* \in \{-1, 1\}$ , recovering integer solution

in general:

- ▶ SDP provides an upper bound on MaxCut value
- ▶ Goemans-Williamson algorithm (1995) uses randomized rounding to obtain integer solution with approximation ratio of 0.878
- ▶ this approximation ratio is optimal assuming
  - ▶ the Unique Games conjecture and
  - ▶  $P \neq NP$

## When is the relaxation tight?

The SDP relaxation is **tight** when  $X^*$  is rank one:  $X^* = x^*(x^*)^T$

- ▶  $(x^*(x^*)^T)_{ii} = 1 \implies x_i^* \in \{-1, 1\}$ , recovering integer solution

in general:

- ▶ SDP provides an upper bound on MaxCut value
- ▶ Goemans-Williamson algorithm (1995) uses randomized rounding to obtain integer solution with approximation ratio of 0.878
- ▶ this approximation ratio is optimal assuming
  - ▶ the Unique Games conjecture and
  - ▶  $P \neq NP$

For details, see <https://math.mit.edu/~goemans/PAPERS/maxcut-jacm.pdf>

# Outline

Semidefinite programming

Conic optimization

Conic form

## Convex cone

### Definition (Convex cone)

A convex set  $K \subseteq \mathbf{R}^n$  is a **cone** if for all  $x \in K$  and  $\alpha \geq 0$ , we have  $\alpha x \in K$ .

## Convex cone

### Definition (Convex cone)

A convex set  $K \subseteq \mathbf{R}^n$  is a **cone** if for all  $x \in K$  and  $\alpha \geq 0$ , we have  $\alpha x \in K$ .

examples of convex cones:

- ▶ the zero cone  $\{0\}$
- ▶ the nonnegative orthant  $\mathbf{R}_+^n = \{x \in \mathbf{R}^n \mid x \geq 0\}$
- ▶ the second-order cone  $\{(x, t) \in \mathbf{R}^n \times \mathbf{R} \mid \|x\| \leq t\}$
- ▶ the positive semidefinite cone  $\mathbf{S}_+^n = \{X \in \mathbf{S}^n \mid X \succeq 0\}$
- ▶ the exponential cone  $\{(x, y, z) \in \mathbf{R}^3 \mid ye^{x/y} \leq z, y > 0\}$
- ▶ sums of cones  $K_1 + K_2 = \{x_1 + x_2 \mid x_1 \in K_1, x_2 \in K_2\}$

## Conic optimization

### Definition (Conic optimization)

A **conic optimization problem** is a convex optimization problem of the form

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & Ax + b \in K \\ \text{variable} & x \in \mathbf{R}^n\end{array}$$

where  $K$  is a convex cone.

## Conic optimization

### Definition (Conic optimization)

A **conic optimization problem** is a convex optimization problem of the form

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & Ax + b \in K \\ \text{variable} & x \in \mathbf{R}^n\end{array}$$

where  $K$  is a convex cone.

- ▶ generalizes linear programming ( $K = \mathbf{R}_+^m$ )
- ▶ structured representation of constraints: no oracles needed!
- ▶ can be solved efficiently for many cones

## Conic duality

for cone  $K$ , replace

$$b - Ax \geq 0 \quad \text{with} \quad b - Ax \in K$$

define **slack vector**  $s = b - Ax \in K$

for weak duality, dual  $y$  must satisfy

$$\langle y, s \rangle \geq 0 \quad \forall s \in K$$

## Dual cones

this inequality defines the **dual cone**  $K^*$ :

### Definition (dual cone)

the dual cone  $K^*$  of a cone  $K$  is the set of vectors  $y$  such that

$$\langle y, s \rangle \geq 0 \quad \forall s \in K$$

## Dual cones

this inequality defines the **dual cone**  $K^*$ :

### Definition (dual cone)

the dual cone  $K^*$  of a cone  $K$  is the set of vectors  $y$  such that

$$\langle y, s \rangle \geq 0 \quad \forall s \in K$$

examples of cones and their duals:

- ▶  $K$  acute,  $K^*$  obtuse
- ▶  $K = \mathbf{R}_+^m$ ,  $K^* = \mathbf{R}_+^m$
- ▶  $K = \{x \in \mathbf{R}^{n+1} \mid \|x_{1:n}\| \leq x_{n+1}\}$ ,  $K^* = \{y \in \mathbf{R}^n \mid \|y\| \leq y_0\}$
- ▶  $K = \{X \in \mathbf{S}^n \mid X \succeq 0\}$ ,  $K^* = \{Y \in \mathbf{S}^n \mid Y \succeq 0\}$

inner product  $\langle X, Y \rangle = \text{tr}(X^T Y) = \sum_{ij} X_{ij} Y_{ij}$  for  $X, Y \in \mathbf{S}^n$

## Conic duality

primal problem with solution  $x^* \in \mathbf{R}^n$ , optimal value  $p^*$ , variable  $x \in \mathbf{R}^n$ :

$$\begin{array}{ll} \text{minimize} & \langle c, x \rangle \\ \text{subject to} & b - Ax \in K : \quad y \in K^* \end{array} \quad (\mathcal{P})$$

## Conic duality

primal problem with solution  $x^* \in \mathbf{R}^n$ , optimal value  $p^*$ , variable  $x \in \mathbf{R}^n$ :

$$\begin{array}{ll} \text{minimize} & \langle c, x \rangle \\ \text{subject to} & b - Ax \in K : y \in K^* \end{array} \quad (\mathcal{P})$$

for  $y \in K^*$ , construct Lagrangian  $\mathcal{L}(x, y) = \langle c, x \rangle - \langle y, b - Ax \rangle$ , ensure value is **better** (lower) when  $x$  and  $y$  are feasible

## Conic duality

primal problem with solution  $x^* \in \mathbf{R}^n$ , optimal value  $p^*$ , variable  $x \in \mathbf{R}^n$ :

$$\begin{array}{ll} \text{minimize} & \langle c, x \rangle \\ \text{subject to} & b - Ax \in K : y \in K^* \end{array} \quad (\mathcal{P})$$

for  $y \in K^*$ , construct Lagrangian  $\mathcal{L}(x, y) = \langle c, x \rangle - \langle y, b - Ax \rangle$ , ensure value is **better** (lower) when  $x$  and  $y$  are feasible

$$\mathcal{L}(x, y) := \langle c, x \rangle - \langle y, b - Ax \rangle$$

## Conic duality

primal problem with solution  $x^* \in \mathbf{R}^n$ , optimal value  $p^*$ , variable  $x \in \mathbf{R}^n$ :

$$\begin{array}{ll} \text{minimize} & \langle c, x \rangle \\ \text{subject to} & b - Ax \in K : y \in K^* \end{array} \quad (\mathcal{P})$$

for  $y \in K^*$ , construct Lagrangian  $\mathcal{L}(x, y) = \langle c, x \rangle - \langle y, b - Ax \rangle$ , ensure value is **better** (lower) when  $x$  and  $y$  are feasible

$$\begin{aligned} \mathcal{L}(x, y) &:= \langle c, x \rangle - \langle y, b - Ax \rangle \\ p^* &\geq \inf_{x \text{ feas}} \langle c, x \rangle - \langle y, b - Ax \rangle \\ &\geq \inf_x \langle c, x \rangle - \langle y, b - Ax \rangle \\ &= \langle y, -b \rangle + \inf_x \langle c + A^*y, x \rangle \end{aligned}$$

## Conic duality

primal problem with solution  $x^* \in \mathbf{R}^n$ , optimal value  $p^*$ , variable  $x \in \mathbf{R}^n$ :

$$\begin{array}{ll} \text{minimize} & \langle c, x \rangle \\ \text{subject to} & b - Ax \in K : y \in K^* \end{array} \quad (\mathcal{P})$$

for  $y \in K^*$ , construct Lagrangian  $\mathcal{L}(x, y) = \langle c, x \rangle - \langle y, b - Ax \rangle$ , ensure value is **better** (lower) when  $x$  and  $y$  are feasible

$$\begin{aligned} \mathcal{L}(x, y) &:= \langle c, x \rangle - \langle y, b - Ax \rangle \\ p^* &\geq \inf_{x \text{ feas}} \langle c, x \rangle - \langle y, b - Ax \rangle \\ &\geq \inf_x \langle c, x \rangle - \langle y, b - Ax \rangle \\ &= \langle y, -b \rangle + \inf_x \langle c + A^*y, x \rangle \end{aligned}$$

