

# 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>

# *Introduction*

# Optimization

*The process of finding the best possible solution to a problem.*

## Examples

- Optimal transport
- Optimal control
- Scheduling and logistics
- Regression, support vector machines, deep learning,...
- Portfolio construction, trade execution, risk management,...

# A mathematically precise definition

## 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)  $\mathcal{X} \subseteq \mathbb{R}^n$ .

## Common format: *mathematical programming*

Problem of the form

$$\begin{array}{ll}\min_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\ & \mathbf{h}(\mathbf{x}) = \mathbf{0}\end{array}$$

for  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , and  $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ .

The above format is too general. It can model practically anything but the optimization models are very difficult to solve.

# Taxonomy of optimization problems

## Convex optimization

Objective function  $f(\mathbf{x})$  and constraint set  $\mathcal{X}$  are convex.  
This will be the focus of our discussion today.

## Mixed integer optimization

Models with integrality constraints.  
We will discuss them in our second meeting.

## Stochastic & dynamic optimization

Models involving random and sequential features.  
We will not discuss them.

*Some math background*



# Matrices and vectors

Suppose  $m, n$  are positive integers.

## Notation

$\mathbb{R}^n$  = space of  $n$ -dimensional vectors. Convention:  $\mathbf{x} \in \mathbb{R}^n$  is a *column* vector with entries:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}.$$

$\mathbb{R}^{m \times n}$  = space of  $m$  by  $n$  matrices. Convention:  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is a matrix with entries:

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}.$$

# Operations with matrices and vectors

## Matrix-matrix multiplication

Suppose  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{B} \in \mathbb{R}^{n \times p}$ . Then their product  $\mathbf{AB} \in \mathbb{R}^{m \times p}$  is the  $m$  by  $p$  matrix with  $ij$  entry equal to

$$\sum_{k=1}^n a_{ik} b_{kj}$$

for  $i = 1, \dots, m$  and  $j = 1, \dots, p$ .

## Matrix-vector multiplication

For  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{x} \in \mathbb{R}^n$

$$\mathbf{Ax} = \begin{bmatrix} a_{11}x_1 + \cdots + a_{1n}x_n \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n \end{bmatrix} \in \mathbb{R}^m.$$

# Operations with matrices and vectors

## Transpose

Suppose  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is as follows

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}.$$

The transpose  $\mathbf{A}^T \in \mathbb{R}^{n \times m}$  is

$$\mathbf{A}^T = \begin{bmatrix} a_{11} & \cdots & a_{m1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{mn} \end{bmatrix}.$$

## Exercise

Suppose  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ . Does each of the products  $\mathbf{a}\mathbf{b}$ ,  $\mathbf{a}^T\mathbf{b}$ , and  $\mathbf{a}\mathbf{b}^T$  make sense? Are they the same? If not, how are they different?

## Operations with matrices and vectors

A matrix  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  is symmetric if  $\mathbf{Q}^T = \mathbf{Q}$ .

Suppose  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  is symmetric.

- $\mathbf{Q}$  is *positive semidefinite* (psd) if  $\mathbf{x}^T \mathbf{Q} \mathbf{x} \geq 0$  for all  $\mathbf{x} \in \mathbb{R}^n$ .
- $\mathbf{Q}$  is *positive definite* (pd) if  $\mathbf{x}^T \mathbf{Q} \mathbf{x} > 0$  for all  $\mathbf{0} \neq \mathbf{x} \in \mathbb{R}^n$ .

The *quadratic form* defined by  $\mathbf{Q}$  is  $f(\mathbf{x}) := \mathbf{x}^T \mathbf{Q} \mathbf{x}$ . Observe

$$\mathbf{x}^T \mathbf{Q} \mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n q_{ij} x_i x_j = \sum_{i=1}^n q_{ii} x_i^2 + 2 \sum_{1 \leq i < j \leq n} q_{ij} x_i x_j$$

### Choleski factorization

$\mathbf{Q} \in \mathbb{R}^{n \times n}$  is symmetric and psd if and only if  $\mathbf{Q} = \mathbf{L} \mathbf{L}^T$  for some  $\mathbf{L} \in \mathbb{R}^{n \times k}$ .

