

A bird's eye view of optimization

Javier Peña
MSCF program, CMU

Fall, 2023

A bird's eye view of optimization

Plan

- Introduction
- Math background: matrices & vectors, calculus, convexity
- Popular classes of optimization models:
 - linear programming
 - quadratic programming
 - integer programming
- Hands-on experience with CVXPY Python library

Slides and python notebook available at

<https://github.com/javi-pena/birdseyeview>

Recap from last time

Optimization

Optimization model

Problem of the form

$$\begin{array}{ll}\min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in \mathcal{X}\end{array}$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ and $f : \mathcal{X} \rightarrow \mathbb{R}$.

Terminology

- Decision variables: $\mathbf{x} \in \mathbb{R}^n$
- Objective function: $f(\mathbf{x})$
- Constraint set (feasible region) $\mathcal{X} \subseteq \mathbb{R}^n$.

Convex optimization

Objective function $f(\mathbf{x})$ and constraint set \mathcal{X} are convex.

Convexity

A set $C \subseteq \mathbb{R}^n$ is convex if for all $\mathbf{x}, \mathbf{y} \in C$

$$[\mathbf{x}, \mathbf{y}] := \{\lambda \mathbf{x} + (1 - \lambda) \mathbf{y} : \lambda \in [0, 1]\} \subseteq C.$$

Examples of convex sets

- Half space: $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^\top \mathbf{x} \leq b\}$ where $\mathbf{a} \in \mathbb{R}^n$, $b \in \mathbb{R}$.
- Balls: $\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{c}\|_2 \leq r\}$ where $\mathbf{c} \in \mathbb{R}^n$, $r > 0$.
- Intersections: $C_i \subseteq \mathbb{R}^n$, $i \in I$ convex then $\bigcap_{i \in I} C_i$ convex.

Suppose $C \subseteq \mathbb{R}^n$ convex. A function $f : C \rightarrow \mathbb{R}$ is *convex on C* if for all $\mathbf{x}, \mathbf{y} \in C$ and $\lambda \in [0, 1]$

$$f(\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}).$$

Examples of convex functions

- $f(\mathbf{x}) = \mathbf{c}^\top \mathbf{x}$ where $\mathbf{c} \in \mathbb{R}^n$.
- $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ where $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$.
- $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{Q} \mathbf{x} + \mathbf{c}^\top \mathbf{x}$ where $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite.

Convex sets & convex functions

Suppose $C \subseteq \mathbb{R}^n$ is a convex set and $f : C \rightarrow \mathbb{R}$.

- f is a convex function if and only if $\text{epi}(f) := \{(\mathbf{x}, t) : \mathbf{x} \in C, t \geq f(\mathbf{x})\} \subseteq \mathbb{R}^{n+1}$ is a convex set.
- If f is a convex function on a convex set C then for all $\ell \in \mathbb{R}$ the *sublevel* set $\{\mathbf{x} \in C : f(\mathbf{x}) \leq \ell\}$ is convex.

Unconstrained optimization

Fermat's rule

Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex and differentiable. Then $\bar{\mathbf{x}} \in \mathbb{R}^n$ solves

$$\min_{\mathbf{x}} f(\mathbf{x}).$$

if and only if $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$.

Example

Suppose $\mathbf{A} \in \mathbb{R}^{m \times n}$ is full column rank and $\mathbf{b} \in \mathbb{R}^m$. Then $\bar{\mathbf{x}} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$ solves

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2.$$

We can also solve the above least-squares problem with CVXPY.

Linear program

Problem of the form

$$\begin{array}{ll}\min_{\mathbf{x}} & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{Dx} \geq \mathbf{d}.\end{array}$$

This is convex because $\mathbf{x} \mapsto \mathbf{c}^T \mathbf{x}$ is a convex function and

$$\{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{b}, \mathbf{Dx} \geq \mathbf{d}\}$$

is a convex set.

A simple portfolio construction problem

Example

You would like to allocate \$80,000 among four mutual funds.

Capitalization	Fund 1	Fund 2	Fund 3	Fund 4
large	50%	30%	25%	60%
medium	30%	10%	40%	20%
small	20%	60%	35%	20%
exp. return	10%	15%	16%	8%

- The allocation must contain at least 35% large-cap, 30% mid-cap, and 15% small-cap.
- Find an acceptable long-only allocation with the highest expected return.

