# Modern Portfolio Theory

MEIF Quant training document (internal)

## 1. Introduction

Modern Portfolio Theory is a portfolio optimisation technique which can be used to optimise asset allocation weights. There are 2 main principles of which you can optimise your asset allocation weightings; 1. minimum portfolio variance, 2.  maximising Sharpe ratio.

The remainder of this document will discuss the mathematical definitions of minimum variance portfolios, and maximising Sharpe ratio portfolios. And the methods for solving convex/quadratic optimisation problems.

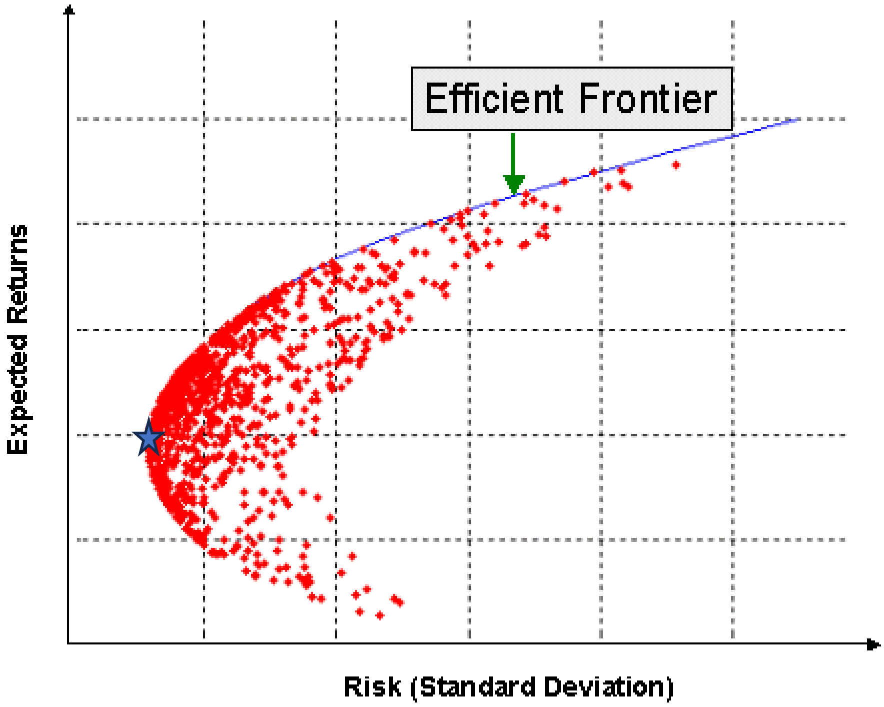
**Prerequisite knowledge**

- Linear algebra
- Quadratic programming

**References and further resources**

- Convex optimisation – Boyd

## 2. Minimum Portfolio Variance



Firstly, the objective function for a minimum portfolio variance must be defined, subject to certain constraints.

Portfolio Variance is defined through via $Portfolio\ Variance = \ W^T\Sigma W$.

Where, $W \in \mathbb{R}^n, \Sigma\ is\ the\ covariance\ matrix\ with\ \Sigma \in \mathbb{R}^{n\times n}$.

Subject to, the sum of our portfolio weights must equal to 1, as 1 represents the total capital available to us, and this being a long only portfolio (hence each weighting must be greater than 0).

$$s.t. \quad \sum_{i=1}^{n} w_i = 1$$

$$s.t. \quad w_i \geq 0 \quad\ i = 1, \dots, n$$

Therefore, the following defines our minimisation problem, subject to certain constraints.

$$\min_{w} W^T\Sigma W$$

$$s.t. \quad \sum_{i=1}^{n} w_i = 1$$

$$s.t. \quad w_i \geq 0 \quad\ i = 1, \dots, n$$

This now needs to be re-written as a loss function, where the aim is to minimize its' value.

One of the 'subject to' constraints is an inequality constraint. This is not an exact mathematical definition, and there are various methods of solving, one involves introducing slack variables to convert the inequality constraint into an equality constraint. However, in the case of this report, log barrier functions will be used to represent the inequality constraints, hence turning the problem into an unconstrained optimisation problem.

The log barrier function helps solve the minimisation problem, as when the inequality constraint is approached, e.g. a weight tends towards 0, the loss function tends towards positive infinity, hence, not minimising the loss function. In other words, the log barrier function can be considered as a penalty function (note: people may disagree with this statement so be careful in interview).

