"Algorithms for Big-Data Analysis"
Mid-Term Project: Portfolio Optimization

Spring 2026

Submission and Formatting Requirements

This mid-term project asks you to complete a small but coherent study on **portfolio optimization with real market data**. The project should reflect the core training of this course: data preparation, mathematical modeling, algorithm design, numerical implementation, empirical comparison, and reproducibility.

1. The written report should contain at least:
   - the investment scenario and modeling assumptions;
   - data source, data availability audit, cleaning pipeline, and time split;
   - mathematical models, variables, parameters, and constraint interpretations;
   - benchmark results from commercial solvers;
   - ADMM and PDHG derivation, implementation, and comparison;
   - backtesting and sensitivity analysis;
   - at least one failure case or limitation.
2. Python is recommended. MATLAB is also acceptable. If another language is used, the main experiments must still be reproducible.
3. The submission should include at least:
   - a PDF report;
   - complete source code;
   - a README explaining dependencies, data preparation, running commands, and random seeds;
   - result figures and tables;
   - the data download and cleaning scripts.

   Pack all of your codes named as "proj1mk-name-ID.zip" and send it to TA: pkuopt@163.com

4. Do not paste large blocks of code into the report. The report should explain the models, algorithms, experiments, and conclusions.

5. Numerical results should be presented by tables and figures rather than screenshots of terminal output.

6. If any code module receives substantial help from external sources or large language models, state this clearly at the beginning of that module.

1 Project Overview

Typical Scenario

A portfolio manager allocates capital among a universe of liquid risky assets using historical daily price data. A typical choice is a pool of 30-100 stocks or ETFs. The investor rebalances periodically, for example every 20 trading days, under long-only and fully invested constraints. The task is to estimate return and risk from cleaned data, solve one classical quadratic portfolio model and one conic portfolio model with commercial solvers, compare them in backtests, and then implement ADMM and PDHG for one core convex model from scratch.

The core project is organized around three components:

(1) A classical QP portfolio model based on mean and variance [3, 1];

(2) A conic portfolio model based on standard deviation [1];

(3) First-order methods for one common core convex model, namely ADMM and PDHG [4, 5, 2].

Optional extensions may include turnover regularization, factor-based covariance modeling, mean-CVaR portfolios, or a parallel PDHG implementation.

Figure 1 gives a classical visual illustration of the risk-return trade-off and the corresponding portfolio allocations. It is useful as a first reference point before introducing the mathematical models in the next section.

2 Problem Setup and Core Models

Let $r_1, \dots, r_T \in \mathbb{R}^n$ denote cleaned return observations for n assets, where $r_t = (r_{t,1}, \dots, r_{t,n})^\top$.

Definition 2.1. Let

- $\hat{\boldsymbol{\mu}} \in \mathbb{R}^n$ be the estimated expected return vector;
- $\hat{\Sigma} \in \mathbb{R}^{n \times n}$ be the estimated covariance matrix;
- $\boldsymbol{x} \in \mathbb{R}^n$ be the portfolio weight vector;
- $\mathbf{1}^\top \boldsymbol{x} = 1$ be the budget constraint;

