Convex Optimization Problems

1 Convex Optimization Problems

1.1 Standard Form

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

$$\begin{aligned} \min \ & f(\boldsymbol{x}) \\ \text{s.t. } & \boldsymbol{x} \in \mathcal{X} \end{aligned}$$

Where $\mathcal{X} \in \mathbb{R}^n$ is a convex set, and $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function. We often put these problems in standard form:

$$\begin{aligned} \min \ & f(\boldsymbol{x}) \\ \text{s.t. } & g_i(\boldsymbol{x}) \leq 0, i \in \mathcal{C}_1 \\ & \boldsymbol{a}_i^\intercal \boldsymbol{x} = b_i, i \in \mathcal{C}_2 \end{aligned}$$

Where g_i is a convex function. When there is no objective function, we call it a **convex feasibility problem** – we are just trying to find if $\mathcal{X}$ is non-empty. Recall that because every function here is convex, every locally optimal $\boldsymbol{x}$ is also a globally optimal $\boldsymbol{x}$.

1.2 Representation

The term convex optimization problem refers to a family of **instances** (i.e., with concrete numerical constraints) described by the problem parameters, given implicitly by an oracle, or something in between.

Explicit: LP standard form.

Oracle: A separation oracle for $\mathcal{X}$ and $epi(f)$ which we can query $\boldsymbol{x} \stackrel{?}{\in} \mathcal{X}$ or $\boldsymbol{r} \stackrel{?}{\in} epi(f)$. Then, given a general range for the optimal value $\boldsymbol{x}$, the oracle can solve the problem in polynomial time.

In-between: For example, graph-encoded problems like LP fractional relaxations of TSP. In this case, the LP representation uses a graph as a "backend", which in implementation uses a separation oracle.

Oftentimes, there exist convex optimization problems which do not seem convex, but can be shown to be **equivalent** to a convex optimization problem in one of the above classes.

2 Interlude: PSD Matrices

Definition 1. A **symmetric matrix** is a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ such that $\mathbf{A}_{ij} = \mathbf{A}_{ji}$ for all i, j. Note that the set of all these matrices forms a cone denoted S^n. A matrix is symmetric if and only if it is orthogonally diagonalizable; that is, if $\mathbf{A} = \mathbf{QDQ}^\intercal$ where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{D} = diag(\lambda_1, \ldots, \lambda_n)$. The columns of $\mathbf{Q}$ are the normalized eigenvectors of $\mathbf{A}$. Geometrically, $\mathbf{A}$ scales the space along the basis $\mathbf{Q}$ with scaling factors λ_i, which can be positive, negative, or zero.

Definition 2. A **positive semi-definite (PSD) matrix** is a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ where all the eigenvalues of $\mathbf{A}$ are non-negative; we denote this by $\mathbf{A} \succeq 0$. This forms a cone denoted S^n_+. Geometrically, $\mathbf{A}$ performs a non-negative scaling along the orthonormal basis $\mathbf{Q}$. A matrix is PSD if and only if:

1. $\boldsymbol{x}^\intercal \mathbf{A} \boldsymbol{x} \geq 0 \; \forall \boldsymbol{x}$ with $\boldsymbol{x}^\intercal \mathbf{A} \boldsymbol{x}$ convex.
2. $\mathbf{A}^{1/2}$ is PSD.
3. $\mathbf{A} = \mathbf{B}^\intercal \mathbf{B}$ for some matrix $\mathbf{B}$. Note that this encodes "pairwise similarity" relationships. This also shows that $\boldsymbol{x}^\intercal \mathbf{A} \boldsymbol{x} = \|\mathbf{B}\boldsymbol{x}\|_2^2$; that is, the outer product of $\boldsymbol{x}$ and $\mathbf{A}$ represents the squared length of the vector $\boldsymbol{x}$ after the linear transformation $\mathbf{B}$.
4. $\mathbf{A}$ is a sum of vector outer products.

3 Convex Optimization Problems

3.1 Linear Fractional Programming

We know that LPs are convex (because they are linear), but linear fractional programs are also convex (in fact quasiconvex/quasilinear). A linear fractional program is defined as:

$$\begin{aligned} \min \quad & \frac{\boldsymbol{c}^\intercal \boldsymbol{x} + d}{\boldsymbol{e}^\intercal \boldsymbol{x} + f} \\ \text{s.t.} \quad & \mathbf{A}\boldsymbol{x} \preceq \boldsymbol{b} \\ & \boldsymbol{e}^\intercal \boldsymbol{x} + f > 0 \end{aligned}$$

