Instructor: Shaddin Dughmi
Outline

1 Convex Optimization Basics

2 Common Classes

3 Interlude: Positive Semi-Definite Matrices

4 More Convex Optimization Problems
Recall: Convex Optimization Problem

A problem of minimizing a convex function (or maximizing a concave function) over a convex set.

\[
\text{minimize } f(x) \\
\text{subject to } x \in \mathcal{X}
\]

- $\mathcal{X} \subseteq \mathbb{R}^n$ is convex, and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex
- Terminology: decision variable(s), objective function, feasible set, optimal solution/value, $\epsilon$-optimal solution/value
Instances typically formulated in the following standard form

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad g_i(x) \leq 0, \quad \text{for } i \in C_1. \\
& \quad a_i^T x = b_i, \quad \text{for } i \in C_2.
\end{align*}
\]

- \( g_i \) is convex

Terminology: equality constraints, inequality constraints, active/inactive at \( x \), feasible/infeasible, unbounded
Instances typically formulated in the following standard form

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad g_i(x) \leq 0, \quad \text{for } i \in C_1. \\
& \quad a_i^\top x = b_i, \quad \text{for } i \in C_2.
\end{align*}
\]

- \(g_i\) is convex
- Terminology: equality constraints, inequality constraints, active/inactive at \(x\), feasible/infeasible, unbounded
- In principle, every convex optimization problem can be formulated in this form (possibly implicitly)
  - Recall: every convex set is the intersection of halfspaces
Instances typically formulated in the following standard form

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad g_i(x) \leq 0, \quad \text{for } i \in C_1. \\
& \quad a_i^\top x = b_i, \quad \text{for } i \in C_2.
\end{align*}
\]

- $g_i$ is convex
- Terminology: equality constraints, inequality constraints, active/inactive at $x$, feasible/infeasible, unbounded
- In principle, every convex optimization problem can be formulated in this form (possibly implicitly)
  - Recall: every convex set is the intersection of halfspaces
- When $f(x)$ is immaterial (say $f(x) = 0$), we say this is convex feasibility problem
Local and Global Optimality

Fact
For a convex optimization problem, every locally optimal feasible solution is globally optimal.
Fact
For a convex optimization problem, every locally optimal feasible solution is globally optimal.

Proof
- Let $x$ be locally optimal, and $y$ be any other feasible point.
Fact
For a convex optimization problem, every locally optimal feasible solution is globally optimal.

Proof
- Let \( x \) be locally optimal, and \( y \) be any other feasible point.
- The line segment from \( x \) to \( y \) is contained in the feasible set.
Fact
For a convex optimization problem, every locally optimal feasible solution is globally optimal.

Proof
- Let $x$ be locally optimal, and $y$ be any other feasible point.
- The line segment from $x$ to $y$ is contained in the feasible set.
- By local optimality $f(x) \leq f(\theta x + (1 - \theta)y)$ for $\theta$ sufficiently close to 1.
Local and Global Optimality

Fact
For a convex optimization problem, every locally optimal feasible solution is globally optimal.

Proof
- Let $x$ be locally optimal, and $y$ be any other feasible point.
- The line segment from $x$ to $y$ is contained in the feasible set.
- By local optimality $f(x) \leq f(\theta x + (1 - \theta)y)$ for $\theta$ sufficiently close to 1.
- Jensen’s inequality then implies that $y$ is suboptimal.

$$f(x) \leq f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$$

$$f(x) \leq f(y)$$
Typically, by problem we mean a family of instances, each of which is described either explicitly via problem parameters, or given implicitly via an oracle, or something in between.
Representation

Typically, by problem we mean a family of instances, each of which is described either explicitly via problem parameters, or given implicitly via an oracle, or something in between.

