

Financial Optimization

ISE 347/447

Lecture 8

Dr. Ted Ralphs

Reading for This Lecture

- C&T Chapter 8

Review from Last Time

- Last time, we discussed the Konno-Yamazaki model for portfolio optimization.
- The decision variables $x = (x_1, \dots, x_n)$ expressed the relative proportions by which available capital is invested in n risky assets with (random) returns $R = (R_1, \dots, R_n)$.
- The model

$$\begin{aligned} \text{(KY)} \quad & \min_x \ell(x) \\ & \text{s.t.} \quad Ax \geq a, \\ & \quad \quad Bx = b \end{aligned}$$

was based on the mean absolute deviation risk $\ell(x)$.

- The linear constraints $Ax \geq a$, $Bx = b$ were assumed to contain the constraints

$$\mu^\top x \geq r \qquad \sum_i x_i = 1.$$

Markowitz Portfolio Optimization

- We now move on to discuss the more classical portfolio model of Henry Markowitz.
- This model is based on using the variance of the portfolio return as a risk measure:

$$\sigma^2(R^\top x) = x^\top Qx,$$

where $Q = \text{Cov}(R_i, R_j)$ is the variance-covariance matrix of the vector of returns R .

- As before, we have a tradeoff between risk and expected return.
- We will consider three different single-objective models that can be used to analyze the tradeoff between these conflicting goals.

Three Markowitz Models

$$\begin{aligned} \text{(M1)} \quad & \min_{x \in \mathbb{R}^n} x^\top Q x \\ & \text{s.t.} \quad \mu^\top x \geq r, \\ & \quad \sum_{i=1}^n x_i = 1, \end{aligned}$$

where r is a targeted minimum expected portfolio return.

$$\begin{aligned} \text{(M2)} \quad & \max_{x \in \mathbb{R}^n} \mu^\top x \\ & \text{s.t.} \quad x^\top Q x \leq \sigma^2 \\ & \quad \sum_{i=1}^n x_i = 1, \end{aligned}$$

where σ^2 is the maximum risk the investor is willing to take on.

Three Markowitz Models (cont.)

$$(M3) \quad \max_{x \in \mathbb{R}^n} \mu^\top x - \lambda x^\top Q x$$
$$\text{s.t.} \quad \sum_{i=1}^n x_i = 1,$$

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

- All three models are examples of *quadratic programming* problems,
- Also, since Q is a positive semidefinite symmetric matrix, then $x \mapsto x^\top Q x$ is a convex function.
- Hence, these are actually *convex quadratic programs*.
- Convex quadratic programs can generally be solved efficiently.

Refining the Models

- In practice, the optimal portfolio weights x^* computed under the above models (M1)–(M3) can have some undesirable features:
- The weights $|x_i|^*$ may be very large for a few of the assets, which means taking large short-long positions in a small number of assets.
- This can be accompanied by instability, whereby a small error in the estimation of the model parameters (μ, Q) can lead to big changes in x^* .
- In a multiperiod situation, this leads to large transaction costs.
- The true risk of the optimal portfolio may not be known accurately.

Additional Constraints

- These drawbacks are addressed by introducing further linear constraints.
- To avoid excessive short or long positions, we can use box constraints

$$l_i \leq x_i \leq u_i, \quad (i = 0, \dots, n)$$

- Similarly, portfolio turnover constraints can be imposed as follows.
 - Suppose that at the beginning of an investment period, we already hold a portfolio that corresponds to the relative weight allocation weights \tilde{x}_i ($i = 1, \dots, n$).
 - Since the model parameters (μ, Q) change over time, we have new estimates that tell us that for the purposes of an optimal investment strategy we should change (“rebalance”) our portfolio to new weights x_i ($i = 1, \dots, n$).

Reducing Transaction Costs

- To keep transaction costs down, an upper bound h on the proportion of the portfolio value that is traded can be imposed.

$$x_i - \tilde{x}_i \leq y_i, \quad y_i \geq 0, \quad \forall i$$

$$\tilde{x}_i - x_i \leq z_i, \quad z_i \geq 0, \quad \forall i$$

$$\sum_i (y_i + z_i) \leq h.$$

- To further refine, transaction costs due to fees and price movement caused by the trade in asset i (“slippage”) can be crudely approximated by linear functions

$$TC_i = \begin{cases} \tau_i^s |x_i - \tilde{x}_i| V & \text{if } x_i - \tilde{x}_i < 0, \\ \tau_i^l |x_i - \tilde{x}_i| V & \text{if } x_i - \tilde{x}_i \geq 0, \end{cases}$$

where the parameters τ_i^s, τ_i^l depend on the asset i and V is the total current value of the portfolio.

