Portfolio Analysis

M. Kateregga

November 21, 2014 @ AIMS South Africa

Outline

Introduction

A portfolio is an investment made in n assets A_k with the returns R_k, $k = 1, \cdots, n$ using some amount of wealth W.

Suppose W_k is the amount of wealth invested in asset A_k, then

$$\sum_{k=1}^{n} W_k = W.$$

It is better to represent the investments in terms of relative values i.e.

$$w_k = \frac{W_k}{W}, \quad \text{such that} \quad \sum_{k=1}^{n} w_k = 1 \quad \text{and} \quad \sum_{k=1}^{n} w_k R_k = R_p.$$

We shall refer to w_k as the weight of asset A_k with respect to the total portfolio.

Preliminary

The return R_k of a financial asset A_k is described as a random variable.

The mean and variance of R_k can be estimated from historical data.

Individual stocks in a portfolio may not be normally distributed but a portfolio of many of them tends to normal distribution.

Performance is measured in terms of realized return and variance of a portfolio.

The expected return is interpreted as the reward of the investment and variance as its risk.

The Optimization Problem

The problem can be formulated as follows:

Consider a set of financial assets, characterized by their expected mean and their covariances.

We need to find the optimal weight of each asset, such that the overall portfolio provides the smallest risk for a given overall return.

Therefore, the problem is to find the 'efficient frontier' or the set of all achievable portfolios that offer the highest rate of return for a given level of risk.

Return of a Portfolio

The expected return of the portfolio, $\mathbb{E}[R_p]$ is the weighted average of the expected returns of the individual components i.e.

$$\mathbb{E}[R_p] = \sum_{k=1}^{n} w_k \mathbb{E}[R_k],$$

w_k is the proportion of the portfolio invested in security k and we require that

$$\sum_{k=1}^{n} w_k = 1.$$

The realized return is a linear combination of the returns of the individual investments.

Variance of a Portfolio

Let $\sigma_{ij} = \mathbb{E}[(R_i - \mathbb{E}[R_i])(R_j - \mathbb{E}[R_j])]$. The portfolio variance is given by

$$\sigma_p^2 = \mathbb{E}[|R_p - \mathbb{E}[R_p]|^2] = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \sigma_{ij}.$$

w_i is the proportion of total investment in security i.

$\sigma_{ij} = \sigma_{ji} = \rho_{ij}\sigma_i\sigma_j$ is the covariance between assets A_i and A_j

$\rho_{ij} \in [-1, 1]$ is the correlation coefficient between securities i and j.

σ_i is the standard deviation of security i.

Example: The 3-Assets portfolio

We consider a portfolio of stocks 1, 2 and 3 respectively.

The portfolio variance is then given by

$$\begin{aligned}\sigma_p^2 &= \sum_{i=1}^{3}\sum_{j=1}^{3} w_i w_j \rho_{ij}\sigma_i\sigma_j.\\ &= w_1w_1\rho_{11}\sigma_1\sigma_1 + w_1w_2\rho_{12}\sigma_1\sigma_2 + w_1w_3\rho_{13}\sigma_1\sigma_3\\ &+ w_2w_1\rho_{21}\sigma_2\sigma_1 + w_2w_2\rho_{22}\sigma_2\sigma_2 + w_2w_3\rho_{23}\sigma_2\sigma_3\\ &+ w_3w_1\rho_{31}\sigma_3\sigma_1 + w_3w_2\rho_{32}\sigma_3\sigma_2 + w_3w_3\rho_{33}\sigma_3\sigma_3.\end{aligned}$$

Observe that $\rho_{ii} = 1$ for all i.

The 3-Asset portfolio

Table 1: 3-Asset portfolio

	Stock 1	Stock 2	Stock 3
Expected Return	0.15	0.20	0.18
Variance	0.50	0.40	0.70
Standard deviation	0.224	0.250	0.324
Weight	35%	45%	20%
Correlation coefficients	$\rho_{12} = 0.4$	$\rho_{23} = 0.5$	$\rho_{13} = 0.6$

Exercise 1: Compute the expected return and variance of this portfolio.

Minimum Variance Portfolio

The task is to determine a particular combination of stocks that will result in a least possible portfolio variance.

This is a minimization problem. We apply basic calculus principles i.e.

$$\frac{d\sigma_p^2}{dw_1} = 0.$$

We make the following restrictions:

$$w_1 = 0.4w_2 \text{ and } w_3 = 1 - w_1 - w_2.$$

Exercise 2: Determine the weights w_1, w_2 and w_3 that constitute the optimal portfolio.