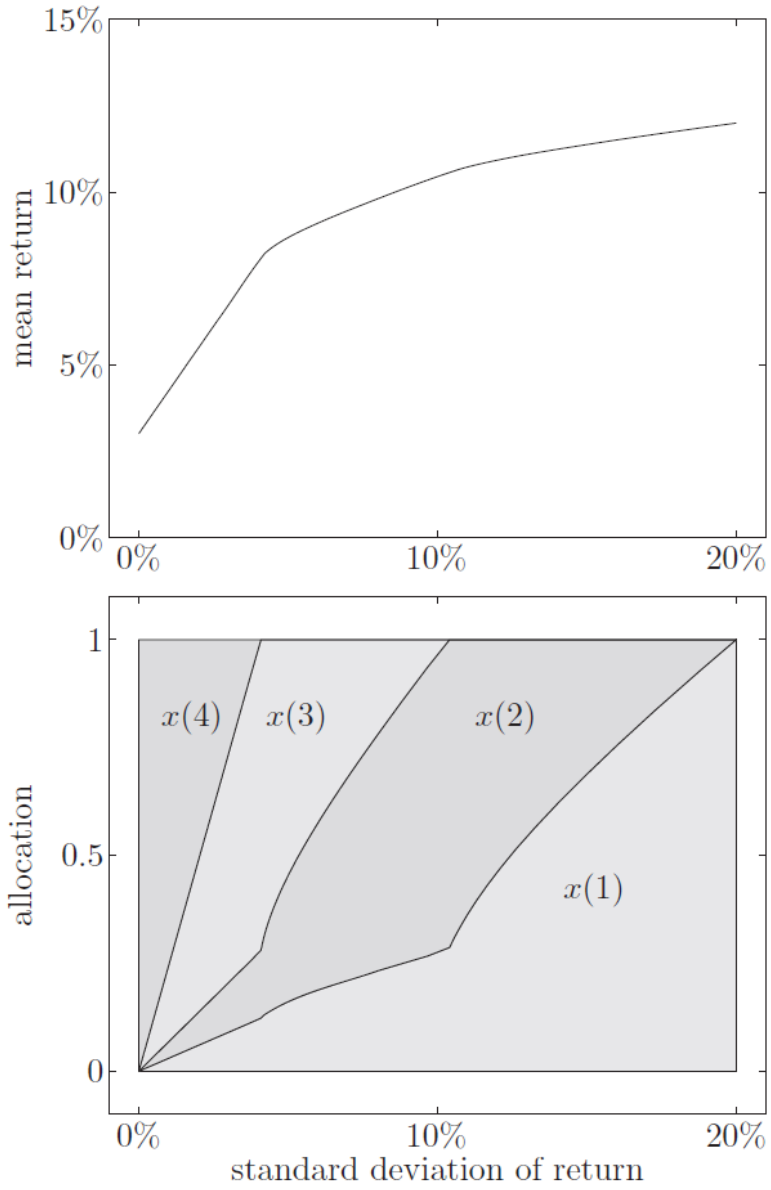


Figure 1: Illustration of the classical risk-return trade-off and the corresponding optimal portfolio allocations. Source: [1].

- $\boldsymbol{x} \geq 0$ denote a long-only portfolio.

The portfolio return and variance are

$$r(\boldsymbol{x}) = \hat{\boldsymbol{\mu}}^\top \boldsymbol{x}, \tag{2.1}$$

$$\sigma^2(\boldsymbol{x}) = \boldsymbol{x}^\top \hat{\Sigma} \boldsymbol{x}. \tag{2.2}$$

If $\hat{\Sigma} \succeq 0$ admits a factorization

$$\hat{\Sigma} = BB^\top, \tag{2.3}$$

then

$$\sqrt{\boldsymbol{x}^\top \hat{\Sigma} \boldsymbol{x}} = \|B^\top \boldsymbol{x}\|_2. \tag{2.4}$$

If $\hat{\Sigma} \succ 0$, one may take B to be the Cholesky factor [1].

2.1 Model A1: Minimum Variance with Target Return

The classical Markowitz model is

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^n} \quad & \frac{1}{2} \boldsymbol{x}^\top \hat{\Sigma} \boldsymbol{x} \\ \text{subject to} \quad & \hat{\boldsymbol{\mu}}^\top \boldsymbol{x} \geq r_{\text{target}}, \\ & \mathbf{1}^\top \boldsymbol{x} = 1, \\ & \boldsymbol{x} \geq 0. \end{aligned} \tag{2.5}$$

2.2 Model A2: Quadratic Utility Portfolio

A closely related formulation is

$$\begin{aligned} \min_{\boldsymbol{x} \in \mathbb{R}^n} \quad & \frac{\gamma}{2} \boldsymbol{x}^\top \hat{\Sigma} \boldsymbol{x} - \hat{\boldsymbol{\mu}}^\top \boldsymbol{x} \\ \text{subject to} \quad & \mathbf{1}^\top \boldsymbol{x} = 1, \\ & \boldsymbol{x} \geq 0, \end{aligned} \tag{2.6}$$

where $\gamma > 0$ is a risk-aversion parameter.