Explicit Representation

A family of linear programs of the following form

\[
\begin{align*}
\text{maximize} & \quad c^T x \\
\text{subject to} & \quad Ax \preceq b \\
& \quad x \succeq 0
\end{align*}
\]

may be described by \( c \in \mathbb{R}^n \), \( A \in \mathbb{R}^{m \times n} \), and \( b \in \mathbb{R}^m \).
Typically, by problem we mean a family of instances, each of which is described either explicitly via problem parameters, or given implicitly via an oracle, or something in between.

At their most abstract, convex optimization problems of the following form

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad x \in \mathcal{X}
\end{align*}
\]

are described via a separation oracle for \( \mathcal{X} \) and \( \text{epi } f \).
Typically, by problem we mean a family of instances, each of which is described either explicitly via problem parameters, or given implicitly via an oracle, or something in between.

At their most abstract, convex optimization problems of the following form

\[
\begin{aligned}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad x \in \mathcal{X}
\end{aligned}
\]

are described via a separation oracle for \(\mathcal{X}\) and \(\text{epi } f\).

Given additional data about instances of the problem, namely a range \([L, H]\) for its optimal value and a ball of volume \(V\) containing \(\mathcal{X}\), the ellipsoid method returns an \(\epsilon\)-optimal solution using only \(\text{poly}(n, \log\left(\frac{H-L}{\epsilon}\right), \log V)\) oracle calls.
Typically, by **problem** we mean a family of **instances**, each of which is described either explicitly via **problem parameters**, or given implicitly via an **oracle**, or something in between.

**In Between**

Consider the following **fractional relaxation** of the Traveling Salesman Problem, described by a network \((V, E)\) and distances \(d_e\) on \(e \in E\).

\[
\begin{align*}
\min \sum_e d_e x_e \\
\text{s.t.} \\
\sum_{e \in \delta(S)} x_e & \geq 2, \quad \forall S \subset V, S \neq \emptyset. \\
x & \geq 0
\end{align*}
\]
Typically, by **problem** we mean a family of **instances**, each of which is described either explicitly via **problem parameters**, or given implicitly via an **oracle**, or something in between.

Consider the following **fractional relaxation** of the Traveling Salesman Problem, described by a network \((V, E)\) and distances \(d_e\) on \(e \in E\).

\[
\begin{aligned}
\min & \sum_e d_e x_e \\
\text{s.t.} & \sum_{e \in \delta(S)} x_e \geq 2, \quad \forall S \subset V, S \neq \emptyset. \\
& x \succeq 0
\end{aligned}
\]

**Representation of LP** is implicit, in the form of a network. Using this representation, separation oracles can be implemented efficiently, and used as subroutines in the ellipsoid method.
Next up: we look at some common classes of convex optimization problems

Technically, not all of them will be convex in their natural representation

However, we will show that they are “equivalent” to a convex optimization problem.
Next up: we look at some common classes of convex optimization problems
Technically, not all of them will be convex in their natural representation
However, we will show that they are “equivalent” to a convex optimization problem

Equivalence

Loosely speaking, two optimization problems are equivalent if an optimal solution to one can easily be “translated” into an optimal solution for the other.
Next up: we look at some common classes of convex optimization problems.

Technically, not all of them will be convex in their natural representation.

However, we will show that they are “equivalent” to a convex optimization problem.

**Equivalence**

Loosely speaking, two optimization problems are equivalent if an optimal solution to one can easily be “translated” into an optimal solution for the other.

**Note**

Deciding whether an optimization problem is equivalent to a tractable convex optimization problem is, in general, a black art honed by experience. There is no silver bullet.
Outline

1. Convex Optimization Basics
2. Common Classes
3. Interlude: Positive Semi-Definite Matrices
4. More Convex Optimization Problems
We have already seen linear programming

\[ \text{minimize} \quad c^T x \]
\[ \text{subject to} \quad Ax \leq b \]

\[ \mathcal{P} \]
\[ x^* \]
Linear Fractional Programming

Generalizes linear programming

\[
\begin{align*}
\text{minimize} & \quad \frac{c^T x + d}{e^T x + f} \\
\text{subject to} & \quad Ax \leq b \\
& \quad e^T x + f \geq 0
\end{align*}
\]