Reducing Transaction Costs (cont.)

- Writing $\chi_i^s = \tau_i^s V$ and $\chi_i^l = \tau_i^l V$, we can keep transaction costs in check by subtracting them from the expected return, e.g.,

$$\min_{x,y,z} x^\top Qx$$

$$\text{s.t.} \quad \sum_i (\mu_i x_i - \chi_i^l y_i - \chi_i^s z_i) \geq r,$$

$$\sum_i x_i = 1,$$

$$x_i - \tilde{x}_i \leq y_i, \quad (i = 1, \dots, n),$$

$$\tilde{x}_i - x_i \leq z_i, \quad (i = 1, \dots, n),$$

$$y_i \geq 0, \quad \forall (i = 1, \dots, n),$$

$$z_i \geq 0, \quad (i = 1, \dots, n).$$

Diversification

- To ensure diversification of the portfolio, assets can be bundled into different sectors corresponding to the (possibly overlapping) index sets $\mathcal{I}_1 \cup \dots \cup \mathcal{I}_k = \{1, \dots, n\}$.
- Then one can impose constraints on the proportion of total wealth that is allocated to each sector.

$$\ell_j \leq \sum_{i \in \mathcal{I}_j} x_i \leq u_j, \quad (j = 1, \dots, k),$$

where $0 \leq \ell_j < u_j \leq 1$.

- For example if a pension fund is required by law to hold at least 25% of their assets in bonds, then $\mathcal{I} \subseteq \{1, \dots, n\}$ could be the indices corresponding to bonds.
- The following constraint would then be imposed,

$$\sum_{i \in \mathcal{I}} x_i \geq 0.25.$$

General Markowitz Portfolio Model

- A general Markowitz portfolio optimization problem would thus take the following form,

$$\begin{aligned} \text{(M)} \quad & \min_x x^\top Q x \\ & \text{s.t.} \quad \mu^\top x \geq r, \\ & \quad Ax \geq a, \\ & \quad Bx = b, \end{aligned}$$

- Any of the other alternative formulations discussed previously could also be used.
- We've singled out the inequality constraint $\mu^\top x \geq r$ because it contains the extra parameter r .
- Note that since the covariance matrix Q is positive semidefinite, (M) is still a convex QP.

AMPL model for Portfolio Optimization

```
set assets;                # asset categories
set T := {1984..1994};    # years

param max_risk default 0.00305;
param R {T,assets};
param mean {j in assets} := (sum{i in T} R[i,j])/card(T);
param Rtilde {i in T, j in assets} := R[i,j] - mean[j];
param Q {i in assets, j in assets} := sum{k in T}
(R[k, i] - model.mean[i])*(model.R[k, j] - model.mean[j]);

var alloc{assets} >=0;

minimize reward: - sum{j in assets} mean[j]*alloc[j] ;

subject to risk_bound: sum{i in assets}
(sum{j in assets} Q[i,j]*alloc[i]*alloc[j]) <= max_risk;
subject to tot_mass: sum{j in assets} alloc[j] = 1;
```

Pyomo model for Portfolio Optimization

```
model = AbstractModel()

model.assets = Set()
model.T = Set(initialize = range(1994, 2014))
model.max_risk = Param(initialize = .00305)
model.R = Param(model.T, model.assets)
def mean_init(model, j):
    return sum(model.R[i, j] for i in model.T)/len(model.T)
model.mean = Param(model.assets, initialize = mean_init)
def Q_init(model, i, j):
    return sum((model.R[k, i] - model.mean[i])*(model.R[k, j]
        - model.mean[j]) for k in model.T)
model.Q = Param(model.assets, model.assets, initialize = Q_init)

model.alloc = Var(model.assets, within=NonNegativeReals)
```

Pyomo model for Portfolio Optimization (cont'd)

```
def risk_bound_rule(model):
    return (
        sum(sum(model.Q[i, j] * model.alloc[i] * model.alloc[j]
                for i in model.assets) for j in model.assets)
        <= model.max_risk)
model.risk_bound = Constraint(rule=risk_bound_rule)

def tot_mass_rule(model):
    return (sum(model.alloc[j] for j in model.assets) == 1)
model.tot_mass = Constraint(rule=tot_mass_rule)

def objective_rule(model):
    return sum(model.alloc[j]*model.mean[j] for j in model.assets)
model.objective = Objective(sense=maximize, rule=objective_rule)
```