Proposition 2.1. *Problem* (2.5) *and problem* (2.6) *are convex quadratic program [1, 3].*

Remark 2.1. Problem (2.6) will be used as the **common core model** for the ADMM and PDHG tasks below. This keeps the first-order-method comparison focused and implementation-friendly.

2.3 Model B: Return Maximization under a Standard Deviation Budget

A standard conic formulation is

$$\begin{aligned} \max_{\boldsymbol{x}\in\mathbb{R}^n} \quad & \hat{\boldsymbol{\mu}}^\top \boldsymbol{x} \\ \text{subject to} \quad & \|B^\top \boldsymbol{x}\|_2 \le \sigma_{\max}, \\ & \mathbf{1}^\top \boldsymbol{x} = 1, \\ & \boldsymbol{x} \ge 0. \end{aligned} \tag{2.7}$$

Proposition 2.2. *Problem* (2.7) *is a second-order cone program [1].*

2.4 Optional Extension: Turnover Regularization

To model transaction frictions or rebalancing costs, one may add an ℓ_1 turnover penalty:

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & \frac{\gamma}{2}\boldsymbol{x}^\top \hat{\Sigma}\boldsymbol{x} - \hat{\boldsymbol{\mu}}^\top \boldsymbol{x} + \lambda\|\boldsymbol{x} - \boldsymbol{x}^{\text{prev}}\|_1 \\ \text{subject to} \quad & \mathbf{1}^\top \boldsymbol{x} = 1, \\ & \boldsymbol{x} \ge 0, \end{aligned} \tag{2.8}$$

where $\boldsymbol{x}^{\text{prev}}$ is the previous portfolio and $\lambda \ge 0$ controls turnover.

Remark 2.2. Problem (2.8) is convex but non-smooth. By introducing auxiliary variables for the absolute values, it can be implemented as a convex QP with additional linear constraints.

3 Task 1: Data access, Cleaning, and Estimation

This project must use **real market data**. Before formal modeling, perform a data availability audit.

Task 3.1 (Data availability audit)**.** For your chosen data source, report:

- whether the interface is accessible;
- whether a token, quota, or permission is required;
- whether the key fields needed for this project are available;
- whether the time span is long enough for train/validation/test splitting or rolling backtesting.

Task 3.2 (Data cleaning and return construction)**.** Choose a concrete asset universe and describe it clearly. Examples include:

- a subset of S&P 500 stocks (.csv file provided),
- a group of liquid ETFs,
- a sector-based stock universe,

- another transparent and reproducible universe.

Then complete the following steps:

(a) Align all assets to a common trading calendar.

(b) Remove assets with too many missing observations.

(c) Decide how to treat delisted assets, suspended assets, or very short histories.

(d) Use adjusted prices when appropriate.

(e) Construct returns explicitly, for example simple returns

$$r_{t,i} = \frac{p_{t+1,i} - p_{t,i}}{p_{t,i}}$$

or log returns

$$r_{t,i} = \log\left(\frac{p_{t+1,i}}{p_{t,i}}\right).$$

(f) State how missing values and outliers are handled.

Report the number of assets before and after cleaning.

