Assignment 5: Markowitz Portfolio Optimization and KKT Conditions

Math4AI: Calculus & Optimization

Fatima Alibabayeva

fatime.elibabayeva25@aiacademy.az

National AI Academy

January 8, 2026

Contents

1 Introduction

Modern Portfolio Theory (MPT), introduced by Harry Markowitz, revolutionized finance by quantifying the trade-off between expected return and risk. The fundamental problem is to allocate weights to different assets such that the portfolio variance (risk) is minimized for a specific target return.

In this report, we implement two variations of the Markowitz optimization problem:

1. **Unconstrained Optimization (Short Selling Allowed):** Solved using Lagrange Multipliers.
2. **Constrained Optimization (No Short Selling, $w \geq 0$):** Solved using Karush-Kuhn-Tucker (KKT) conditions and numerical Quadratic Programming (SLSQP).

2 Mathematical Foundation

2.1 Objective Function and Constraints

Let w be the vector of asset weights, Σ the covariance matrix of asset returns, and μ the vector of expected returns. The optimization problem is:

$$\min_{w} \frac{1}{2} w^T \Sigma w \tag{1}$$

Subject to:

- $\sum w_i = 1$ (Budget constraint: $1^T w = 1$)
- $\sum \mu_i w_i = R$ (Target return constraint: $\mu^T w = R$)
- $w_i \geq 0$ (Optional: No short selling constraint)

2.2 Part 1: Lagrange Multipliers (Equality Constraints)

When short selling is allowed, we only have equality constraints. The Lagrangian function is:

$$\mathcal{L}(w, \lambda_1, \lambda_2) = \frac{1}{2} w^T \Sigma w + \lambda_1 (1 - 1^T w) + \lambda_2 (R - \mu^T w) \tag{2}$$

Taking the partial derivatives and setting them to zero leads to a linear system (KKT system):

$$\begin{bmatrix} \Sigma & 1 & \mu \\ 1^T & 0 & 0 \\ \mu^T & 0 & 0 \end{bmatrix} \begin{bmatrix} w \\ \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ R \end{bmatrix} \tag{3}$$

This system is solved using `np.linalg.solve` for a direct analytical solution.

2.3 Part 2: KKT Conditions and Inequality Constraints

When $w \geq 0$ is enforced, the problem becomes a Quadratic Programming (QP) problem. The KKT conditions introduce complementarity slackness:

$$\lambda_i w_i = 0, \quad \lambda_i \geq 0, \quad w_i \geq 0 \tag{4}$$

Because analytical solutions for many inequality constraints are difficult, we utilize the **SLSQP (Sequential Least Squares Programming)** algorithm via SciPy, which iteratively finds the feasible point satisfying the KKT conditions.

3 Design Choices and Implementation

3.1 Numerical Stability

In the implementation, we ensure the covariance matrix Σ is positive semi-definite. If numerical errors produce tiny negative eigenvalues, we apply a small regularization term (10^{-11}) to the diagonal:

$$\Sigma_{reg} = \Sigma + \epsilon I \tag{5}$$

3.2 Target Return Grid

Achievable returns are bounded by $[\min(\mu), \max(\mu)]$. We generate a grid of 25 points within this range to map the "Efficient Frontier"—the set of portfolios offering the lowest risk for each return level.

4 Analysis of Results

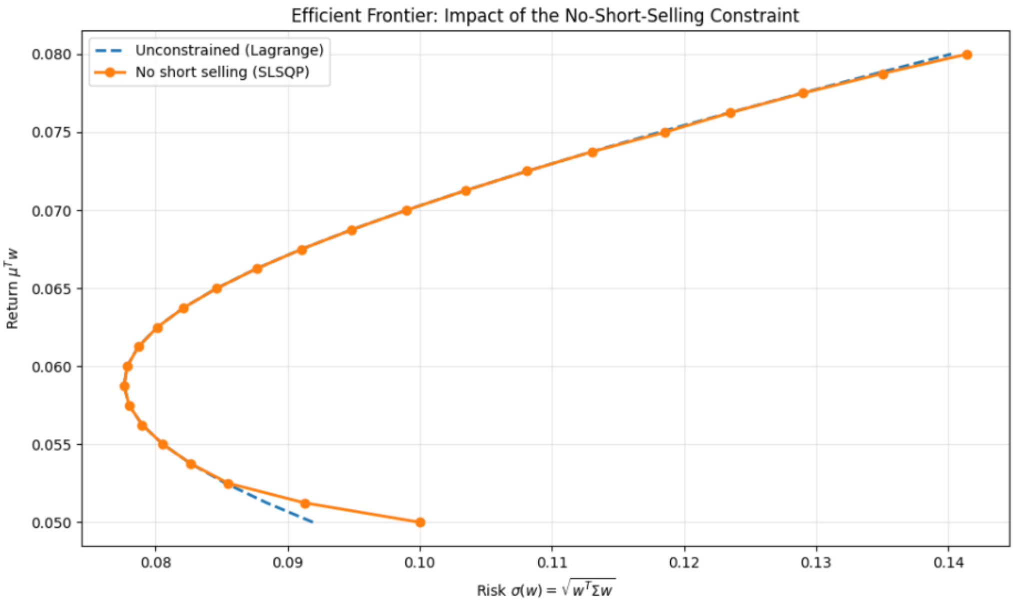

Figure 1: Efficient Frontier Comparison: Lagrange vs. KKT Constraints

4.1 Observations

- **Lagrange (Short Selling Allowed):** The frontier is a smooth hyperbola. Because weights can be negative, the optimizer has more freedom to cancel out risks, leading to a lower overall variance.
- **No Short Selling ($w \geq 0$):** The frontier sits to the "right" of the unconstrained one. Enforcing positivity restricts the feasible region, resulting in higher risk for the same level of return.
- **KKT Implementation:** The SciPy SLSQP optimizer successfully converged at each grid point, maintaining the budget and return constraints within a tolerance of 10^{-9}.

5 Conclusion

This assignment demonstrates how optimization theory applies to financial decision-making. We showed that adding inequality constraints (like the no-short-selling rule) increases the portfolio's minimum risk but reflects real-world market limitations. The transition from solving a linear KKT system (Lagrange) to using iterative numerical methods (SLSQP) highlights the shift from theoretical calculus to applied computational optimization.

6 Appendix: Code Excerpt

```
# Construction of the Block KKT Matrix for Lagrange
mu_col = mu.reshape(-1, 1)
A = np.block([
    [Sigma,    ones,      mu_col],
    [ones.T,   np.zeros((1, 1)), np.zeros((1, 1))],
    [mu_col.T, np.zeros((1, 1)), np.zeros((1, 1))]
])

# Solving the inequality constrained problem via SLSQP
res = minimize(objective, w0, method="SLSQP", bounds=bounds,
               constraints=constraints, jac=grad)
```