Matrix notation: 3-Asset portfolio

The weight and covariance matrices are given by

$$w = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \quad \text{and} \quad V = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix}.$$

where $\sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$ thus $\sigma_{ii} = \sigma_i^2$. Therefore the portfolio variance is given by

$$\sigma_p^2 = \begin{pmatrix} w_1 & w_2 & w_3 \end{pmatrix} \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix}.$$

General Case: n-Asset portfolio

We calculate the covariance matrix which is a pairwise combination of all portfolio constituents.

A n-stock portfolio requires $(n^2 - n)/2$ covariances of its components to compute its variance.

The portfolio variance σ_p^2 in this case is given by

$$\sigma_p^2 = \begin{pmatrix} w_1 & \cdots & w_n \end{pmatrix} \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix}.$$

In Summary

We shall denote:

- w to be the vertical vector of weights of the different stocks in the portfolio
- μ as the vertical vector of expected returns.
- V to be the variance-covariance matrix.
- the variance of the portfolio is therefore given by

$$\sigma_p^2 = w^t V w,$$

  where w^t denotes the transpose of w.

The idea is to minimize this variance subject to the constraints of achieving a given expected return [Portfolio Optimization].

Portfolio Optimization in n-stocks case

We wish to minimize $w^t V w$ over w i.e.

$$\underset{w}{\text{Minimize}}\; w^t V w,$$

subject to:

$$\begin{cases} \mathbf{1}^t w = 1 \\ \mu^t w = \mathbb{E}[R_p] \end{cases},$$

where $\mathbb{E}[R_p]$ is the desired return on the portfolio, $\mathbf{1}$ is a vector of ones,

$$\mu = \begin{pmatrix} \mathbb{E}[R_1] \\ \vdots \\ \mathbb{E}[R_n] \end{pmatrix} \quad \text{and} \quad w = \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix}.$$

The Lagrangian Method

The Lagrangian is defined as

$$L(w, \lambda, \gamma) = w^t V w + (\mathbb{E}[R_p] - w^t \mu)\lambda + (1 - w^t \mathbf{1})\gamma,$$

where λ and γ are Lagrangian multipliers. The critical point of the Lagrangian is obtained by solving a system of equations:

$$\nabla_w L(w, \lambda, \gamma) = 2Vw - \mu\lambda - \mathbf{1}\gamma = 0. \tag{1}$$

$$\frac{\partial L(w, \lambda, \gamma)}{\partial \lambda} = \mathbb{E}[R_p] - w^t \mu = 0. \tag{2}$$

$$\frac{\partial L(w, \lambda, \gamma)}{\partial \gamma} = 1 - w^t \mathbf{1} = 0. \tag{3}$$

The Lagrangian Method

From equation (1), we obtain

$$w = \frac{1}{2}V^{-1}[\mu\lambda + \mathbf{1}\gamma].$$

Then from equations (2) and (3) we deduce

$$\mu^t V^{-1}\mu\lambda + \mathbf{1}^t V^{-1}\mu\gamma = 2\mathbb{E}[R_p].$$
$$\mu^t V^{-1}\mathbf{1}\lambda + \mathbf{1}^t V^{-1}\mathbf{1}\gamma = 2.$$

We can rewrite this as

$$A\begin{pmatrix}\lambda\\ \gamma\end{pmatrix} = 2\begin{pmatrix}\mathbb{E}[R_p]\\ 1\end{pmatrix} \quad \text{where} \quad A = \begin{pmatrix}\mu^t V^{-1}\mu & \mathbf{1}^t V^{-1}\mu\\ \mu^t V^{-1}\mathbf{1} & \mathbf{1}^t V^{-1}\mathbf{1}\end{pmatrix}.$$

The Lagrangian Method

For the system to have a solution, the following matrix A must be invertible

The matrix V is positive definite and so is its inverse V^{-1}. Thus, $x^t V^{-1} x > 0$ for any vector $x \neq 0$.

The determinant d of A is greater than zero and is given by

$$d = ([\mu^t V^{-1} \mu][\mathbf{1}^t V^{-1} \mathbf{1}] - [\mathbf{1}^t V^{-1} \mu]^2) > 0.$$

where $\mathbf{1}^t V^{-1} \mu = \mu^t V^{-1} \mathbf{1}$.