Linear programming formulation

$$\begin{array}{ll}\max_{\mathbf{x}} & \mathbf{r}^T \mathbf{x} \\ \text{s.t.} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{Dx} \geq \mathbf{d} \\ & \mathbf{x} \geq \mathbf{0}\end{array}$$

where

$$\mathbf{r} = \begin{bmatrix} 0.10 \\ 0.15 \\ 0.16 \\ 0.08 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}, \mathbf{b} = 80,$$

and

$$\mathbf{D} = \begin{bmatrix} 0.50 & 0.30 & 0.25 & 0.60 \\ 0.30 & 0.10 & 0.40 & 0.20 \\ 0.20 & 0.60 & 0.35 & 0.20 \end{bmatrix}, \mathbf{d} = \begin{bmatrix} 0.35 \cdot 80 \\ 0.30 \cdot 80 \\ 0.15 \cdot 80 \end{bmatrix}.$$

Solution to optimization problems

- Optimality conditions (Fermat's rule and more general KKT)
- Numerical methods

Optimization software

- Excel Solver
- CVXPY Python library for convex optimization
- Other solvers
 - commercial: GUROBI, MOSEK, CPLEX
 - open-source: CVXOPT, ECOS, OSQP, SCS

Unsolvable linear programs

A linear program always has a solution unless one of two pathologies occur: *infeasibility* or *unboundedness*.

Other linear programs: transportation problem

Ship some commodity from sources to destinations.

s_i = supply in source $i = 1, \dots, m$

d_j = demand in destination $j = 1, \dots, n$

c_{ij} = per unit shipping cost from source i to destination j .

Linear programming formulation (assuming $\sum_{i=1}^m s_i = \sum_{j=1}^n d_j$)

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^n x_{ij} = s_i, \quad i = 1, \dots, m \\ & \sum_{i=1}^m x_{ij} = d_j, \quad j = 1, \dots, n \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

There are variants of the above, e.g., when $\sum_{i=1}^m s_i \neq \sum_{j=1}^n d_j$.

Other linear programs: ℓ_1 minimization

Suppose $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $n \gg m$ and want the sparsest solution to

$$\mathbf{Ax} = \mathbf{b}.$$

ℓ_1 norm

For $\mathbf{x} \in \mathbb{R}^n$ define $\|\mathbf{x}\|_1 := \sum_{j=1}^n |x_j|$.

ℓ_1 minimization

Used in compressed sensing and related to lasso regression:

$$\begin{array}{ll} \min_{\mathbf{x}} & \|\mathbf{x}\|_1 \\ \text{s.t.} & \mathbf{Ax} = \mathbf{b} \end{array} \Leftrightarrow \begin{array}{ll} \min_{\mathbf{x}, \mathbf{u}} & \mathbf{1}^\top \mathbf{u} \\ \text{s.t.} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \leq \mathbf{u} \\ & -\mathbf{x} \leq \mathbf{u} \end{array}$$

Quadratic programming

Quadratic program

Problem of the form

$$\begin{array}{ll}\min_{\mathbf{x}} & \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{D}\mathbf{x} \geq \mathbf{d}.\end{array}$$

This is convex if \mathbf{Q} is positive semidefinite.

Lasso regression

Suppose $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ and $\lambda > 0$. The *lasso regression* problem

$$\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_1 \right\}$$

can be reformulated as

$$\begin{array}{ll} \min_{\mathbf{x}, \mathbf{u}} & \frac{1}{2} \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - \mathbf{b}^\top \mathbf{Ax} + \lambda \cdot \mathbf{1}^\top \mathbf{u} \\ \text{s.t.} & \mathbf{x} \leq \mathbf{u} \\ & -\mathbf{x} \leq \mathbf{u} \end{array}$$

Markowitz mean-variance model

Consider an investment universe with n risky assets and a single-period investment horizon.

Let

- \mathbf{r} = vector of asset returns
- $\boldsymbol{\mu} = \mathbb{E}(\mathbf{r}) \in \mathbb{R}^n$: vector of expected returns
- $\mathbf{V} = \text{cov}(\mathbf{r}) \in \mathbb{R}^{n \times n}$: covariance matrix (symmetric and positive definite)

For a given portfolio $\mathbf{x} = [x_1 \ \cdots \ x_n]^\top \in \mathbb{R}^n$

- x_i : portfolio holding in asset i
- Expected portfolio return: $\boldsymbol{\mu}^\top \mathbf{x} = \mathbb{E}(\mathbf{r}^\top \mathbf{x})$
- Variance of portfolio return: $\mathbf{x}^\top \mathbf{V} \mathbf{x} = \text{var}(\mathbf{r}^\top \mathbf{x})$

Mean-variance models

Efficient portfolios

Optimal tradeoff between expected return $\boldsymbol{\mu}^T \mathbf{x}$ and risk $\mathbf{x}^T \mathbf{V} \mathbf{x}$.