However, they are not convex in this form. Note that the sublevel sets are convex, which is a hint that we are looking for a variable change to transform this program into a convex program. In this case, we have the new variables

$$\begin{aligned} \boldsymbol{y} &= \frac{\boldsymbol{x}}{\boldsymbol{e}^\intercal \boldsymbol{x} + f} \\ \boldsymbol{z} &= \frac{1}{\boldsymbol{e}^\intercal \boldsymbol{x} + f} \end{aligned}$$

Then, we know that $(\boldsymbol{y}, \boldsymbol{z})$ is a solution if $\boldsymbol{e}^\intercal \boldsymbol{y} + f\boldsymbol{z} = 1$. Note also that we are allowed to close strict inequalities because it keeps equivalence in the limit. So we end up with the convex program:

$$\begin{aligned} \min \quad & \boldsymbol{c}^\intercal \boldsymbol{y} + d\boldsymbol{z} \\ \text{s.t.} \quad & \mathbf{A}\boldsymbol{y} \preceq \boldsymbol{b}\boldsymbol{z} \\ & \boldsymbol{e}^\intercal \boldsymbol{y} + f\boldsymbol{z} = 1 \end{aligned}$$

This is a variant of optimal production where each product requires some initial investment and we wish to maximize ROI.

3.2 Geometric Programming

Definition 3. A **monomial** is a function $f : \mathbb{R}^n_+ \to \mathbb{R}_+$ such that

$$f(\boldsymbol{x}) = c\boldsymbol{x}_1^{a_1} \ldots \boldsymbol{x}_n^{a_n}, c \geq 0, a_i \in \mathbb{R}$$

Definition 4. A **posynomial** is a sum of monomials, i.e., $\sum_i f_i(\boldsymbol{x})$ where each f_i is a monomial.

A **geometric program** is an LP representation of volume/surface area minimization problems which uses monomials and posynomials:

$$\begin{aligned} \min\ & f_0(\boldsymbol{x}) \\ \text{s.t.}\ & f_i(\boldsymbol{x}) \leq b_i, i \in \mathcal{C}_1 \\ & h_i(\boldsymbol{x}) = b_i, i \in \mathcal{C}_2 \\ & \boldsymbol{x} \succeq 0 \end{aligned}$$

Where each f_i is a posynomial, each h_i is a monomial, and all $b_i > 0$ (wlog 1). To prove equivalence to a convex program, simply change the variables to logs. Then each monomial becomes a convex exponent, each posynomial is a sum of convex exponents, and the equality constraint reduces to an affine constraint by taking a double log.

3.3 Quadratic Programming

In a **quadratic program**, we would like to minimize a convex quadratic function (containing the PSD matrix $\mathbf{P}$) over a polyhedron.

$$\begin{aligned} \min\ & \boldsymbol{x}^\mathsf{T}\mathbf{P}\boldsymbol{x} + \boldsymbol{c}^\mathsf{T}\boldsymbol{x} + d \\ \text{s.t.}\ & \mathbf{A}\boldsymbol{x} \preceq \boldsymbol{b} \end{aligned}$$

The sublevel sets of this convex quadratic unction are scaled copies of the ellipsoid generated by $\mathbf{P}$ centered at $\mathbf{0}$. Thus, in a quadratic program, we are attempting to minimize the size of an ellipsoid such that it satisfies the constraints. One example is the constrained least squares problem from statistics. To see this, note that the objective function $\|\mathbf{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \boldsymbol{x}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{A}\boldsymbol{x} - 2\boldsymbol{b}^\mathsf{T}\mathbf{A}\boldsymbol{x} + \boldsymbol{b}^\mathsf{T}\boldsymbol{b}$ is in fact a convex quadratic function when $\mathbf{A}$ is PSD. Another useful example is the distance between polyhedrons problem. Given $\mathbf{A}\boldsymbol{x} \preceq \boldsymbol{b}$ and $\mathbf{C}\boldsymbol{y} \preceq \boldsymbol{d}$, we'd like to find $\boldsymbol{z}$ such that $\|\boldsymbol{z}\|_2^2$ is minimized. So we have the quadratic program:

$$\begin{aligned} \min\ & \boldsymbol{z}^\mathsf{T}\mathbf{I}\boldsymbol{z} \\ \text{s.t.}\ & \boldsymbol{z} = \boldsymbol{y} - \boldsymbol{x} \\ & \mathbf{A}\boldsymbol{x} \preceq \boldsymbol{b} \\ & \mathbf{C}\boldsymbol{y} \preceq \boldsymbol{d} \end{aligned}$$

4 Conic Optimization Problems

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

$$\begin{aligned} \min\ & \boldsymbol{c}^\mathsf{T}\boldsymbol{x} \\ \text{s.t.}\ & \mathbf{A}\boldsymbol{x} + \boldsymbol{b} \in K \end{aligned}$$

Where K is a convex cone. All conic problems are convex.

