Building Optimal Portfolios

David Puelz and Zack Liu

MSF and MSBA Quant Investing
March 30, 2016

1. Optimization
2. Implementing in Excel
3. Implementing in R

Optimization

Portfolios and optimal weights

1. SPY - S&P 500 ETF (Market)
2. IWM - Russel 2000 ETF (Small Cap)
3. EFA - Europe, Australia, Asia, and the Far East (Global)
4. IYR - US Real Estate ETF

Expected Returns

SPY	0.491
IWM	0.375
IYR	0.466
EFA	0.095

Covariance Matrix

	SPY	IWM	IYR	EFA
SPY	19.359	16.586	5.494	13.503
IWM	16.586	29.769	9.501	15.001
IYR	5.494	9.501	13.368	4.449
EFA	13.503	15.001	4.449	19.345

Portfolios and optimal weights

1. SPY - S&P 500 ETF (Market)
2. IWM - Russel 2000 ETF (Small Cap)
3. EFA - Europe, Australia, Asia, and the Far East (Global)
4. IYR - US Real Estate ETF

Expected Returns

SPY	0.491
IWM	0.375
IYR	0.466
EFA	0.095

Covariance Matrix

	SPY	IWM	IYR	EFA
SPY	19.359	16.586	5.494	13.503
IWM	16.586	29.769	9.501	15.001
IYR	5.494	9.501	13.368	4.449
EFA	13.503	15.001	4.449	19.345

Optimal Weights

w_SPY	0.995
w_IWM	-0.186
w_IYR	0.802
w_EFA	-0.611

What a mean-variance investor needs

Assume we have N risky assets with mean vector (ie, expected excess returns) $\boldsymbol{\mu}$ and variance covariance matrix $\boldsymbol{\Sigma}$.

$$
\begin{aligned}
\boldsymbol{\mu}_{(N\times 1)} &= \begin{bmatrix} E(r_1) \\ E(r_2) \\ \vdots \\ E(r_N) \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_N \end{bmatrix} \\
\boldsymbol{\Sigma}_{(N\times N)} &= \begin{pmatrix} var(r_1) & cov(r_1, r_2) & \cdots & cov(r_1, r_N) \\ cov(r_2, r_1) & var(r_2) & \cdots & cov(r_2, r_N) \\ \ldots & & \ldots & \\ cov(r_N, r_1) & \ldots & \ldots & var(r_N) \end{pmatrix} \\
&= \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2N} \\ \ldots & & \ldots & \\ \sigma_{N1} & \ldots & \ldots & \sigma_N^2 \end{pmatrix}
\end{aligned}
$$

From moments to an optimal portfolio

Define a portfolio by its weights, $\boldsymbol{w}_p$.

- Portfolio mean: $\mu_p = \boldsymbol{w}_p'\boldsymbol{\mu}$
- Portfolio variance: $\sigma_p^2 = \boldsymbol{w}_p'\boldsymbol{\Sigma}\boldsymbol{w}_p$

Goal: Find an optimal set of weights. We care about the *return/risk trade-off* so we will solve the following optimization problem:

$$\min_{\boldsymbol{w}}\{\boldsymbol{w}_p'\boldsymbol{\Sigma}\boldsymbol{w}_p\} \qquad \text{s.t.} \quad \boldsymbol{w}_p'\boldsymbol{\mu} = c$$

Solving for the optimal weights

Set up the Lagrangian function:

$$L = \boldsymbol{w}_p' \boldsymbol{\Sigma} \boldsymbol{w}_p - \lambda \left(\boldsymbol{w}_p' \boldsymbol{\mu} - c \right)$$

Differentiating L with respect to to $\boldsymbol{w}$ and setting it to zero leads to:

$$2\boldsymbol{\Sigma}\boldsymbol{w}_p - \lambda\boldsymbol{\mu} = 0$$

$$\Longrightarrow$$

$$\boldsymbol{w}_p = \lambda\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$$

Solving for the optimal weights

Optimal weights are given by:

$$\boldsymbol{w}_p^* \propto \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$$

We didn't enforce the weights to sum to 1, so we are assuming we can borrow at the risk-free rate.