The Lagrangian Method

Thus the multipliers are given by

$$\begin{pmatrix}\lambda \\ \gamma\end{pmatrix} = \frac{2}{d}\begin{pmatrix}[\mathbf{1}^t V^{-1}\mathbf{1}] - [\mathbf{1}^t V^{-1}\mu]\mathbb{E}[R_p] \\ -[\mathbf{1}^t V^{-1}\mu] + [\mu^t V^{-1}\mu]\mathbb{E}[R_p]\end{pmatrix}.$$

Therefore the weights of the optimal portfolio can then be given by

$$w = \mathbf{f} + \mathbb{E}[R_p]\mathbf{g},$$

where

$$\mathbf{f} = \frac{1}{d}V^{-1}\Big([\mathbf{1}^t V^{-1}\mathbf{1}]\mathbf{1} - [\mathbf{1}^t V^{-1}\mu]\mu\Big).$$

$$\mathbf{g} = \frac{1}{d}V^{-1}\Big(-[\mathbf{1}^t V^{-1}]\mathbf{1} + [\mu^t V^{-1}\mu]\mu\Big).$$

The Lagrangian Method

Let

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} \mu^t V^{-1} \mu & \mathbf{1}^t V^{-1} \mu \\ \mu^t V^{-1} \mathbf{1} & \mathbf{1}^t V^{-1} \mathbf{1} \end{pmatrix} \quad \text{and} \quad q = \mathbb{E}[R_p].$$

The variance σ_p^2 is then given by

$$\begin{aligned} \sigma_p^2 &= w^t V w. \\ &= q^2 \mathbf{g}^t V \mathbf{g} + q(\mathbf{g}^t V \mathbf{f} + \mathbf{f}^t V \mathbf{g}) + \mathbf{f}^t V \mathbf{f}. \\ &= \frac{a_{11}}{d} \left(q - \frac{a_{12}}{a_{11}} \right)^2 + \frac{1}{a_{11}}. \\ &= \frac{1}{d}(a_{11} q^2 - 2 a_{12} q + a_{22}). \end{aligned}$$

The Lagrangian Method

The equation below of a hyperbola in the space (σ, q)-plane represents the "efficient frontier"

$$\sigma_p^2 = \frac{1}{d}(a_{11}q^2 - 2a_{12}q + a_{22}).$$

We can now obtain the weights of the minimum variance portfolio as

$$w(q) = \mathbf{f} + \frac{a_{12}}{a_{11}}\mathbf{g},$$

and the corresponding risk and return values:

$$\sigma(q) = \sqrt{1/a_{11}} \quad \text{and} \quad q = a_{12}/a_{11}.$$

The sharpe ratio η of a portfolio is defined by $\eta = q/\sigma(q)$.

R Implementation of portfolio Theory

This session will be conducted in the computer lab.

Learners will be provided with a set of step by step instructions for constructing an optimal portfolio in R and plotting the efficient frontier.

The objective of the session is to help learners develop a fully working routine from an algorithm and, learn how to construct optimal portfolios using R.

PORTFOLIO OPTIMIZATION USING R'S SOLVE.QP FUNCTION

IMPORTING OF ESSENTIAL LIBRARIES

1. Get the following packages installed and loaded in R
   stockPortfolio, quadprog, ggplot2

CREATE A PORTFOLIO OF STOCKS FROM ETFS (EXCHANGE TRADED FUNDS)

2. Create a vector called **stocks** containing the following ETFs with their corresponding non efficient weight allocations:

   "VWEHX" = .10, "DIA" = .15, "^GSPC" = .10, "AAPL" = .15, "MSFT" = .25, "JNK" = .15, "GLD" = .10

OBTAIN THE WEEKLY RETURNS FOR EACH SECURITY IN THE PORTFOLIO ABOVE

3. Obtain the portfolio **returns** using getReturns() that takes arguments: names() and freq (frequency). Use **stocks** to obtain the names and set freq to "weekly".

DEFINE THE EFFICIENT FRONTIER FUNCTION

4. Define **eff.frontier** as a function() that takes arguments: **returns** (must be a (m x n) matrix with one column per stock), short (short selling or no), max.allocation (maximum allocation), risk.premium.up (highest risk premium) and risk.increment (increment for the loop). Set short to "no", max.allocation to NULL, risk.premium.up to 0.5, risk.increment to 0.005
   The **eff.frontier** should do the following:
   a. Compute the **covariance** of the **returns** using cov()
   b. Print **covariance**
   c. Set **n** to be the number of columns of the **covariance** matrix. Hint: Use ncol()

SETTING THE CONSTRAINTS ON THE PORTFOLIO

 # If short selling is allowed then:

d. Set the constraints as follows:
   Amat is a (n x 1) matrix of 1's Hint: Use matrix (1, nrow = n)
   bvec is 1
   meq is 1