Mean-variance model

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{x}^T \mathbf{V} \mathbf{x} \\ & \boldsymbol{\mu}^T \mathbf{x} \geq \bar{\mu} \\ & \mathbf{x} \in \mathcal{X}. \end{aligned}$$

Here \mathcal{X} : portfolio constraints.

Equivalent formulation when \mathcal{X} is closed and convex:

$$\begin{aligned} \max_{\mathbf{x}} \quad & \boldsymbol{\mu}^T \mathbf{x} - \frac{\gamma}{2} \cdot \mathbf{x}^T \mathbf{V} \mathbf{x} \\ & \mathbf{x} \in \mathcal{X}. \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \min_{\mathbf{x}} \quad & \frac{\gamma}{2} \cdot \mathbf{x}^T \mathbf{V} \mathbf{x} - \boldsymbol{\mu}^T \mathbf{x} \\ & \mathbf{x} \in \mathcal{X}. \end{aligned}$$

“Equivalent” means: set of efficient portfolios can be obtained by varying $\bar{\mu}$ or γ in each of the two formulations.

Popular simple case: fully-invested (long-only) portfolios

Consider the case when the portfolio constraint set is

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{1}^\top \mathbf{x} = 1, \mathbf{x} \geq \mathbf{0}\}.$$

The previous mean-variance model reads

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{\gamma}{2} \cdot \mathbf{x}^\top \mathbf{V} \mathbf{x} - \boldsymbol{\mu}^\top \mathbf{x} \\ & \mathbf{1}^\top \mathbf{x} = 1 \\ & \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Without the long-only constraint, get the simpler model

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{\gamma}{2} \cdot \mathbf{x}^\top \mathbf{V} \mathbf{x} - \boldsymbol{\mu}^\top \mathbf{x} \\ & \mathbf{1}^\top \mathbf{x} = 1. \end{aligned}$$

Example: efficient frontier for a one-factor model

Suppose asset returns satisfy

$$r_i = \beta_i \cdot f + u_i, \quad i = 1, \dots, n$$

where

- f = common factor that applies to all asset returns
- β_i = known exposure to common factor f
- u_i = asset-specific return

and

$$\text{cov}(u_i, f) = 0, \quad \text{cov}(u_i, u_j) = 0 \text{ for } i \neq j.$$

Some matrix algebra shows that in this case

$$\mathbf{V} = \sigma^2 \cdot \boldsymbol{\beta} \boldsymbol{\beta}^T + \mathbf{D}, \quad \boldsymbol{\mu} = \mathbb{E}(f) \cdot \boldsymbol{\beta} + \mathbb{E}(\mathbf{u})$$

where $\sigma^2 = \text{var}(f)$ and $\mathbf{D} = \text{diag}(\text{var}(u_1), \dots, \text{var}(u_n))$.

In this case efficient portfolios are a tradeoff of betas, systematic (i.e., σ^2), and idiosyncratic (i.e., $\text{var}(u_i)$) risks.

Common constraints in mean-variance models

- Upper/lower bounds on individual positions

$$\mathbf{x} \leq \mathbf{u} \text{ and/or } \mathbf{x} \geq \ell$$

- Bounds on exposure to sectors: for $S \subseteq \{1, \dots, n\}$

$$\sum_{i \in S} x_i \leq u \text{ and/or } \sum_{i \in S} x_i \geq \ell$$

- Turnover constraints: suppose \mathbf{x}^0 and \mathbf{x} are respectively a current and new portfolio. A turnover constraint is of the form

$$\sum_{i=1}^n |x_i^0 - x_i| \leq t \Leftrightarrow \begin{cases} \mathbf{x}^0 - \mathbf{x} \leq \mathbf{u} \\ \mathbf{x} - \mathbf{x}^0 \leq \mathbf{u} \\ \mathbf{1}^\top \mathbf{u} \leq t \end{cases}$$

Integer programming

Mixed integer programming

Optimization problems with integrality constraints. That is, where some variables are restricted to be integer.

$$\begin{array}{ll}\min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in \mathcal{X} \\ & x_j \in \mathbb{Z}, j \in J\end{array}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\mathcal{X} \subseteq \mathbb{R}^n$, and $J \subseteq \{1, \dots, n\}$.
We shall assume that f and \mathcal{X} are convex.