which is  $-\infty$  unless  $c + A^*y = 0$ , so

## Conic duality

primal problem with solution  $x^* \in \mathbf{R}^n$ , optimal value  $p^*$ , variable  $x \in \mathbf{R}^n$ :

$$\begin{array}{ll} \text{minimize} & \langle c, x \rangle \\ \text{subject to} & b - Ax \in K : y \in K^* \end{array} \quad (\mathcal{P})$$

for  $y \in K^*$ , construct Lagrangian  $\mathcal{L}(x, y) = \langle c, x \rangle - \langle y, b - Ax \rangle$ , ensure value is **better** (lower) when  $x$  and  $y$  are feasible

$$\begin{aligned} \mathcal{L}(x, y) &:= \langle c, x \rangle - \langle y, b - Ax \rangle \\ p^* &\geq \inf_{x \text{ feas}} \langle c, x \rangle - \langle y, b - Ax \rangle \\ &\geq \inf_x \langle c, x \rangle - \langle y, b - Ax \rangle \\ &= \langle y, -b \rangle + \inf_x \langle c + A^*y, x \rangle \end{aligned}$$

which is  $-\infty$  unless  $c + A^*y = 0$ , so define the **dual problem** with variable  $y \in K^*$ :

$$\begin{array}{ll} \text{maximize} & \langle y, -b \rangle \\ \text{subject to} & c + A^*y = 0 \end{array} \quad (\mathcal{D})$$

## Conic duality

primal problem with solution  $x^* \in \mathbf{R}^n$ , optimal value  $p^*$ , variable  $x \in \mathbf{R}^n$ :

$$\begin{array}{ll} \text{minimize} & \langle c, x \rangle \\ \text{subject to} & b - Ax \in K : y \in K^* \end{array} \quad (\mathcal{P})$$

for  $y \in K^*$ , construct Lagrangian  $\mathcal{L}(x, y) = \langle c, x \rangle - \langle y, b - Ax \rangle$ , ensure value is **better** (lower) when  $x$  and  $y$  are feasible

$$\begin{aligned} \mathcal{L}(x, y) &:= \langle c, x \rangle - \langle y, b - Ax \rangle \\ p^* &\geq \inf_{x \text{ feas}} \langle c, x \rangle - \langle y, b - Ax \rangle \\ &\geq \inf_x \langle c, x \rangle - \langle y, b - Ax \rangle \\ &= \langle y, -b \rangle + \inf_x \langle c + A^*y, x \rangle \end{aligned}$$

which is  $-\infty$  unless  $c + A^*y = 0$ , so define the **dual problem** with variable  $y \in K^*$ :

$$\begin{array}{ll} \text{maximize} & \langle y, -b \rangle \\ \text{subject to} & c + A^*y = 0 \end{array} \quad (\mathcal{D})$$

again we have weak duality  $p^* \geq d^*$  and (under constraint qual) strong duality

# Outline

Semidefinite programming

Conic optimization

Conic form

## Conic form: LP example

we can represent many functions as the solution to a conic-form problem using an epigraph transformation, by **lifting** the problem to a higher dimension:

$$\|x\|_1 =$$

## Conic form: LP example

we can represent many functions as the solution to a conic-form problem using an epigraph transformation, by **lifting** the problem to a higher dimension:

$$\begin{aligned}\|x\|_1 &= \min && \mathbf{1}^T s \\ &\text{subject to} && -s \leq x \leq s \\ & && \min && \mathbf{1}^T s \\ &= \text{subject to} && s - x \in \mathbf{R}_+^n \\ & && s + x \in \mathbf{R}_+^n\end{aligned}$$

we say that  $\|x\|_1$  is **LP-representable** since this conic representation is a linear program.