- The objective is quasiconvex (in fact, quasilinear) over the halfspace where the denominator is nonnegative.
Linear Fractional Programming

Generalizes linear programming

\[
\begin{align*}
\text{minimize} & \quad \frac{c^\top x + d}{e^\top x + f} \\
\text{subject to} & \quad Ax \leq b \\
& \quad e^\top x + f \geq 0
\end{align*}
\]

- The objective is quasiconvex (in fact, quasilinear) over the halfspace where the denominator is nonnegative.
- Can be reformulated as an equivalent linear program
  - Change variables to \( y = \frac{x}{e^\top x + f} \) and \( z = \frac{1}{e^\top x + f} \)

\[
\begin{align*}
\text{minimize} & \quad c^\top y + dz \\
\text{subject to} & \quad Ay \leq bz \\
& \quad z \geq 0 \\
& \quad y = \frac{x}{e^\top x + f} \\
& \quad z = \frac{1}{e^\top x + f}
\end{align*}
\]
Linear Fractional Programming

Generalizes linear programming

\[
\begin{align*}
\text{minimize} \quad & \frac{c^T x + d}{e^T x + f} \\
\text{subject to} \quad & Ax \leq b \\
& e^T x + f \geq 0
\end{align*}
\]

- The objective is quasiconvex (in fact, quasilinear) over the halfspace where the denominator is nonnegative.
- Can be reformulated as an equivalent linear program
  1. Change variables to \( y = \frac{x}{e^T x + f} \) and \( z = \frac{1}{e^T x + f} \)
  2. \((y, z)\) is a solution to the above iff \( e^T y + f z = 1 \)

\[
\begin{align*}
\text{minimize} \quad & c^T y + d z \\
\text{subject to} \quad & Ay \leq b z \\
& z \geq 0 \\
& y = \frac{x}{e^T x + f} \\
& z = \frac{1}{e^T x + f} \\
& e^T y + f z = 1
\end{align*}
\]
Example: Optimal Production Variant

- $n$ products, $m$ raw materials
- Every unit of product $j$ uses $a_{ij}$ units of raw material $i$
- There are $b_i$ units of material $i$ available
- Product $j$ yields profit $c_j$ dollars per unit, and requires an investment of $e_j$ dollars per unit to produce, with $f$ as a fixed cost
- Facility wants to maximize “Return rate on investment”

$$\text{maximize } \frac{c^T x}{e^T x + f}$$

$$\text{subject to } \begin{align*}
a_i^T x &\leq b_i, \quad \text{for } i = 1, \ldots, m. \\
x_j &\geq 0, \quad \text{for } j = 1, \ldots, n.
\end{align*}$$
Geometric Programming

Definition

- A **monomial** is a function \( f : \mathbb{R}^n_+ \to \mathbb{R}_+ \) of the form

\[
f(x) = cx_1^{a_1}x_2^{a_2} \ldots x_n^{a_n},
\]

where \( c \geq 0, a_i \in \mathbb{R} \).

- A **posynomial** is a sum of monomials.

Interpretation

GP model volume/area minimization problems, subject to constraints.
Geometric Programming

**Definition**

- A **monomial** is a function $f : \mathbb{R}_+^n \rightarrow \mathbb{R}_+$ of the form
  $$f(x) = cx_1^{a_1}x_2^{a_2} \ldots x_n^{a_n},$$
  where $c \geq 0$, $a_i \in \mathbb{R}$.

- A **posynomial** is a sum of monomials.

A **Geometric Program** is an optimization problem of the following form

minimize $f_0(x)$

subject to $f_i(x) \leq b_i$, for $i \in C_1$.

$h_i(x) = b_i$, for $i \in C_2$.