Task 3.3 (Estimation of input parameters)**.** On each training window, estimate $\hat{\boldsymbol{\mu}}$ and $\hat{\Sigma}$. At minimum, write down the formulas you use. For example, with sample estimates,

$$\hat{\boldsymbol{\mu}} = \frac{1}{T}\sum_{t=1}^{T} r_t, \tag{3.1}$$

$$\hat{\Sigma} = \frac{1}{T-1}\sum_{t=1}^{T}(r_t - \hat{\boldsymbol{\mu}})(r_t - \hat{\boldsymbol{\mu}})^\top. \tag{3.2}$$

If numerical stabilization is needed, you may use

$$\hat{\Sigma}_\epsilon = \hat{\Sigma} + \epsilon I, \qquad \epsilon > 0. \tag{3.3}$$

You may also use shrinkage or factor-based estimates, but the exact method must be stated clearly.

Remark 3.1. The report should distinguish clearly between **data cleaning** and **statistical estimation**. These are not the same step.

4 Task 2: Commercial-solver Baseline

Use one of the following modeling interfaces:

- `CVX` in MATLAB, or

- `CVXPY` in Python.

Use at least one commercial solver among:

- **MOSEK**,
- **Gurobi**.

Python is recommended, so `CVXPY + MOSEK` or `CVXPY + Gurobi` is a natural choice.

Task 4.1 (QP baseline and efficient frontier)**.** Using Model A1 or Model A2:

(a) Solve the classical QP portfolio model with a commercial solver.

(b) Produce an efficient frontier, either by sweeping r_{target} in (2.5) or by sweeping γ in (2.6).

(c) Plot risk-return points for at least 20 parameter values.

(d) For several representative portfolios, report:

- expected return,
- variance or standard deviation,
- objective value,
- solver status,
- running time,
- portfolio weights.

Task 4.2 (SOCP baseline)**.** Solve Model B in (2.7) for several values of $\sigma_{\max}$.

(a) Plot the resulting portfolios in the same risk-return plane.

(b) Compare them against the QP-based frontier.

(c) Explain the modeling and computational difference between the QP and SOCP formulations.

Task 4.3 (Optional realistic modification)**.** Add at least one realistic modification to the baseline model. Examples include:

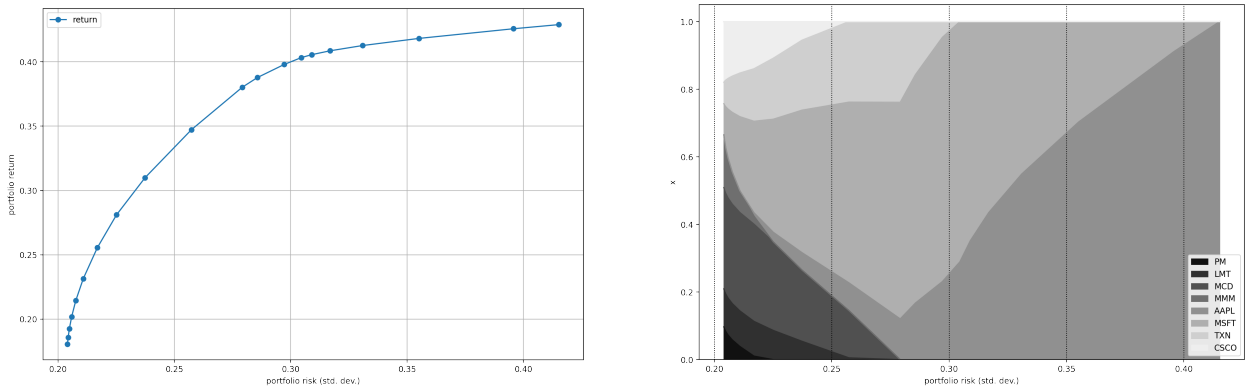
- box constraints,
- leverage or no-short constraints,
- turnover regularization,
- sector or exposure constraints,
- linear transaction-cost terms.

Explain clearly:

- why the modification is introduced;
- whether it enters the model as a hard constraint or a penalty;

- how it changes the optimization class, if at all;
- how it affects the frontier or the resulting portfolio weights.