 # If short selling is not allowed then:

e. Use if() that takes short == "no" as argument and does the following:
   Create a (**n** x **n**+1) diagonal matrix **Amat** of 1's with 1's on first column. Hint: Use cbind(1 diag(**n**)).
   Create a (1 x n+1) vector **bvec** with 1 as first element and zeros elsewhere. Hint: Use c(1 rep(0,**n**))

 # If a maximum allocation is specified and is greater than 1 or less than zero,

f. Use if() that takes !is.null(max.allocation) as argument and does the following:

Use if() that takes max.allocation > 1 or max.allocation < 0 and does the following:

Use stop() that takes argument "max.allocation must be greater than 0 and less than 1"

 # If a maximum allocation is specified and its value times n is less than 1,

Use if() that takes max.allocation*n <1 and does the following:

Use stop() that takes "max.allocation must be set higher, enough assets needed to add to 1"

 # Otherwise

Amat is cbind(**Amat**, -diag(**n**))
bvec is c(**bvec**, rep(-max.allocation, **n**))

 # Calculating the number of loops

g. Define **loops** and set it to (risk.premium.up / risk.increment + 1)
h. Define **loop** and set it to 1

 # Initializing an empty matrix that will contain optimal weights and statistics

i. Create an (**loops** x **n**+3) empty matrix and call it **eff** Hint: Use matrix(nrow=**loops**, ncol=**n**+3).

 # Renaming the columns of the matrix initialized above

j. Name the columns of **eff** as colnames(**returns**), "Std.Dev", "Exp.Return", "Sharpe". Hint: Use colnames(**eff**) and c(a, b, c, d)

USING THE QUADRATIC PROGRAM SOLVER

k. Create a vector **L** using seq() starting from 0 to risk.premium.up and increases by risk.increment.
l. Use for() to loop through **L** and for each element **i** in **L** do the following:

Define **dvec** as colMeans(returns) * **i** # This moves the solution along the EF
Obtain the solution **sol** by using the solver function i.e. solve.QP(**covariance**, **dvec**=**dvec**, **Amat**=**Amat**, **bvec**=**bvec**, **meq**=**meq**)
Define **eff**[**loop**,"Std.Dev"] as sqrt(sum(**sol**$solution*colSums((**covariance***sol$solution))))
Define **eff**[**loop**,"Exp.Return"] as as.numeric(**sol**$solution %*% colMeans(**returns**))
Define **eff**[**loop**,"Sharpe"] as **eff**[**loop**,"Exp.Return"] / **eff**[**loop**,"Std.Dev"]
Define **eff**[**loop**,1:n] as **sol**$solution
Define **loop** as **loop**+1

m. return(as.data.frame(**eff**))

Run the efficient frontier based on no short and with 50% maximum allocation restrictions

5. Obtain the efficient frontier defined as **eff** with no **short** and allow 50% allocation restrictions. Use the **eff.frontier** function and set return=returns$R, short == “no”, max.allocation = .0.5, risk.premium.up = 1, risk.increment = 0.001

The Optimal Portfolio

6. Obtain the optimal portfolio by
   a. **eff.optimal.point** = **eff**[**eff**$Sharpe==max(**eff**$Sharpe),]

Plotting the efficient frontier

```
# Color Scheme
 ealred  <- "#7D110C"
 ealtan  <- "#CDC4B6"
 eallighttan <- "#F7F6F0"
 ealdark  <- "#423C30"

ggplot(eff, aes(x=Std.Dev, y=Exp.Return)) + geom_point(alpha=.1, color=ealdark) +
  geom_point(data=eff.optimal.point, aes(x=Std.Dev, y=Exp.Return, label=Sharpe),
             color=ealred, size=5) +
  annotate(geom="text", x=eff.optimal.point$Std.Dev,
           y=eff.optimal.point$Exp.Return,
           label=paste("Risk: ",
                       round(eff.optimal.point$Std.Dev*100, digits=3),"\nReturn: ",
                       round(eff.optimal.point$Exp.Return*100, digits=4),"%\nSharpe: ",
                       round(eff.optimal.point$Sharpe*100, digits=2), "%", sep=""),
           hjust=0, vjust=1.2) +
  ggtitle("Efficient Frontier\nand Optimal Portfolio") +
  labs(x="Risk (standard deviation of portfolio)", y="Return") +
  theme(panel.background=element_rect(fill=eallighttan),
        text=element_text(color=ealdark),
        plot.title=element_text(size=24, color=ealred))
ggsave("Efficient Frontier.png")
```