We nail down the proportionality constant by enforcing weights to sum to 1:

$$\boldsymbol{w}_p^* = \left(\frac{1}{\mathbf{1}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}\right)\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$$

This is called the Tangency Portfolio!

An investor can minimize variance, too

In this case, the optimization problem is:

$$\min_{\boldsymbol{w}}\{\boldsymbol{w}_p'\boldsymbol{\Sigma}\boldsymbol{w}_p\} \qquad \text{s.t.} \quad \boldsymbol{w}_p'\mathbf{1}' = 1$$

Set up the Lagrangian function:

$$L = \boldsymbol{w}_p'\boldsymbol{\Sigma}\boldsymbol{w}_p - \lambda\left(\boldsymbol{w}_p'\mathbf{1}' - 1\right)$$

Differentiating L with respect to to $\boldsymbol{w}$,setting it to zero and normalizing the weights leads to:

$$2\boldsymbol{\Sigma}\boldsymbol{w}_p - \lambda\mathbf{1} = 0$$

$$\Longrightarrow$$

$$\boldsymbol{w}_p^* = \left(\frac{1}{\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1}}\right)\boldsymbol{\Sigma}^{-1}\mathbf{1}$$

Summary

Given moment estimates, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$:

- Mean-variance portfolio: $\boldsymbol{w}_p^* = \left(\frac{1}{\mathbf{1}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}\right)\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$
- Minimum-variance portfolio: $\boldsymbol{w}_p^* = \left(\frac{1}{\mathbf{1}'\boldsymbol{\Sigma}^{-1}\mathbf{1}}\right)\boldsymbol{\Sigma}^{-1}\mathbf{1}$
- Long-only versions of these portfolios need to be solved for numerically - there is not a nice solution :(

Implementing in Excel

Optimal Portfolio Weights in Excel

1. SPY - S&P 500 ETF (Market)
2. IWM - Russel 2000 ETF (Small Cap)
3. EFA - Europe, Australia, Asia, and the Far East (Global)
4. IYR - US Real Estate ETF

Given 10 years of monthly returns, how do we find optimal portfolio weights for the mean-variance efficient (min variance) portfolio?

FUNCTIONS: MMULT, TRANSPOSE, MINVERSE

Process

1. Calculate expected excess returns of each portfolio.
   =AVERAGE(B2:B121)

2. Calculate covariance matrix
   $\Sigma = E(X * X') - \mu * \mu'$
   =(MMULT(TRANSPOSE(B2:E121),B2:E121)/120)-MMULT(TRANSPOSE(H3:H6),H3:H6)

3. Solve for weights
   $\Sigma^{-1}\mu$ or $\Sigma^{-1}1$
   =MMULT(MINVERSE(L3:O6),H3:H6)

4. Make sure weights add up to 1

Implementing in R

Optimal 8 ETF portfolio in R

1. SPY - S&P 500 ETF (Market)
2. IWM - Russel 2000 ETF (Small Cap)
3. EFA - Europe, Australia, Asia, and the Far East (Global)
4. EEM - Emerging Markets ETF
5. EWJ - Japanese Equity ETF
6. EWU - United Kingdom Equity ETF
7. EWY - South Korean Equity ETF
8. IYR - US Real Estate ETF

Given 13 years of monthly returns, how do we find optimal portfolio weights for the mean-variance efficient (min variance) portfolio?

Estimating μ and Σ

There are **so many** ways!

- Historical: Each observation weighted equally.

  R functions: `colMeans()` and `cov()`.

- Exponential weighting: Observation t weighted with α^{T-t} for an $0 \leq \alpha \leq 1$. Here, T is the size of your rolling window.

  R functions: `cov.wt()`.

- Factor models: Assume asset returns have a factor structure. Think of the CAPM: $r_i^e = \beta r_{market}^e + \epsilon_i$

Process

1. Estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ using functions `colMeans()`, `cov()`, and `cov.wt()`.
2. Calculate optimal weights using a matrix product `%*%` or, for long only weights, the function `optim()`.
3. Renormalize weights to 1, use something like: `w/sum(w)`.
4. Rolling window? Use a `for` loop!