Special case: mixed binary programming

$$\begin{array}{ll}\min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in \mathcal{X} \\ & x_j \in \{0, 1\}, j \in J\end{array}$$

What is interesting about integer programming?

Powerful modeling (much more than convex optimization)

- Sometimes quantities are naturally integer
- Binary variables enable us to model logical conditions
- Binary variables enable us to model cardinality constraints, that is, “ n choose k ” constraints.

Some canonical examples

- Knapsack and set covering problems
- Scheduling problems
- Benchmark tracking
- Sparse regression

Tradeoff

Integer programs are computationally harder than convex optimization. Integer programming is NP-hard.

Knapsack problem

Select the most valuable items to pack in a knapsack with limited weight capacity.

Suppose

v_i := value of item i , $i = 1, \dots, n$

w_i := weight of item i , $i = 1, \dots, n$

W := capacity of the knapsack

Formulation

Let x_i indicate whether item i is selected

$$x_i := \begin{cases} 1 & \text{if item } i \text{ is selected} \\ 0 & \text{otherwise.} \end{cases}$$

$$\begin{aligned} \max_{\mathbf{x}} \quad & v_1x_1 + \dots + v_nx_n \\ & w_1x_1 + \dots + w_nx_n \leq W \\ & x_i \in \{0, 1\}, \quad i = 1, \dots, n. \end{aligned}$$

Set covering problem

Consider a finite “ground set” $\{1, \dots, n\}$ and a collection of sets $S_j \subseteq \{1, \dots, n\}$, $j = 1, \dots, m$ such that $\bigcup_{j=1}^m S_j = \{1, \dots, n\}$.

Suppose there is a cost c_j associated to each set S_j , $j = 1, \dots, m$.

Set covering problem

Find the cheapest collection of sets that covers the ground set:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{j=1}^m c_j x_j \\ & \sum_{j:i \in S_j} x_j \geq 1 \text{ for } i = 1, \dots, n \\ & x_j \in \{0, 1\} \text{ for } j = 1, \dots, m. \end{aligned}$$

Common generic constraint: sparsity or “ n choose k ”

For $\mathbf{x} \in \mathbb{R}^n$ let

$$\|\mathbf{x}\|_0 := |\{i : x_i \neq 0\}| = \text{number of non-zero entries in } \mathbf{x}$$

Example: sparse regression

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \\ & \|\mathbf{x}\|_0 \leq k \end{aligned}$$

Example: benchmark tracking

$$\begin{aligned} \min_{\mathbf{x}} \quad & (\mathbf{x} - \mathbf{x}_B)^\top \mathbf{V}(\mathbf{x} - \mathbf{x}_B) \\ & \mathbf{1}^\top \mathbf{x} = 1 \\ & \mathbf{x} \geq \mathbf{0} \\ & \|\mathbf{x}\|_0 \leq k \end{aligned}$$

The above problems can be recast as mixed integer programs but the resulting formulations are extremely difficult to solve.

Heuristics for “ n choose k ” constraints

Consider a problem with sparsity constraints:

$$\begin{array}{ll}\min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{x} \in \mathcal{X} \\ & \|\mathbf{x}\|_0 \leq k.\end{array}$$

Natural heuristic approach: stepwise selection

- Do either “forward” or “backward” selection.
- Forward selection:
 - Choose the non-zero component i that gives the best solution among all single-component sparse \mathbf{x} .
 - Add a new non-zero component, each time selecting the one that gives the “most” improvement over the current selection.
- Backward selection:
 - Start by letting the entire set of components be non-zero.
 - Set a new component to zero, each time selecting the one that creates the “least” worsening of the current selection.

References for further reading

Books on optimization

- Boyd & Vandenberghe, “Convex Optimization”
- Nocedal & Wright, “Numerical Optimization”
- Conforti, Cornuéjols & Zambelli, “Integer Programming”

Optimization software

- <https://www.cvxpy.org>
- <http://cvxr.com/cvx/>
- <https://www.gurobi.com>
- <https://www.mosek.com>

Book for our MSCF course

Cornuéjols, Peña & Tütüncü, “Optimization Methods in Finance”