### Exercise

Show that  $\mathbf{Q} = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$  is psd if and only if  $\rho^2 \leq 1$ .

# Calculus

Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . The *gradient*  $\nabla f(\mathbf{x}) \in \mathbb{R}^n$  is

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(\mathbf{x}) \\ \vdots \\ \frac{\partial f}{\partial x_n}(\mathbf{x}) \end{bmatrix},$$

and the *Hessian*  $\nabla^2 f(\mathbf{x}) \in \mathbb{R}^{n \times n}$  is

$$\nabla^2 f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(\mathbf{x}) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(\mathbf{x}) \\ \vdots & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1}(\mathbf{x}) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(\mathbf{x}) \end{bmatrix}.$$

# Calculus

## First-order Taylor's approximation

If  $\nabla f$  is continuous at  $\bar{\mathbf{x}} \in \mathbb{R}^n$  then for small  $\mathbf{p} \in \mathbb{R}^n$

$$f(\bar{\mathbf{x}} + \mathbf{p}) \approx f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T \mathbf{p}.$$

## Second-order Taylor's approximation

If both  $\nabla f$  and  $\nabla^2 f$  are continuous at  $\bar{\mathbf{x}} \in \mathbb{R}^n$  then for small  $\mathbf{p} \in \mathbb{R}^n$

$$f(\bar{\mathbf{x}} + \mathbf{p}) \approx f(\bar{\mathbf{x}}) + \nabla f(\bar{\mathbf{x}})^T \mathbf{p} + \frac{1}{2} \mathbf{p}^T \nabla^2 f(\bar{\mathbf{x}}) \mathbf{p}.$$

## Exercise

Suppose  $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x}$  where  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  symm,  $\mathbf{c} \in \mathbb{R}^n$ . Compute  $\nabla f(\mathbf{x})$  and  $\nabla^2 f(\mathbf{x})$ . Verify the above approximations.

# Calculus

## Euclidean norm

Suppose  $\mathbf{x} \in \mathbb{R}^n$ , the Euclidean norm  $\|\mathbf{x}\|_2$  is defined as

$$\|\mathbf{x}\|_2 = \sqrt{\mathbf{x}^\top \mathbf{x}} = \sqrt{x_1^2 + \cdots + x_n^2}.$$

## Exercise

Suppose  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{b} \in \mathbb{R}^m$ . Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be defined as

$$f(\mathbf{x}) = \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2.$$

Compute  $\nabla f(\mathbf{x})$  and  $\nabla^2 f(\mathbf{x})$ .

# 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  $\lambda \in [0, 1]$

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



# Convexity and differentiability

## Theorem

*Suppose  $C \subseteq \mathbb{R}^n$  is open and convex and  $f : C \rightarrow \mathbb{R}$  is continuously differentiable. Then the following are equivalent:*

- (a)  $f$  is convex on  $C$*
- (b)  $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x})$  for all  $\mathbf{x}, \mathbf{y} \in C$*
- (c)  $(\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}))^\top (\mathbf{y} - \mathbf{x}) \geq 0$  for all  $\mathbf{x}, \mathbf{y} \in C$*

## Theorem

*Suppose  $C \subseteq \mathbb{R}^n$  is open and convex and  $f : C \rightarrow \mathbb{R}$  is twice continuously differentiable. Then  $f$  is convex on  $C$  if and only if  $\nabla^2 f(\mathbf{x})$  is positive semidefinite for all  $\mathbf{x} \in C$ .*

## Examples of convex functions

- $f(\mathbf{x}) = \mathbf{c}^\top \mathbf{x}$  where  $\mathbf{c} \in \mathbb{R}^n$ .
- $f(\mathbf{x}) = \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2$  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 convex optimization*

# Unconstrained optimization

Suppose  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  and consider the problem

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

## Fermat's rule

Suppose  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is convex and differentiable. Then  $\bar{\mathbf{x}} \in \mathbb{R}^n$  solves the above problem if and only if  $\nabla f(\bar{\mathbf{x}}) = \mathbf{0}$ .