$x \succeq 0$

where $f_i$’s are posynomials, $h_i$’s are monomials, and $b_i > 0$ (wlog 1).
Geometric Programming

Definition

- A **monomial** is a function \( f : \mathbb{R}_+^n \rightarrow \mathbb{R}_+ \) of the form
  \[
  f(x) = cx_1^{a_1}x_2^{a_2} \cdots x_n^{a_n},
  \]
  where \( c \geq 0 \), \( a_i \in \mathbb{R} \).
- A **posynomial** is a sum of monomials.

A **Geometric Program** is an optimization problem of the following form

minimize \( f_0(x) \)

subject to

- \( f_i(x) \leq b_i \), for \( i \in C_1 \).
- \( h_i(x) = b_i \), for \( i \in C_2 \).
- \( x \succeq 0 \)

where \( f_i \)’s are posynomials, \( h_i \)’s are monomials, and \( b_i > 0 \) (wlog 1).

Interpretation

GP model volume/area minimization problems, subject to constraints.
Example: Designing a Suitcase

- A suitcase manufacturer is designing a suitcase
- Variables: $h, w, d$
- Want to minimize surface area $2(hw + hd + wd)$ (i.e. amount of material used)
- Have a target volume $hwd \geq 5$
- Practical/aesthetic constraints limit aspect ratio: $h/w \leq 2, h/d \leq 3$
- Constrained by airline to $h + w + d \leq 7$

minimize $2hw + 2hd + 2wd$

subject to $h^{-1}w^{-1}d^{-1} \leq \frac{1}{5}$
$h^{-1}w^{-1} \leq 2$
$hd^{-1} \leq 3$
$h + w + d \leq 7$
$h, w, d \geq 0$
Example: Designing a Suitcase

- A suitcase manufacturer is designing a suitcase
- Variables: $h, w, d$
- Want to minimize surface area $2(hw + hd + wd)$ (i.e. amount of material used)
- Have a target volume $hwd \geq 5$
- Practical/aesthetic constraints limit aspect ratio: $h/w \leq 2, h/d \leq 3$
- Constrained by airline to $h + w + d \leq 7$

$$\begin{align*}
\text{minimize} & \quad 2hw + 2hd + 2wd \\
\text{subject to} & \quad h^{-1}w^{-1}d^{-1} \leq \frac{1}{5} \\
& \quad hw^{-1} \leq 2 \\
& \quad hd^{-1} \leq 3 \\
& \quad h + w + d \leq 7 \\
& \quad h, w, d \geq 0
\end{align*}$$

More interesting applications involve optimal component layout in chip design.
minimize \[ 2hw + 2hd + 2wd \]
subject to \[ h^{-1}w^{-1}d^{-1} \leq \frac{1}{5} \]
\[ hw^{-1} \leq 2 \]
\[ hd^{-1} \leq 3 \]
\[ h + w + d \leq 7 \]
\[ h, w, d \geq 0 \]
Designing a Suitcase in Convex Form

minimize \[ 2hw + 2hd + 2wd \]
subject to \[ h^{-1}w^{-1}d^{-1} \leq \frac{1}{5} \]
\[ hw^{-1} \leq 2 \]
\[ hd^{-1} \leq 3 \]
\[ h + w + d \leq 7 \]
\[ h, w, d \geq 0 \]

Change of variables to \( \tilde{h} = \log h, \tilde{w} = \log w, \tilde{d} = \log d \)

minimize \[ 2e^{\tilde{h}+\tilde{w}} + 2e^{\tilde{h}+\tilde{d}} + 2e^{\tilde{w}+\tilde{d}} \]
subject to \[ e^{-\tilde{h}-\tilde{w}-\tilde{d}} \leq \frac{1}{5} \]
\[ e^{\tilde{h}+\tilde{w}} \leq 2 \]
\[ e^{\tilde{h}+\tilde{d}} \leq 3 \]
\[ e^{\tilde{h}} + e^{\tilde{w}} + e^{\tilde{d}} \leq 7 \]
Geometric Programs in Convex Form

minimize \( f_0(x) \)
subject to \( f_i(x) \leq b_i, \text{ for } i \in C_1. \)
\( h_i(x) = b_i, \text{ for } i \in C_2. \)
\( x \succeq 0 \)

where \( f_i \)'s are posynomials, \( h_i \)'s are monomials, and \( b_i > 0 \) (wlog 1).