## Conic form: SOC example

many functions involving quadratics can be represented using the second-order cone:  
for example, for  $x \in \mathbf{R}^n$ ,

$$\|x\|^2 =$$

## Conic form: SOC example

many functions involving quadratics can be represented using the second-order cone:  
for example, for  $x \in \mathbf{R}^n$ ,

$$\|x\|^2 = \begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \|(2x, t-1)\|_2 \leq t+1 \iff (2x, t-1, t+1) \in \text{SOC} \end{array}$$

since

$$\begin{aligned} \|(2x, t-1)\|_2 &\leq t+1 \\ 0 &\leq (t+1)^2 - \|(2x, t-1)\|_2^2 = (t+1)^2 - 4\|x\|^2 - (t-1)^2 \\ &= 4t - 4\|x\|^2 \\ \|x\|^2 &\leq t \end{aligned}$$

we say that  $\|x\|^2$  is **SOC-representable** since this conic representation is a second-order cone program.

## Conic form: SDP example

many functions of the eigenvalues of a matrix can be represented as a semidefinite program: for example, for  $X \in \mathbf{S}_+^n$ ,

$$\lambda_{\max}(X) =$$

## Conic form: SDP example

many functions of the eigenvalues of a matrix can be represented as a semidefinite program: for example, for  $X \in \mathbf{S}_+^n$ ,

$$\lambda_{\max}(X) = \begin{array}{ll} \text{minimize} & t \\ \text{subject to} & tI - X \succeq 0 \end{array}$$

we say that  $\lambda_{\max}(X)$  is **SDP-representable** since this conic representation is a semidefinite program.

- ▶ particularly useful in controls, where we may have the constraint  $X = \sum_{i=1}^m x_i A_i$ , where  $A_i$  are known matrices

## Conic form constraints

which of the following is a convex constraint?

$$\|x\|_1 \leq 1 \quad \text{or} \quad \|x\|_1 \geq 1$$

## Conic form constraints

which of the following is a convex constraint?

$$\|x\|_1 \leq 1 \quad \text{or} \quad \|x\|_1 \geq 1$$

epigraph is preserved by conic transformation

$$\{(x, t) \mid \|x\|_1 \leq t\} = \{(x, t) \mid t \geq \mathbf{1}^T s, \quad -s \leq x \leq s\}$$

## Conic form constraints

which of the following is a convex constraint?

$$\|x\|_1 \leq 1 \quad \text{or} \quad \|x\|_1 \geq 1$$

epigraph is preserved by conic transformation

$$\{(x, t) \mid \|x\|_1 \leq t\} = \{(x, t) \mid t \geq \mathbf{1}^T s, -s \leq x \leq s\}$$

intersection of epigraph with hyperplane  $\{t \mid t \leq 1\}$  is convex:

$$\{(x, t) \mid \|x\|_1 \leq t, t \leq 1\} = \{(x, t) \mid t \leq 1, t \geq \mathbf{1}^T s, -s \leq x \leq s\}$$

so convex constraint  $\|x\|_1 \leq 1$  is also LP-representable

## Example: transforming a problem to conic form

consider the square-root Lasso problem: minimize regularized loss with  $\lambda > 0$  fixed:

$$\text{minimize} \quad \|Ax - b\|_2 + \lambda \|x\|_1$$

## Example: transforming a problem to conic form

consider the square-root Lasso problem: minimize regularized loss with  $\lambda > 0$  fixed:

$$\text{minimize } \|Ax - b\|_2 + \lambda \|x\|_1$$

**Q:** Transform this problem to conic form.

## Example: transforming a problem to conic form

consider the square-root Lasso problem: minimize regularized loss with  $\lambda > 0$  fixed:

$$\text{minimize} \quad \|Ax - b\|_2 + \lambda \|x\|_1$$

**Q:** Transform this problem to conic form.

$$\begin{array}{ll} \text{minimize} & t + 1^T s \\ \text{subject to} & -s \leq x \leq s \quad \text{LP constraints} \\ & r = Ax - b \quad \text{zero-cone constraints} \\ & \|r\|_2 \leq t \quad \text{SOC constraint} \end{array}$$

## Solving conic programs

modeling conic programs:

- ▶ CVX, cvxpy, Convex.jl reformulate problems as conic programs
- ▶ (gurobipy, pyomo, JuMP do the same for (MI)LPs)

solvers specialize in solving conic programs with different cones:

- ▶ solvers compatible with cvxpy
- ▶ Mosek and COpt: excellent commercial conic solvers
- ▶ Clarabel: open-source conic solver, with Julia and Rust implementations