Therefore, the problem can be redefined as:

$$\min_{w} \quad W^T\Sigma W + \lambda(W^T\mathbf{1} - 1) - \mu \sum_{i=1}^{n} \ln(w_i) \qquad Eq\ 1$$

Where, $\lambda$ is a lagrangian multiplier, and $\mu$ is the stiffness with respect to the log barrier function (smaller value $\mu$ allows for a more accurate solution but convergence period increases).

Eq 1 returns a scalar quantity.

Proceed onto section 2.1, and 2.2 for the methods to solve the minimisation problem.

## 2.1 Gradient Descent with Backtracking Line Search

This approach involves the objective function being first order differentiable.

To proceed, the gradient of the objective function must be defined.

$$\emptyset = 2\Sigma W + \ \lambda\mathbf{1}^T - -\mu\frac{1}{W}$$

Eq 2 now returns a vector, with $\emptyset \in \mathbb{R}^n$.

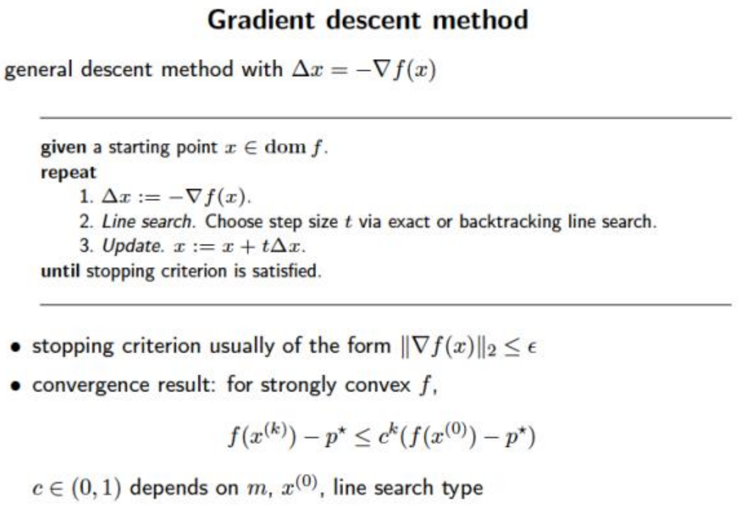
The procedure then continues as follows (view Figure 2.1).

Therefore, in this case, $\Delta w = -\emptyset$.

Additionally, the stopping criterion is the L2 norm of the gradient vector being less than or equal to a pre-defined tolerance.

The stopping criterion in this case is therefore defined as: $\sqrt{\sum_{i=1}^{n} {\emptyset_i}^2} \leq \epsilon$ (alternative stopping criterions are available, but this is standard).

Initial conditions, $\mu = 0.1$, and W must be selected with respect to number of assets which you are trying to optimise for. $W_{inital} = 1/n$, where n is the number of assets you are optimising for.



**Gradient descent method**

general descent method with $\Delta x = -\nabla f(x)$

```
given a starting point $x \in \mathbf{dom}\, f$.
repeat
    1. $\Delta x := -\nabla f(x)$.
    2. Line search. Choose step size $t$ via exact or backtracking line search.
    3. Update. $x := x + t\Delta x$.
until stopping criterion is satisfied.
```

- stopping criterion usually of the form $\|\nabla f(x)\|_2 \leq \epsilon$
- convergence result: for strongly convex $f$,

$$f(x^{(k)}) - p^\star \leq c^k (f(x^{(0)}) - p^\star)$$

$c \in (0,1)$ depends on $m$, $x^{(0)}$, line search type

*Figure 2.1*

This procedure is iteratively performed until the stopping criterion is met, with the solution being the final value of $W$, which is a vector, informing you of what weights you should select to minimise your portfolio variance subject to certain constraints.

## 2.2 Barrier Method with Newton Method

Barrier method is significantly quicker at convergence for convex optimisation problems. A key difference between gradient descent and barrier method, is that gradient descent is a first order method, making use of the first derivative of the objective function. Barrier method however takes advantage of the Hessian, which is the second order derivative of the objective function. Therefore, for barrier method to be employed, our objective function must be second order differentiable.

Barrier method is primarily made up from Newton method.

From section 2.1, the first order derivative has been solved for, therefore this document will continue straight onto the second order derivative of the objective function (known as the Hessian).

$$\nabla^2 f(w) = 2\Sigma - \mu Diag(w_i)$$

$$\nabla^2 f(w) \in \mathbb{R}^{n \times n}$$

With the Hessian calculated, the next step is to calculate the newton step (how much the weights will be updated by with each iteration) (Newton step = $\Delta_{newt}$).

$$\Delta_{newt} \coloneqq -\nabla^2 f(w)^{-1} \nabla f(w)$$

Continuing with undamped newton method.