- In their natural parametrization by \( x_1, \ldots, x_n \in \mathbb{R}^+ \), geometric programs are not convex optimization problems.
minimize $f_0(x)$
subject to $f_i(x) \leq b_i$, for $i \in C_1$.
$h_i(x) = b_i$, for $i \in C_2$.
$x \succeq 0$

where $f_i$’s are posynomials, $h_i$’s are monomials, and $b_i > 0$ (wlog 1).

- In their natural parametrization by $x_1, \ldots, x_n \in \mathbb{R}_+$, geometric programs are not convex optimization problems
- However, the feasible set and objective function are convex in the variables $y_1, \ldots, y_n \in \mathbb{R}$ where $y_i = \log x_i$
Geometric Programs in Convex Form

minimize \( f_0(x) \)
subject to \( f_i(x) \leq b_i, \) for \( i \in C_1. \)
\( h_i(x) = b_i, \) for \( i \in C_2. \)
\( x \preceq 0 \)

where \( f_i \)'s are posynomials, \( h_i \)'s are monomials, and \( b_i > 0 \) (wlog 1).

- Each monomial \( cx_1^{a_1}x_2^{a_2} \ldots x_k^{a_k} \) can be rewritten as a convex function \( ce^{a_1y_1 + a_2y_2 + \ldots + a_ky_k} \)
- Therefore, each posynomial becomes the sum of these convex exponential functions
- Inequality constraints and objective become convex
- Equality constraint \( cx_1^{a_1}x_2^{a_2} \ldots x_k^{a_k} = b \) reduces to an affine constraint \( a_1y_1 + a_2y_2 \ldots a_ky_k = \log \frac{b}{c} \)
Outline

1. Convex Optimization Basics
2. Common Classes
3. Interlude: Positive Semi-Definite Matrices
4. More Convex Optimization Problems
Symmetric Matrices

A matrix $A \in \mathbb{R}^{n \times n}$ is symmetric if and only if it is square and $A_{ij} = A_{ji}$ for all $i, j$.

- We denote the cone of $n \times n$ symmetric matrices by $S^n$. 

Interlude: Positive Semi-Definite Matrices
A matrix $A \in \mathbb{R}^{n \times n}$ is symmetric if and only if it is square and $A_{ij} = A_{ji}$ for all $i, j$.

- We denote the cone of $n \times n$ symmetric matrices by $S^n$.

Fact

A matrix $A \in \mathbb{R}^{n \times n}$ is symmetric if and only if it is orthogonally diagonalizable.
Symmetric Matrices

A matrix $A \in \mathbb{R}^{n \times n}$ is symmetric if and only if it is square and $A_{ij} = A_{ji}$ for all $i, j$.

- We denote the cone of $n \times n$ symmetric matrices by $S^n$.

Fact

A matrix $A \in \mathbb{R}^{n \times n}$ is symmetric if and only if it is orthogonally diagonalizable.

- i.e. $A = QDQ^\top$ where $Q$ is an orthogonal matrix and $D = \text{diag}(\lambda_1, \ldots, \lambda_n)$.
- The columns of $Q$ are the (normalized) eigenvectors of $A$, with corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$
- Equivalently: As a linear operator, $A$ scales the space along an orthonormal basis $Q$
- The scaling factor $\lambda_i$ along direction $q_i$ may be negative, positive, or 0.
A matrix $A \in \mathbb{R}^{n \times n}$ is positive semi-definite if it is symmetric and moreover all its eigenvalues are nonnegative.