Figure 2 provides two standard visual references for this part: the efficient frontier in the risk-return plane and the change in portfolio composition as the risk parameter varies. These plots are useful when discussing both the QP-SOCP comparison and the economic effect of risk aversion.



(a) Efficient frontier in the risk-return plane for a standard portfolio optimization model.

(b) Optimal portfolio weights for different values of the risk-aversion parameter γ.

Figure 2: Reference figures that are useful for explaining the QP-SOCP comparison and the effect of varying the risk parameter. Source: MOSEK Portfolio Optimization Cookbook 2.4.

5 Task 3: ADMM for the Core QP

In this section, implement ADMM from scratch for Model A2:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \frac{\gamma}{2}\boldsymbol{x}^\top\hat{\Sigma}\boldsymbol{x} - \hat{\boldsymbol{\mu}}^\top\boldsymbol{x} \quad \text{subject to} \quad \mathbf{1}^\top\boldsymbol{x} = 1, \quad \boldsymbol{x} \geq 0.$$

Define the simplex

$$\Delta = \left\{\boldsymbol{z} \in \mathbb{R}^n : \mathbf{1}^\top\boldsymbol{z} = 1,\ \boldsymbol{z} \geq 0\right\}. \tag{5.1}$$

Introduce a splitting variable $\boldsymbol{z}$ and write

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{z}} \quad & \frac{\gamma}{2}\boldsymbol{x}^\top\hat{\Sigma}\boldsymbol{x} - \hat{\boldsymbol{\mu}}^\top\boldsymbol{x} + I_\Delta(\boldsymbol{z}) \\ \text{subject to} \quad & \boldsymbol{x} - \boldsymbol{z} = 0. \end{aligned} \tag{5.2}$$

Scaled ADMM Updates

With penalty parameter $\rho > 0$, the scaled ADMM iterations are

$$\boldsymbol{x}^{k+1} = (\gamma\hat{\Sigma} + \rho I)^{-1}\left(\hat{\boldsymbol{\mu}} + \rho(\boldsymbol{z}^k - \boldsymbol{u}^k)\right), \tag{5.3}$$

$$\boldsymbol{z}^{k+1} = \mathrm{Proj}_{\Delta}\left(\boldsymbol{x}^{k+1} + \boldsymbol{u}^{k}\right), \tag{5.4}$$

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^{k} + \boldsymbol{x}^{k+1} - \boldsymbol{z}^{k+1}. \tag{5.5}$$

Define the primal and dual residuals by

$$r_{\mathrm{prim}}^{k} = \|\boldsymbol{x}^{k} - \boldsymbol{z}^{k}\|_2, \tag{5.6}$$

$$r_{\mathrm{dual}}^{k} = \rho\|\boldsymbol{z}^{k} - \boldsymbol{z}^{k-1}\|_2. \tag{5.7}$$

Task 5.1 (ADMM implementation)**.** (a) Implement ADMM for the simplex-constrained QP above.

(b) Implement the Euclidean projection onto the simplex Δ.

(c) Report objective-value history, residual history, running time, and iteration count.

(d) Study at least four values of ρ.

(e) Compare the final ADMM solution against the commercial-solver baseline.

6 Task 4: PDHG for the Core QP

Use the same core model (2.6). Define

$$f(\boldsymbol{x}) = \frac{\gamma}{2}\boldsymbol{x}^{\top}\hat{\Sigma}\boldsymbol{x} - \hat{\boldsymbol{\mu}}^{\top}\boldsymbol{x}, \tag{6.1}$$

and

$$K\boldsymbol{x} = \begin{pmatrix} \mathbf{1}^{\top}\boldsymbol{x} \\ -\boldsymbol{x} \end{pmatrix}. \tag{6.2}$$

Let

$$g(u_0, u_1) = I_{\{1\}}(u_0) + I_{\mathbb{R}^n_-}(u_1), \tag{6.3}$$

so that the constraint set is encoded by

$$K\boldsymbol{x} \in \{1\} \times \mathbb{R}^n_-.$$

Then the problem becomes

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) + g(K\boldsymbol{x}). \tag{6.4}$$