Prior to updating the weights, a stopping criterion should be checked, which follows:

- First compute $\lambda \coloneqq \nabla f(w)^T \nabla^2 f(w)^{-1} \nabla f(w)$
- Check and stop if: $\lambda \leq User\ specified\ tolerance$

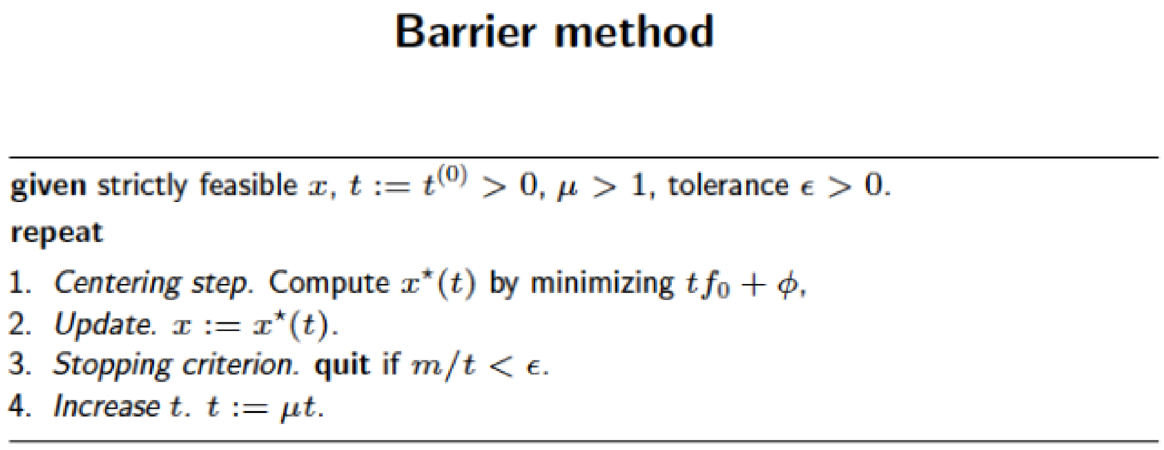
If Newton's method has not stopped due to the avoid stopping criterion, update the weights.

$$W \coloneqq W + \Delta_{newt}$$

(If the algorithm was damped Newton's method, then the Newton Step would be multiplied to a step size value, which could have been computed via backtracking line search).

The above steps have been regarded an 'Newton Method'. But, for barrier method, newton's method is nested within the barrier method algorithm and iteratively computed. Therefore, in a single barrier method optimisation code, Newton's method will be solved for several times. Hence, the centering is computed by Newton's method.

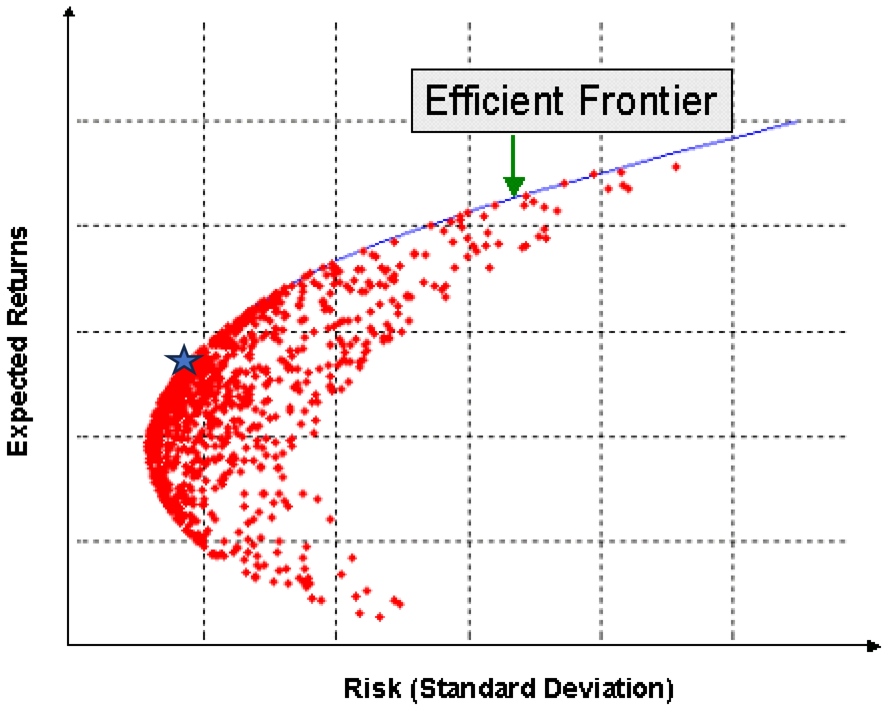
The methodology behind barrier method is simply to compute serval iterations. Whereby the stopping criterion tolerance is user specified, but it is often written is the following form:



## Barrier method

```
given strictly feasible $x$, $t := t^{(0)} > 0$, $\mu > 1$, tolerance $\epsilon > 0$.
repeat
    1. Centering step. Compute $x^\star(t)$ by minimizing $tf_0 + \phi$,
    2. Update. $x := x^\star(t)$.
    3. Stopping criterion. quit if $m/t < \epsilon$.
    4. Increase $t$. $t := \mu t$.
```

## 3. Maximising Sharpe Ratio



Defining the Sharpe Ratio for a single security:

$$Sharpe\ Ratio = \frac{Return - Risk\ Free\ Rate}{Volatility\ of\ security}$$

As we are working with a portfolio of securities (e.g. a portfolio of stocks), the Sharpe Ratio must be defined to represent the whole portfolio of securities, this is defined through linear algebra for simplicity.

$$Sharpe\ Ratio = \frac{W^T R - Risk\ Free\ Rate}{\sqrt{W^T \Sigma W}}$$

Where,

- $W$ is the vector of weights (where the weights represent how much we should invest in each of the securities – currently unknow, but our methods will solve for).
- $R$ is the vector of expected returns for each of the corresponding securities which we are analysing for.
- $Risk\ Free\ Rate$ is simply just a scalar (known).
- $\Sigma$ is the covariance of the securities (known).

In Section 3, the objective is the maximise the Sharpe ratio, hence this is a maximisation problem, solving for $W$. However, for portfolio optimisation, we will encounter certain equality, and inequality constraints, as a result we must maximise the Sharpe ratio subject to given constraints.

In this example, the following limitations will be put on the portfolio:

1. No leveraging is available for this portfolio, this is mathematically defined through the sum of the investment weights for each of the individual securities must be equal to or less than 1, less than 1, because due to limitation 2 below states the maximum investment into a single asset is 25% of our portfolio equity, and if we only have 3 equities, then the maximum equity which we can invest is 75% (hence):
$$\sum_{i=1}^{n} w_i \leq 1.0$$
2. No shorting positions are allowed, and the maximum investment from a single security is with the upper limit of 25% of our total portfolio equity.
$$0.0 \leq w_i \leq 0.25$$

Now, with the Sharpe ration and portfolio constraints mathematically defined, the problem at hand can be written as:

$$\max_{w} \frac{W^T R - Risk\ Free\ Rate}{\sqrt{W^T \Sigma W}}$$

$$s.t.$$

$$\sum_{i=1}^{n} w_i \leq 1.0$$

$$0.0 \leq w_i \leq 0.25$$

Next, we would like to mathematically define this as a single objective function. (In this example, I will continue to make this problem a minimisation problem, but I will correspondingly invert the objective function, which will still return the optimal investment weights).

$$\min_{w} - \frac{W^T R - Risk\ Free\ Rate}{\sqrt{W^T \Sigma W}}$$

In Section 2, the use of log barrier functions was introduced to represent the inequality constraints, simply due to the fact that inequality constraints do not have an explicit mathematical definition, and therefore we must attempt to approximate this inequality constraint (using log barriers). Due to this, the problem can now be considered as an 'unconstrained convex optimisation problem'. Remember, as we are now minimising the function, when our inequality constraints are approached, the negative log barrier tends toward infinity, which is counteracting our goal of minimisation.

Therefore, the function which we are trying to minimise for, along with the constraints can now be defined as:

$$\min_{w} - \frac{W^T R - r_f}{\sqrt{W^T \Sigma W}} - \mu(\sum_{i=1}^{n} (\ln(w_i) + \ln(0.25 - w_i)) + \ln{(1 - W^T \mathbf{1})}$$

In this case, $\mu$ determines the stiffness of our minimisation process with respect to the constraints. The smaller $\mu$, as solution closer to the true solution will be achieved (but computational time will correspondingly increase), and vice versa.