- We denote the cone of $n \times n$ positive semi-definite matrices by $S^n_+$
- We use $A \succeq 0$ as shorthand for $A \in S^n_+$
A matrix $A \in \mathbb{R}^{n \times n}$ is positive semi-definite if it is symmetric and moreover all its eigenvalues are nonnegative.

- We denote the cone of $n \times n$ positive semi-definite matrices by $S_n^+$.
- We use $A \succeq 0$ as shorthand for $A \in S_n^+$.

\[ A = QDQ^\top \] where $Q$ is an orthogonal matrix and $D = \text{diag}(\lambda_1, \ldots, \lambda_n)$, where $\lambda_i \geq 0$.

As a linear operator, $A$ performs nonnegative scaling along an orthonormal basis $Q$.
A matrix $A \in \mathbb{R}^{n \times n}$ is positive semi-definite if it is symmetric and moreover all its eigenvalues are nonnegative.

- We denote the cone of $n \times n$ positive semi-definite matrices by $S^n_+$.
- We use $A \succeq 0$ as shorthand for $A \in S^n_+$.

- $A = QDQ^\top$ where $Q$ is an orthogonal matrix and $D = \text{diag}(\lambda_1, \ldots, \lambda_n)$, where $\lambda_i \geq 0$.
- As a linear operator, $A$ performs nonnegative scaling along an orthonormal basis $Q$.

Note

Positive definite, negative semi-definite, and negative definite defined similarly.
For $A \succeq 0$, let $q_1, \ldots, q_n$ be the orthonormal eigenbasis for $A$, and let $\lambda_1, \ldots, \lambda_n \geq 0$ be the corresponding eigenvalues.

The linear operator $x \rightarrow Ax$ scales the $q_i$ component of $x$ by $\lambda_i$.

When applied to every $x$ in the unit ball, the image of $A$ is an ellipsoid with principal directions $q_1, \ldots, q_n$ and corresponding diameters $2\lambda_1, \ldots, 2\lambda_n$.

When $A$ is positive definite (i.e., $\lambda_i > 0$), and therefore invertible, the ellipsoid is the set $\{x : x^T A^{-1} x \leq 1\}$.
Useful Properties of PSD Matrices

If $A \succeq 0$, then

- $x^T Ax \geq 0$ for all $x$
- $A$ has a positive semi-definite square root $A^{\frac{1}{2}}$
- $A^{\frac{1}{2}} = Q \text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n}) Q^T$
- $A = BB^T$ for some matrix $B$.
  - Interpretation: PSD matrices encode the “pairwise similarity” relationships of a family of vectors
  - Interpretation: The quadratic form $x^T Ax$ is the length of an affine transformation of $x$, namely $\|Bx\|_2^2$

- The quadratic function $x^T Ax$ is convex
- $A$ can be expressed as a sum of vector outer-products $(xx^T)$
  - E.g., sum of outer-products of columns of $B$ with $A = BB^T$
Useful Properties of PSD Matrices

If $A \succeq 0$, then

- $x^T A x \geq 0$ for all $x$
- $A$ has a positive semi-definite square root $A^{\frac{1}{2}}$
  - $A^{\frac{1}{2}} = Q \text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_n}) Q^T$
- $A = BB^T$ for some matrix $B$.
  - Interpretation: PSD matrices encode the “pairwise similarity” relationships of a family of vectors
  - Interpretation: The quadratic form $x^T A x$ is the length of an affine transformation of $x$, namely $\|Bx\|_2^2$
- The quadratic function $x^T A x$ is convex
- $A$ can be expressed as a sum of vector outer-products $(xx^T)$
  - E.g., sum of outerproducts of columns of $B$ with $A = BB^T$

As it turns out, each of the above is also sufficient for $A \succeq 0$ (assuming $A$ is symmetric).
Outline

1. Convex Optimization Basics
2. Common Classes
3. Interlude: Positive Semi-Definite Matrices
4. More Convex Optimization Problems
Quadratic Programming

Minimizing a convex quadratic function over a polyhedron.