The convex conjugate is

$$g^*(p, q) = p + I_{\mathbb{R}^n_+}(q), \qquad p \in \mathbb{R},\ q \in \mathbb{R}^n, \tag{6.5}$$

and the saddle-point form is

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \max_{p\in\mathbb{R},\ q\in\mathbb{R}^n_+} \left\{\frac{\gamma}{2}\boldsymbol{x}^{\top}\hat{\Sigma}\boldsymbol{x} - \hat{\boldsymbol{\mu}}^{\top}\boldsymbol{x} + p(\mathbf{1}^{\top}\boldsymbol{x} - 1) - q^{\top}\boldsymbol{x}\right\}. \tag{6.6}$$

PDHG Updates

Choose step sizes $\tau > 0$ and $\sigma > 0$ satisfying

$$\tau\sigma\|K\|_2^2 < 1. \tag{6.7}$$

Using the standard Chambolle–Pock form with extrapolation parameter $\theta = 1$, the iterations are

$$p^{k+1} = p^k + \sigma(\mathbf{1}^\top \bar{\boldsymbol{x}}^k - 1), \tag{6.8}$$

$$q^{k+1} = \mathrm{Proj}_{\mathbb{R}^n_+}\left(q^k - \sigma\bar{\boldsymbol{x}}^k\right), \tag{6.9}$$

$$\boldsymbol{x}^{k+1} = (I + \tau\gamma\hat{\Sigma})^{-1}\left(\boldsymbol{x}^k - \tau(p^{k+1}\mathbf{1} - q^{k+1}) + \tau\hat{\boldsymbol{\mu}}\right), \tag{6.10}$$

$$\bar{\boldsymbol{x}}^{k+1} = \boldsymbol{x}^{k+1} + (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k). \tag{6.11}$$

Task 6.1 (PDHG implementation)**.** (a) Implement PDHG for the same core QP.

(b) Test several valid step-size pairs (τ, σ).

(c) Report objective-value history, feasibility history, running time, and iteration count.

(d) Compare PDHG against ADMM and against the commercial-solver baseline.

Remark 6.1. A clean comparison table should include at least the final objective value, final feasibility residual, running time, iteration count, and a brief note on parameter sensitivity [4, 5, 2].

7 Task 5: real-data backtesting and comparison

This section is mandatory. The project should not stop at in-sample optimization.

Task 7.1 (Time split and model selection)**.** Split the data in strict time order. You may use either

- a train/validation/test split, or
- a rolling-window protocol.

In either case, the report must explain:

- the start and end dates of each stage;
- where hyperparameter tuning is performed;
- how model selection is made;
- how information leakage is avoided.

Task 7.2 (Backtesting)**.** Backtest at least the following strategies:

- an equal-weight benchmark;
- one QP-based optimized strategy;

- one SOCP-based optimized strategy.

If you implement the turnover model, include it as an additional strategy.

For each strategy, report at least:

- cumulative return or cumulative wealth;
- annualized return;
- annualized volatility;
- Sharpe ratio;
- maximum drawdown;
- average turnover;
- sensitivity to transaction costs.

You must also state:

- rebalancing frequency;
- training-window length;
- whether transaction costs are included;
- whether market impact or slippage is ignored or approximated.

Task 7.3 (Sensitivity analysis and failure case)**.** At least one sensitivity study is required. Possible directions include:

- sensitivity to γ in Model A2;
- sensitivity to $\sigma_{\max}$ in Model B;
- sensitivity to λ in the turnover model;
- sensitivity to ADMM or PDHG hyperparameters;
- sensitivity to the asset-universe size.

In addition, document at least **one failure case**, for example:

- unstable weights under a poor parameter choice;
- weak out-of-sample performance in a specific market regime;
- excessive turnover;
- numerical difficulty caused by nearly singular covariance estimates.

