Chapter 5
Portfolio

O. Afonso, P. B. Vasconcelos

Computational Economics: a concise introduction
Overview

1. Introduction
2. Economic model
3. Numerical solution
4. Computational implementation
5. Numerical results and simulation
6. Highlights
7. Main references
The portfolio optimisation model, originally proposed by Markowitz (1952), selects proportions of assets to be included in a portfolio.

- To have an efficient portfolio:
  - the expected return should be maximised contingent on any given number of risks; or
  - the risk should be minimised for a given expected return.

- Thus, investors are confronted with a trade-off between expected return and risk.

- The expected return-risk relationship of efficient portfolios is represented by an efficient frontier curve.

Optimisation knowledge is required to solve this problem. Focus is on a Monte Carlo optimisation and on advanced numerical solutions provided by MATLAB/Octave.
The aim is to maximise the expected return constrained to a given risk

$$\max_{x} c^T x, \text{ s.t. } x^T H x = \sigma^2, \sum_{i=1}^{n} x_i = 1 \text{ and } x_i \geq 0,$$

(1)

where

- $n$ is the number of assets,
- $x, n \times 1$, is the vector of the shares invested in each asset $i$,
- $c, n \times 1$, is the vector of the average benefit per asset,
- $H, n \times n$, is the covariance matrix, and
- $\sigma^2$ is the expected risk goal.

Problem (1) is known as a quadratic programming problem.

Alternatively, minimise the risk subject to an expected return, $\bar{c}$,

$$\min_{x} x^T H x, \text{ s.t. } c^T x = \bar{c}, \sum_{i=1}^{n} x_i = 1 \text{ and } x_i \geq 0.$$

(2)
The **global minimum variance portfolio** is the one satisfying

$$\min_{x} x^T H x, \text{ s.t. } \sum_{i=1}^{n} x_i = 1 \text{ and } x_i \geq 0. \quad (3)$$

The **efficient frontier** is the set of pairs (risk, return) for which the returns are greater than the return provided by the minimum variance portfolio.

The aim is to find the values of variables that optimise an objective, conditional or not to constraints.

Numerical methods overcome limitations of size, but there is no universal algorithm to solve optimisation problems.

The topic is addressed only in a cursory manner exploiting the MATLAB/Octave optimisation potentialities.
Consider the minimisation problem

\[
\begin{align*}
\min_{x \in \mathbb{R}^n} & \quad f(x) \\
\text{s.t.} & \quad c_i(x) = 0, \quad i \in E \\
& \quad c_i(x) \geq 0, \quad i \in I
\end{align*}
\]

where \( f : \mathbb{R}^n \to \mathbb{R} \), \( c_E : \mathbb{R}^n \to \mathbb{R}^{nE} \) and \( c_I : \mathbb{R}^n \to \mathbb{R}^{nI} \), respectively, the equality and inequality constraints.

- A feasible region is the set points satisfying the constraints \( S = \{ x : c_i(x) = 0, \ i \in I \text{ and } c_i(x) \geq 0, \ i \in D \} \).

- Problems without restrictions \( I = D = \emptyset \) emerge in many applications and as a recast of constraint problems where restrictions are replaced by penalty terms added to the objective function.
Optimisation problems can be classified in various ways, according to, for example: (i) functions involved; (ii) type of variables used; (iii) type of restrictions considered; (iv) type of solution to be obtained; and (v) differentiability of the functions involved.

Among the countless optimisation problems, linear, quadratic and nonlinear programming are the most usual.

Many algorithms for nonlinear programming problems only seek local solutions; in particular, for convex linear programming, local solutions are global.
Unconstrained optimisation in practice

- **Unconstrained optimisation problems**
  - Methods such as steepest descent, Newton and quasi-Newton are the most used.
  - MATLAB/Octave:
    - `fminunc(f,x0)` attempts to find a local minimum of function $f$, starting at point $x_0$;
    - similarly with `fminsearch(f,x0)` but using a derivative-free method;
    - $x_0$ can be a scalar, vector, or matrix.
Constrained optimisation in practice: linear programming

- Linear programming problem: both the objective and constraints are linear

\[
\begin{align*}
\min_x & \quad c^T x \\
\text{s.t.} \quad & \quad Ax \leq b, \quad A_{eq}x = b_{eq}, \quad lb \leq x \leq ub
\end{align*}
\]

where \( c \) and \( x \) are vectors.

- MATLAB/Octave: \texttt{linprog(c, A, b, Aeq, beq, lb, ub)}. 

Constrained optimisation in practice: quadratic programming

- **Quadratic programming** problem (portfolio problem): this involves a quadratic objective function and linear constraints

\[
\begin{align*}
\min_x \quad & \frac{1}{2} x^T H x + x^T c \\
\text{s.t.} \quad & Ax \leq b, \ A_{eq} x = b_{eq}, \ lb \leq x \leq ub
\end{align*}
\]

where \(c, x\) and \(a_i\) are vectors, and \(H\) is a symmetric (Hessian) matrix.