\[
\begin{align*}
\text{minimize} \quad & x^\top P x + c^\top x + d \\
\text{subject to} \quad & A x \leq b
\end{align*}
\]

- \( P \succeq 0 \)
- Objective can be rewritten as \((x - x_0)^\top P(x - x_0)\) for some center \(x_0\)
- Sublevel sets are scaled copies of an ellipsoid centered at \(x_0\)
Examples

Constrained Least Squares

Given a set of measurements \((a_1, b_1), \ldots, (a_m, b_m)\), where \(a_i \in \mathbb{R}^n\) is the \(i\)'th input and \(b_i \in \mathbb{R}\) is the \(i\)'th output, fit a linear function minimizing mean square error, subject to known bounds on the linear coefficients.

\[
\text{minimize} \quad \|Ax - b\|_2^2 = x^\top A^\top A x - 2 b^\top A x + b^\top b
\]

subject to \(l_i \leq x_i \leq u_i\), for \(i = 1, \ldots, n\).
Examples

Distance Between Polyhedra

Given two polyhedra $Ax \leq b$ and $Cx \leq d$, find the distance between them.

\begin{align*}
\text{minimize} & \quad ||z||^2_2 = z^\top I z \\
\text{subject to} & \quad z = y - x \\
& \quad Ax \leq b \\
& \quad By \leq d
\end{align*}
This is an umbrella term for problems of the following form

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax + b \in K
\end{align*}
\]

Where \( K \) is a convex cone (e.g. \( \mathbb{R}^n_+ \), positive semi-definite matrices, etc). Evidently, such optimization problems are convex.
Conic Optimization Problems

This is an umbrella term for problems of the following form

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax + b \in K
\end{align*}
\]

Where \( K \) is a convex cone (e.g. \( \mathbb{R}^n_+ \), positive semi-definite matrices, etc). Evidently, such optimization problems are convex.

As shorthand, the cone containment constraint is often written using generalized inequalities

- \( Ax + b \succeq_K 0 \)
- \( -Ax \preceq_K b \)
- \( \ldots \)
Example: Second Order Cone Programming

We will exhibit an example of a conic optimization problem with $K$ as the second order cone

$$K = \{(x, t) : \|x\|_2 \leq t\}$$
Consider the following optimization problem, where each $a_i$ is a gaussian random variable with mean $\overline{a}_i$ and covariance matrix $\Sigma_i$.

minimize $c^T x$
subject to $a_i^T x \leq b_i$ w.p. at least 0.9, for $i = 1, \ldots, m$.

- $u_i := a_i^T x$ is a univariate normal r.v. with mean $\overline{u}_i := \overline{a}_i^T x$ and
- stddev $\sigma_i := \sqrt{x^T \Sigma_i x} = \|\Sigma_i^{1/2} x\|_2$
Consider the following optimization problem, where each $a_i$ is a gaussian random variable with mean $\overline{a_i}$ and covariance matrix $\Sigma_i$.

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad a_i^T x \leq b_i \ \text{w.p. at least 0.9, for } i = 1, \ldots, m.
\end{align*}
\]

- $u_i := a_i^T x$ is a univariate normal r.v. with mean $\overline{u_i} := \overline{a_i^T x}$ and stddev $\sigma_i := \sqrt{x^T \Sigma_i x} = \left\| \Sigma_i^{1/2} x \right\|_2$.
- $u_i \leq b_i$ with probability $\phi \left( \frac{b_i - \overline{u_i}}{\sigma_i} \right)$, where $\phi$ is the CDF of the standard normal random variable.
Consider the following optimization problem, where each \( a_i \) is a gaussian random variable with mean \( \bar{a}_i \) and covariance matrix \( \Sigma_i \).