### 3.1 Gradient Descent with Backtracking Line Search

Firstly, taking the gradient of the objective function, with respect to the weights.

$$\emptyset = R(W^T\Sigma W)^{-\frac{1}{2}} - \left(W^TR - r_f\right)(W^T\Sigma W)^{-\frac{3}{2}}\Sigma W - \mu(\sum_{i=1}^{n}\left(\frac{1}{w_i} + \frac{1}{0.25 - w_i}\right) + \frac{1}{1 - W^T\mathbf{1}} \times -\mathbf{1}^T)$$

$$\emptyset \in \mathbb{R}^n$$

In this case, $\Delta w = -\emptyset$.

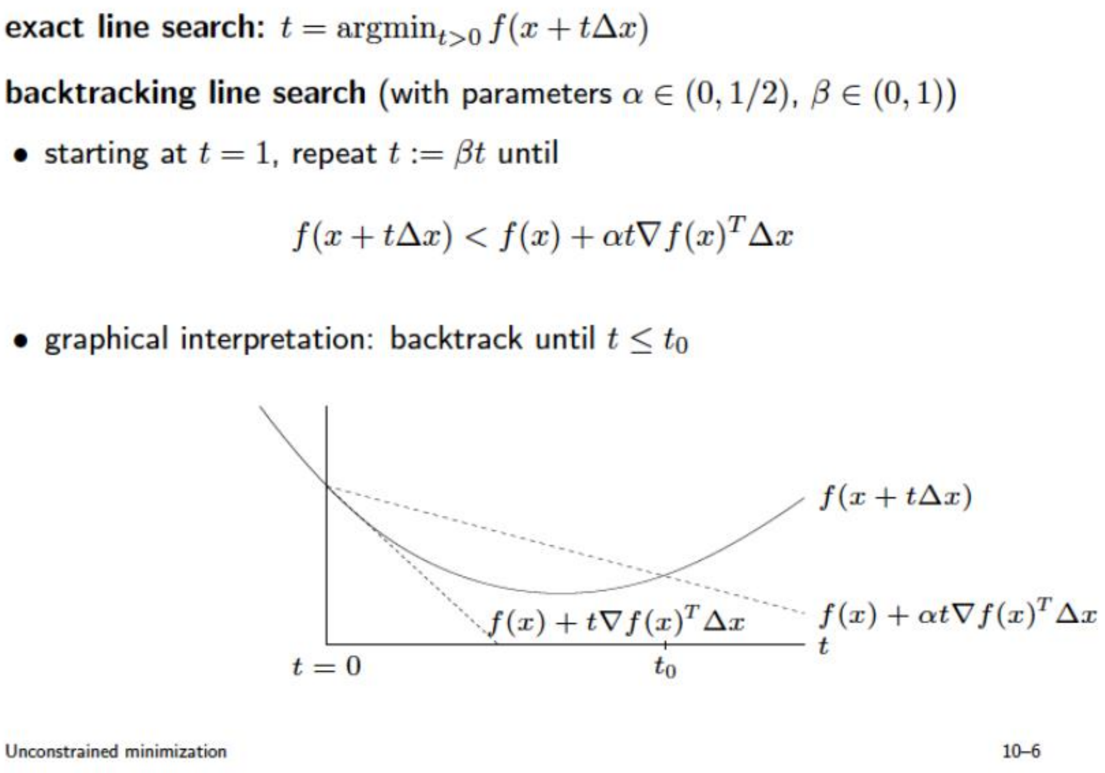
Therefore, we will perform an iterative procedure of gradient descent, following a similar process to finding the minimisation point (x coordinate) of a quadratic curve, which is covered in A-level, but we are applying this to a multi-dimensional optimisation problem, where we have n securities which we are analysing for.

At each step $W^k$, we will calculate the corresponding gradient, $\nabla w$, which is the change in each of the security weights, as we are descending down the gradient our step length is $\Delta w = -\emptyset$ (negative due to the descent).

Therefore, $W^{k+1} = W^k - t\emptyset$ , where t=1 if no backtracking line search is present.

As we would like to include backtracking line search, to reduce the number of gradient descent steps, but many iterations may be performed to calculate an appropriate step size.

Backtracking line search is performed through the following steps:



**exact line search:** $t = \operatorname{argmin}_{t>0} f(x + t\Delta x)$

**backtracking line search** (with parameters $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$)

- starting at $t = 1$, repeat $t := \beta t$ until

$$f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$$

- graphical interpretation: backtrack until $t \leq t_0$

$f(x + t\Delta x)$
$f(x) + t\nabla f(x)^T\Delta x$
$f(x) + \alpha t\nabla f(x)^T\Delta x$
$t$
$t = 0$
$t_0$

Unconstrained minimization 10–6