- **MATLAB:** `quadprog(H, c, A, b, Aeq, beq, lb, ub)`
- **Octave:** `qp([], H, c, Aeq, beq, lb, ub)`
Constrained optimisation in practice: nonlinear programming

- **Nonlinear** programming: $f$ and/or constraints are nonlinear

\[
\min_x \quad f(x) \\
\text{s.t.} \quad c(x) \leq 0, \quad c_{eq}(x) = 0, \\
\quad A x \leq b, \quad A_{eq} x = b_{eq}, \quad lb \leq x \leq ub.
\]

- **MATLAB:** \texttt{fmincon}(f, x0, A, b, Aeq, beq, lb, ub, nonlcon)

- **Octave:** \texttt{minimize}(f, args) (where \texttt{args} is a list or arguments to \texttt{f})
Monte Carlo approach

- **Monte Carlo**: experiments anchored on repeated random sampling to obtain numerical approximations of the solution.

- A Monte Carlo procedure can be schematised as follows.
  1. Set a possible solution, and consider it to be the best for the moment
  2. For a certain number of times, do:
     1. generate (randomly) a set of feasible solutions from the best one available;
     2. select (possibly) a better one;
     3. repeat the process.
Consider the following data, respectively, for the returns vector and covariance matrix

\[ c = \begin{bmatrix} 0.100 \\ 0.200 \\ 0.150 \end{bmatrix} \text{ and } H = \begin{bmatrix} 0.005 & -0.010 & 0.004 \\ -0.010 & 0.040 & -0.002 \\ 0.004 & -0.002 & 0.023 \end{bmatrix} \]
Monte Carlo approach: portfolio with minimum variance

```matlab
function [x, x_hist] = portfolio_mcarlo_fun(H, nruns, const)

% Monte Carlo solution approach
% Implemented by: P.B. Vasconcelos and O. Afonso
% based on: Computational Economics,
% D. A. Kendrick, P. R. Mercado and H. M. Amman
% Princeton University Press, 2006
% input:
%   H, covariance matrix
%   nruns, number of Monte Carlo runs
%   const, constant to increase/reduce the magnitude of the random
%   numbers generated
% output:
%   x, best found portfolio
%   x_hist, search history for best portfolio
```
% initialization parameters and weights;
popsize = 10;
n = size(H,1);
pwm = (1/n)*ones(n,popsize);
crit = zeros(1,popsize);
x_hist = zeros(n,1);

% compute nruns x popsize portfolios
for k = 1:nruns
    for j = 1:popsize;
        crit(j) = pwm(:,j)'*H*pwm(:,j);
    end

% selection of the best portfolio
[~, top_index] = min(crit);
x = pwm(:,top_index);

% store the best portfolio
x_hist(:,k) = x;
if k == nruns, break, end
pwm(:,1) = x;
for i = 2:popsize;
    x = x + randn(n,1)*const;
    pwm(:,i) = abs(x/sum(abs(x)));
end
end

To solve the problem just do:

nruns = 40;
const = 0.1;
[x,x_hist] = portfolio_mcarlo_fun(H,nruns,const);
disp('best portfolio:');
for i=1:length(x)
    fprintf('Asset %d \t %5.4f \n',i,x(i));
end
fprintf('expected return: %g \n',c'*x);
fprintf('risk : %g \n',sqrt(x' *H*x));
Portfolio optimization: global minimum variance
    Monte Carlo solution approach

best portfolio:
Asset 1  0.7649
Asset 2  0.2355
Asset 3  0.0004

expected return: 0.123644
risk     : 0.0392812
Monte Carlo convergence path for the portfolio with minimum variance
Quadratic programming approach: portfolio with minimum variance

\[ A_{eq} = \text{ones}(1, \text{length}(c)); \quad \text{beq} = 1; \quad \text{lb} = \text{zeros}(1, \text{length}(c)); \]
\[ x = \text{quadprog}(2*H, [], [], [], A_{eq}, \text{beq}, \text{lb}); \]
\[ \text{disp('best portfolio:')} ; \]
\[ \text{for } i=1: \text{length}(x) \]
\[ \quad \text{fprintf('Asset %d \t %5.4f \n', i, x(i));} \]
\[ \text{end} \]
\[ \text{fprintf('expected return: %g \n', c' \times x);} \]
\[ \text{fprintf('risk : %g \n', sqrt(x' \times H \times x));} \]
Portfolio optimization: global minimum variance
quadratic programming approach

best portfolio:
Asset 1  0.7692
Asset 2  0.2308
Asset 3  0.0000
expected return: 0.123077
risk        : 0.0392232
The portfolio optimisation model selects the optimal proportions of various assets to be included in a portfolio, according to certain criteria.

A rational investor aims at choosing a set of assets (diversification) delivering collectively the lowest risk for a target expected return.

A portfolio is considered efficient if it is not possible to obtain a higher return without increasing the risk.

The expected return-risk relationship of efficient portfolios is represented by an efficient frontier curve.

The model is a quadratic programming problem. It is solved by using a simple Monte Carlo approach that only requires the notion of a minimum conditioned to a set of restrictions, and by a more sophisticated method deployed by MATLAB/Octave.
References

R. A. Haugen and N. L. Baker
The efficient market inefficiency of capitalization-weighted stock portfolios

H. Markowitz
Portfolio selection

R. C. Merton
An analytic derivation of the efficient portfolio frontier

J. Nocedal and S. J. Wright
Numerical Optimization
*Springer (2006)*

D. Pachamanova and F. J. Fabozzi
Simulation and optimization in finance: modeling with MATLAB, @RISK, or VBA
*vol. 173, John Wiley & Sons (2010)*