# Unconstrained optimization

Suppose  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  is symmetric and positive semidefinite and  $\mathbf{c} \in \mathbb{R}^n$ . Then  $\bar{\mathbf{x}} \in \mathbb{R}^n$  solves

$$\min_{\mathbf{x}} \left\{ \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{c}^T \mathbf{x} \right\}$$

if and only if  $\mathbf{Q}\bar{\mathbf{x}} + \mathbf{c} = \mathbf{0}$ . In particular, if  $\mathbf{Q}$  is non-singular, then the unique minimizer is  $\bar{\mathbf{x}} = -\mathbf{Q}^{-1}\mathbf{c}$ .

## Exercise

Suppose  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is full column rank and  $\mathbf{b} \in \mathbb{R}^m$ . Find the minimizers of both

$$\frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 \text{ and } \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{x}\|_2^2$$

for  $\lambda > 0$ .

# Projections

Suppose  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and consider the linear subspace  $\mathcal{L} \subseteq \mathbb{R}^m$  spanned by the columns of  $\mathbf{A}$ :

$$\mathcal{L} := \{\mathbf{Ax} : \mathbf{x} \in \mathbb{R}^n\}.$$

The *projection* mapping  $P_{\mathcal{L}} : \mathbb{R}^m \rightarrow \mathcal{L}$  is defined as

$$P_{\mathcal{L}}(\mathbf{y}) := \operatorname{argmin}_{\mathbf{v} \in \mathcal{L}} \|\mathbf{v} - \mathbf{y}\|_2 = \operatorname{argmin}_{\mathbf{v} \in \mathcal{L}} \frac{1}{2} \|\mathbf{v} - \mathbf{y}\|_2^2.$$

## Projection matrix

If  $\mathbf{A}$  is full column rank then

$$P_{\mathcal{L}}(\mathbf{y}) = \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}.$$

## Exercise (scaled projections)

Suppose  $\mathbf{Q} \in \mathbb{R}^{m \times m}$  is symmetric and positive definite. Find

$$\operatorname{argmin}_{\mathbf{v} \in \mathcal{L}} \frac{1}{2} (\mathbf{v} - \mathbf{y})^T \mathbf{Q} (\mathbf{v} - \mathbf{y})$$

## *Linear programming*

# 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 always convex.



# 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

*Variables:*

$x_i$ : amount (in \$1000s) invested in fund  $i$  for  $i = 1, \dots, 4$ .

*Objective:*

$$\max \quad 0.10x_1 + 0.15x_2 + 0.16x_3 + 0.08x_4$$

*Constraints:*

$$\begin{array}{llll} x_1 + x_2 + x_3 + x_4 & = & 80 & \text{(budget)} \\ 0.50x_1 + 0.30x_2 + 0.25x_3 + 0.60x_4 & \geq & 0.35 \cdot 80 & \text{(large-cap)} \\ 0.30x_1 + 0.10x_2 + 0.40x_3 + 0.20x_4 & \geq & 0.30 \cdot 80 & \text{(mid-cap)} \\ 0.20x_1 + 0.60x_2 + 0.35x_3 + 0.20x_4 & \geq & 0.15 \cdot 80 & \text{(small-cap)} \\ x_1, \dots, x_4 & \geq & 0 & \text{(long-only).} \end{array}$$

## Linear programming formulation (matrix form)

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

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: $\ell_1$ minimization

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

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

### $\ell_1$ norm

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

### $\ell_1$ 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  $\gamma$  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
- $\beta_i$  = 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.,  $\sigma^2$ ), 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”