Exercises

Portfolio Optimization: Theory and Application
Appendix B – Optimization Algorithms

Daniel P. Palomar (2025). *Portfolio Optimization: Theory and Application.* Cambridge University Press.

portfoliooptimizationbook.com

Exercise B.1: Euclidean norm approximation

a. Randomly generate the parameters $\boldsymbol{A} \in \mathbb{R}^{10\times 5}$ and $\boldsymbol{b} \in \mathbb{R}^{10}$.
b. Formulate a regression problem to approximate $\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{b}$ based on the ℓ_2-norm.
c. Solve it directly with the least squares closed-form solution.
d. Solve it using a modeling framework (e.g., CVX).
e. Solve it invoking a QP solver.

Exercise B.2: Manhattan norm approximation

a. Randomly generate the parameters $\boldsymbol{A} \in \mathbb{R}^{10\times 5}$ and $\boldsymbol{b} \in \mathbb{R}^{10}$.
b. Formulate a regression problem to approximate $\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{b}$ based on the ℓ_1-norm.
c. Solve it using a modeling framework (e.g., CVX).
d. Rewrite it as an LP and solve it invoking an LP solver.

Exercise B.3: Chebyshev norm approximation

a. Randomly generate the parameters $\boldsymbol{A} \in \mathbb{R}^{10\times 5}$ and $\boldsymbol{b} \in \mathbb{R}^{10}$.
b. Formulate a regression problem to approximate $\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{b}$ based on the ℓ_∞-norm.
c. Solve it using a modeling framework (e.g., CVX).
d. Rewrite it as an LP and solve it invoking an LP solver.

Exercise B.4: Solving an LP

Consider the following LP:

$$\begin{array}{ll}\underset{x_1,x_2}{\text{maximize}} & 3x_1 + x_2\\ \text{subject to} & x_1 + 2x_2 \leq 4,\\ & 4x_1 + 2x_2 \leq 12,\\ & x_1, x_2 \geq 0.\end{array}$$

a. Solve it using a modeling framework (e.g., CVX).
b. Solve it by directly invoking an LP solver.
c. Solve it by invoking a general-purpose nonlinear solver.
d. Implement the projected gradient method to solve the problem.
e. Implement the constrained Newton's method to solve the problem.
f. Implement the log-barrier interior-point method to solve the problem (use (1,1) as the initial point).
g. Compare all the solutions and the computation time.

Exercise B.5: Central path

Formulate the log-barrier problem corresponding to the LP in Exercise B.4 and plot the central path as the parameter t varies.

Exercise B.6: Phase I method

Design a phase I method to find a feasible point for the LP in Exercise B.4, which can then be used as the starting point for the barrier method.

Exercise B.7: Dual problem

Formulate the dual problem corresponding to the LP in Exercise B.4 and solve it using a solver of your choice.

Exercise B.8: KKT conditions

Write down the Karush–Kuhn–Tucker (KKT) conditions for the LP in Exercise @ref(exr:solving-LP) and discuss their role in determining the optimality of a solution.

Exercise B.9: Solving a QP

Consider the following QP:

$$\begin{array}{ll}\underset{x_1,x_2}{\text{maximize}} & x_1^2 + x_2^2\\ \text{subject to} & x_1 + x_2 = 1,\\ & x_1 \geq 0, x_2 \geq 0.\end{array}$$

a. Solve it using a modeling framework (e.g., CVX).
b. Solve it by directly invoking a QP solver.

c. Solve it by invoking a general-purpose nonlinear solver.
d. Implement the projected gradient method to solve the problem.
e. Implement the constrained Newton's method to solve the problem.
f. Implement the log-barrier interior-point method to solve the problem (use (0.5,0.5) as the initial point).
g. Compare all the solutions and the computation time.

Exercise B.10: Fractional programming

Consider the following fractional program:

$$\begin{array}{ll} \underset{\boldsymbol{w}}{\text{maximize}} & \dfrac{\boldsymbol{w}^\mathsf{T}\mathbf{1}}{\sqrt{\boldsymbol{w}^\mathsf{T}\boldsymbol{\Sigma}\boldsymbol{w}}} \\ \text{subject to} & \mathbf{1}^\mathsf{T}\boldsymbol{w} = 1, \quad \boldsymbol{w} \geq \mathbf{0}, \end{array}$$

where $\boldsymbol{\Sigma} \succ \mathbf{0}$.

a. Solve it with a general-purpose nonlinear solver.
b. Solve it via bisection.
c. Solve it via the Dinkelbach method as a sequence of SOCPs.
d. Develop a modified algorithm that solves the problem as a sequence of QPs instead.
e. Solve it via the Schaible transform method.
f. Reformulate the problem as a minimization and then solve it via the Schaible transform method.
g. Compare all the previous approaches in terms of the accuracy of the solution and the computation time.

Exercise B.11: Soft-thresholding operator

Consider the following convex optimization problem:

$$\underset{x}{\text{minimize}} \quad \tfrac{1}{2}\|\boldsymbol{a}x - \boldsymbol{b}\|_2^2 + \lambda|x|,$$

with $\lambda \geq 0$. Derive the solution and show that it can be written as

$$x = \frac{1}{\|\boldsymbol{a}\|_2^2}\mathcal{S}_\lambda\left(\boldsymbol{a}^\mathsf{T}\boldsymbol{b}\right),$$

where $\mathcal{S}_\lambda(\cdot)$ is the so-called soft-thresholding operator defined as

$$\mathcal{S}_\lambda(u) = \text{sign}(u)(|u| - \lambda)^+,$$

with $\text{sign}(\cdot)$ denoting the sign function and $(\cdot)^+ = \max(0, \cdot)$.

Exercise B.12: ℓ_2–ℓ_1-norm minimization

Consider the following ℓ_2–ℓ_1-norm minimization problem (with $\boldsymbol{A} \in \mathbb{R}^{10\times 5}$ and $\boldsymbol{b} \in \mathbb{R}^{10}$ randomly generated):

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_1.$$

a. Solve it using a modeling framework (e.g., CVX).
b. Rewrite the problem as a QP and solve it by invoking a QP solver.
c. Solve it with an ad hoc LASSO solver.

Exercise B.13: BCD for ℓ_2–ℓ_1-norm minimization

Solve the ℓ_2–ℓ_1-norm minimization problem in Exercise B.12 via BCD. Plot the convergence vs. iterations and CPU time.

Exercise B.14: MM for ℓ_2–ℓ_1-norm minimization

Solve the ℓ_2–ℓ_1-norm minimization problem in Exercise B.12 via MM and its accelerated version. Plot the convergence vs. iterations and CPU time.

Exercise B.15: SCA for ℓ_2–ℓ_1-norm minimization

Solve the ℓ_2–ℓ_1-norm minimization problem in Exercise B.12 via SCA. Plot the convergence vs. iterations and CPU time.

Exercise B.16: ADMM for ℓ_2–ℓ_1-norm minimization

Solve the ℓ_2–ℓ_1-norm minimization problem in Exercise B.12 via ADMM. Plot the convergence vs. iterations and CPU time.