Explain which behavior is due to the optimization model and which behavior is due to the market sample.

8 Recommended Extensions

The following directions are **recommended**, but not required for every submission.

Extension A: Factor-based Covariance Modeling

Instead of using the sample covariance directly, you may use a factor model

$$r = Ff + \varepsilon, \tag{8.1}$$

with

$$\mathrm{Cov}(f, \varepsilon) = 0, \qquad \mathrm{Cov}(\varepsilon) = D, \tag{8.2}$$

so that the covariance structure becomes

$$\Sigma = F\Sigma_f F^\top + D, \tag{8.3}$$

where F is a factor-loading matrix, Σ_f is the factor covariance matrix, and D is a diagonal or structured idiosyncratic covariance matrix.

Possible questions to explore:

- How does factor-based estimation change the stability of the optimized weights?
- How does it affect out-of-sample performance and computation?
- Can factor exposure constraints be incorporated naturally?

Extension B: Mean-CVaR Portfolio Optimization

You may also study a scenario-based mean-CVaR model. Let $R \in \mathbb{R}^{T\times n}$ be a return matrix whose t-th row is the asset return vector in scenario t. For a confidence level $\beta \in (0, 1)$, one convex formulation is

$$\begin{aligned} \max_{\boldsymbol{x},\alpha,\xi} \quad & \hat{\boldsymbol{\mu}}^\top \boldsymbol{x} - \gamma \left(\alpha + \frac{1}{(1-\beta)T} \sum_{t=1}^{T} \xi_t \right) \\ \text{subject to} \quad & \xi_t \geq -R_t^\top \boldsymbol{x} - \alpha, \qquad t = 1, \ldots, T, \\ & \xi_t \geq 0, \qquad t = 1, \ldots, T, \\ & \mathbf{1}^\top \boldsymbol{x} = 1, \\ & \boldsymbol{x} \geq 0. \end{aligned} \tag{8.4}$$

Possible questions to explore:

- How do mean-variance and mean-CVaR portfolios differ in extreme market periods?
- How sensitive are the solutions to the confidence level β?
- How do the weights and drawdowns compare out of sample?

9 Bonus Task: Parallel PDHG

This part is optional.

Consider a multi-period portfolio sequence $\boldsymbol{x}_t$, $t = 1, \ldots, T$, with model

$$\begin{aligned} \min_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T} \quad & \sum_{t=1}^{T} \left(\frac{\gamma}{2} \boldsymbol{x}_t^\top \hat{\Sigma}_t \boldsymbol{x}_t - \hat{\boldsymbol{\mu}}_t^\top \boldsymbol{x}_t \right) + \lambda \sum_{t=2}^{T} \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_1 \\ \text{subject to} \quad & \mathbf{1}^\top \boldsymbol{x}_t = 1, \quad \boldsymbol{x}_t \geq 0, \quad t = 1, \ldots, T. \end{aligned} \tag{9.1}$$

If the coupling terms and constraints are moved into the linear operator of a PDHG formulation, then the primal update separates over time:

$$\boldsymbol{x}_t^{k+1} = (I + \tau\gamma\hat{\Sigma}_t)^{-1} \left(v_t + \tau \hat{\boldsymbol{\mu}}_t \right), \tag{9.2}$$

where v_t collects the current linear terms coming from the dual update.

Task 9.1 (Bonus: parallel PDHG)**.** (a) Identify which parts of the PDHG iteration can be parallelized.

(b) Implement a serial version and a parallel version.

(c) Compare wall-clock time on a reasonably large instance.

(d) Explain when parallelism helps and when it does not.

References

[1] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[2] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[3] Harry M Markowitz. *Portfolio selection: efficient diversification of investments*. Yale university press, 2008.

[4] Parikh Neal, Chu Eric, Peleato Borja, and Eckstein Jonathan. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[5] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.