MIE377 – Financial Optimization Models

Lab 1 (12–Jan–2024)

Linear and Quadratic Programming

1 Linear Programming

Reference: `https://www.mathworks.com/help/optim/ug/linprog.html`

Suppose a bank has liabilities at the end of the next three years:

Year 1: \$12,000 Year 2: \$18,000 Year 3: \$20,000

Now, suppose the bank wishes to meet these liabilities by buying units of three bonds:

Bond	Price (\$)	Yearly Coupon(\$)	Maturity Year
1	102	5	1
2	99	3.5	2
3	98	3.5	3

All bonds have a face value of \$100. The bank must meet or exceed their liability for each year end. How many units of each bond should the bank purchase to meet the liabilities with minimum cost?

We can formulate the problem as a linear program to find the optimal number of bonds to purchase today (present-time). First, consider the liabilities we MUST meet (i.e., the constraints).

Let x_i denote the quantity of bond i we should buy. Let F_i and c_i denote the face value and coupon of bond i, respectively. Now, to meet the liabilities L we must comply with the following requirements

$$\begin{aligned}(F_1 + c_1)x_1 + c_2x_2 + c_3x_3 &\geq L_1\\ (F_2 + c_2)x_2 + c_3x_3 &\geq L_2\\ (F_3 + c_3)x_3 &\geq L_3\end{aligned}$$

Finally, if we wish to minimize cost, we should set the following objective function

$$\min_x \quad P_1x_1 + P_2x_2 + P_3x_3$$

where P_i denotes the price of bond i. Thus, our complete linear optimization program is

$$\begin{array}{lll}\min\limits_x & P_1x_1 + P_2x_2 + P_3x_3 & \\ \text{s.t.} & (F_1 + c_1)x_1 + c_2x_2 + c_3x_3 & \geq L_1\\ & (F_2 + c_2)x_2 + c_3x_3 & \geq L_2\\ & (F_3 + c_3)x_3 & \geq L_3\\ & x_i \geq 0 \quad \text{for } i = 1, 2, 3 & \end{array}$$

We add the last constraint to prohibit short-selling the bonds. We can solve this problem using the function 'linprog' in MATLAB.

2 Quadratic Programming

Reference: `https://www.mathworks.com/help/optim/ug/quadprog.html`

For the second example, we will construct an optimal portfolio using n stocks. This portfolio will seek to minimize risk while being subject to a target return.

Suppose you are given some raw pricing data for asset i, P_{it}, corresponding to time t. First, we must calculate the rate of return,

$$r_{it} = \frac{P_{it} - P_{i,t-1}}{P_{i,t-1}}$$

If we start with $t = 0, ..., T$ price data points, we will have $t = 1, ..., T$ return data points.

Next, we must calculate the expected return of asset i. If we are using raw data, we can calculate the geometric average as follows

$$\mu_i = \big(\prod_{t=1}^{T}(1 + r_{it})\big)^{1/T} - 1,$$

which we can use to set a target expected return for our portfolio. We will use variance as a measure of financial risk. To calculate the variance of asset i from raw data, we have

$$\sigma_i^2 = \frac{1}{T-1}\sum_{t=1}^{T}(r_{it} - \bar{r}_i)^2$$

where $\bar{r}_i$ denotes the arithmetic mean of asset i. If we have two assets i and j, their covariance is

$$\sigma_{ij} = \frac{1}{T-1}\sum_{t=1}^{T}(r_{it} - \bar{r}_i)(r_{jt} - \bar{r}_j)$$

If we let $\boldsymbol{x} \in \mathbb{R}^n$ denote the asset weights of our portfolio, then the portfolio expected return can be described as

$$\mu_p = \sum_{i=1}^{n} \mu_i x_i = \boldsymbol{\mu}^T \boldsymbol{x}$$

where $\boldsymbol{\mu} \in \mathbb{R}^n$ is the vector of our estimated expected returns. In the same fashion, our portfolio variance is

$$\sigma_p^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{ij} x_i x_j = \boldsymbol{x}^T \boldsymbol{Q}\ \boldsymbol{x}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ denotes the covariance matrix, where the diagonal elements are the asset variances and the off-diagonal terms correspond to the covariances.

Finally, for a target return $R = 0.25\%$, our optimization problem is

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \boldsymbol{x}^T \boldsymbol{Q}\ \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{\mu}^T \boldsymbol{x} \geq 0.0025 \\ & \mathbf{1}^T \boldsymbol{x} = 1 \\ & \boldsymbol{x} \geq \mathbf{0} \end{aligned}$$

As before, we add the last constraint to prohibit short-selling the stocks. We can solve this problem using the function 'quadprog' in MATLAB.