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad a_i^T x \leq b_i \text{ w.p. at least 0.9, for } i = 1, \ldots, m.
\end{align*}
\]

- \( u_i := a_i^T x \) is a univariate normal r.v. with mean \( \bar{u}_i := \bar{a}_i^T x \) and

\[
\text{stddev } \sigma_i := \sqrt{x^T \Sigma_i x} = \| \Sigma_i^{1/2} x \|_2
\]

- \( u_i \leq b_i \) with probability \( \phi\left( \frac{b_i - \bar{u}_i}{\sigma_i} \right) \), where \( \phi \) is the CDF of the standard normal random variable.

- Since we want this probability to exceed 0.9, we require that

\[
\frac{b_i - \bar{u}_i}{\sigma_i} \geq \phi^{-1}(0.9) \approx 1.3 \approx 1/0.77
\]

\[
\| \Sigma_i^{1/2} x \|_2 \leq 0.77(b_i - \bar{a}_i^T x)
\]
Semi-Definite Programming

These are conic optimization problems where the cone in question is the set of positive semi-definite matrices.

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad x_1 F_1 + x_2 F_2 + \ldots + x_n F_n + G \succeq 0
\end{align*}
\]

Where \( F_1, \ldots, F_n \) are matrices, and \( \succeq \) refers to the positive semi-definite cone \( S^n_+ \).
These are conic optimization problems where the cone in question is the set of positive semi-definite matrices.

\[
\text{minimize} \quad c^\top x \\
\text{subject to} \quad x_1 F_1 + x_2 F_2 \ldots x_n F_n + G \succeq 0
\]

Where \( F_1, \ldots, F_n \) are matrices, and \( \succeq \) refers to the positive semi-definite cone \( S^n_+ \).

**Examples**

- Fitting a distribution, say a Gaussian, to observed data. Variable is a positive semi-definite covariance matrix.
- As a relaxation to combinatorial problems that encode pairwise relationships: e.g. finding the maximum cut of a graph.
Example: Max Cut Problem

Given an undirected graph \( G = (V, E) \), find a partition of \( V \) into \((S, V \setminus S)\) maximizing number of edges with exactly one end in \( S \).

\[
\begin{align*}
\text{maximize} & \quad \sum_{(i,j) \in E} \frac{1-x_i x_j}{2} \\
\text{subject to} & \quad x_i \in \{-1, 1\} \quad \text{for } i \in V.
\end{align*}
\]
Example: Max Cut Problem

Given an undirected graph $G = (V, E)$, find a partition of $V$ into $(S, V \setminus S)$ maximizing number of edges with exactly one end in $S$.

maximize $\sum_{(i,j) \in E} \frac{1-x_i x_j}{2}$
subject to $x_i \in \{-1, 1\}$, for $i \in V$.

Vector Program relaxation

maximize $\sum_{(i,j) \in E} \frac{1-x_i \cdot x_j}{2}$
subject to $||x_i||_2 = 1$, for $i \in V$.
$x_i \in \mathbb{R}^n$, for $i \in V$. 
Example: Max Cut Problem

Given an undirected graph $G = (V, E)$, find a partition of $V$ into $(S, V \setminus S)$ maximizing number of edges with exactly one end in $S$.

\[
\begin{align*}
\text{maximize} & \quad \sum_{(i,j) \in E} \frac{1-x_ix_j}{2} \\
\text{subject to} & \quad x_i \in \{-1, 1\}, \quad \text{for } i \in V.
\end{align*}
\]

Vector Program relaxation

\[
\begin{align*}
\text{maximize} & \quad \sum_{(i,j) \in E} \frac{1-x_ix_j}{2} \\
\text{subject to} & \quad \|x_i\|_2 = 1, \quad \text{for } i \in V. \\
& \quad x_i \in \mathbb{R}^n, \quad \text{for } i \in V.
\end{align*}
\]

SDP Relaxation

\[
\begin{align*}
\text{maximize} & \quad \sum_{(i,j) \in E} \frac{1-X_{ij}}{2} \\
\text{subject to} & \quad X_{ii} = 1, \quad \text{for } i \in V. \\
& \quad X \in S^n_+
\end{align*}
\]