4.1 Stochastic Programming

A **stochasic program** is a conic program where each constraint includes random variables. In particular, if each $\boldsymbol{a}_i$ is a Gaussian random variable with mean $\bar{\boldsymbol{a}}_i$ and covariance matrix $\sum_i$, we have the stochastic program:

$$\begin{aligned} \min\ & \boldsymbol{c}^\mathsf{T}\boldsymbol{x} \\ \text{s.t.}\ & \boldsymbol{a}_i^\mathsf{T}\boldsymbol{x} \leq \boldsymbol{b}_i \text{ w.p. } \geq 0.9 \end{aligned}$$

Though this problem does not look conic at first, we can show that it is in fact contained in the second-order cone $K = \{(\boldsymbol{x}, t) : \|\boldsymbol{x}\|_2 \leq t\}$. Let $\boldsymbol{u}_i$ be a univariate Gaussian random variable with mean $\bar{\boldsymbol{u}}_i := \bar{\boldsymbol{a}}_i^\intercal \boldsymbol{x}$ and standard deviation $\boldsymbol{\sigma}_i := \sqrt{\boldsymbol{x}^\intercal \sum_i \boldsymbol{x}} = \| \sum_i^{1/2} \boldsymbol{x}\|_2$. Then $\boldsymbol{u}_i \leq \boldsymbol{b}_i$ w.p. $\Phi((\boldsymbol{b}_i - \bar{\boldsymbol{u}}_i)/\boldsymbol{\sigma}_i)$, where Φ is the Gaussian cdf. Then, we require w.p. ≥ 0.9 to obtain:

$$\frac{\boldsymbol{b}_i - \bar{\boldsymbol{u}}_i}{\boldsymbol{\sigma}_i} \geq \Phi^{-1}(0.9) \approx 1.3 \approx \frac{1}{0.77}$$

Thus:

$$\| \textstyle\sum_i^{1/2} \boldsymbol{x}\|_2 \leq 0.77(\boldsymbol{b}_i - \boldsymbol{a}_i^\intercal \boldsymbol{x}) \implies (\| \textstyle\sum_i^{1/2} \boldsymbol{x}\|_2, 0.77(\boldsymbol{b}_i - \boldsymbol{a}_i^\intercal \boldsymbol{x})) \in K$$

4.2 Semi-definite Programming

A **semi-definite program** is a conic program where $K = S_+^n$. If $\mathbf{F}_i$ and $\mathbf{G}$ are matrices, then we have the form:

$$\begin{aligned} \min\ & \boldsymbol{c}^\intercal \boldsymbol{x} \\ \text{s.t.}\ & \boldsymbol{x}_1 \mathbf{F}_1 + \boldsymbol{x}_2 \mathbf{F}_2 + \cdots + \boldsymbol{x}_n \mathbf{F}_n + \mathbf{G} \succeq \mathbf{0} \end{aligned}$$

These programs often arise when fitting a distribution to observed data, or as a relaxation to combinatorial programs which encode pairwise relationships. For example, we can show that the maximum cut problem is a semi-definite program. Given $G = (V, E)$, we wish to partition V into $(S, V \setminus S)$ maximizing the number of $e \in E$ with exactly one end in each partition. The corresponding combinatorial program is:

$$\begin{aligned} \max\ & \sum_{(i,j) \in E} \frac{1 - \boldsymbol{x}_i \boldsymbol{x}_j}{2} \\ \text{s.t.}\ & \boldsymbol{x}_i \in \{-1, 1\}\ \forall i \in V \end{aligned}$$

We can relax this program to a vector program, which allows us to encode the probability of cutting an edge given its similarity to another edge:

$$\begin{aligned} \max\ & \sum_{(i,j) \in E} \frac{1 - \boldsymbol{x}_i \boldsymbol{x}_j}{2} \\ \text{s.t.}\ & \|\boldsymbol{x}_i\|_2 = 1 \\ & \boldsymbol{x}_i \in \mathbb{R}^n \end{aligned}$$

Notice that this program encodes pairwise relationships between i and j. So, we can encode this program as an semi-definite program, which we know is conic:

$$\begin{aligned} \max\ & \sum_{(i,j) \in E} \frac{1 - \mathbf{X}_{ij}}{2} \\ \text{s.t.}\ & \mathbf{X}_{ii} = 1 \\ & \boldsymbol{x}_i \in S_+^n \end{aligned}$$