Modern Portfolio Theory:

This theory suggests an optimal portfolio made up of given securities by analyzing their risk and return, minimizing risk and maximizing return.

NOTE: Coming up with a solution for maximized return on a given variance was more of a mathematical challenge than intended for practical uses. Most of the time, this comes up with a solution that is not practically feasible (for example, shorting 900% of your portfolio's worth is pointless). There are better ways to come up with better portfolio optimizations.

Notation:

n: Scalar denoting the number of Securities being analyzed.

s: Scalar denoting the desired variance of the portfolio.

r: a vector of returns of size n, i^{th} entry denoting the return for i^{th} security.

w: the weight vector of size n, this has the weight for every security and is the vector we need to optimize.

$\boldsymbol{\Sigma}$: covariance matrix for size n × n. $\boldsymbol{\Sigma}_{ij} = \boldsymbol{\Sigma}_{ji}$ = covariance between the i^{th} and j^{th} security.

E(r): The expected return of the portfolio, to be maximized for a given s. Given by $\mathbf{w}^T\mathbf{R}$.

Variance = $\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}$

Problem formulation:

We have n securities, we have their returns, risk factors, and covariances. We need to find an optimal weight for each security. The sum has to be equal to 100% (100% is all of the portfolio net worth). This can be written mathematically as:

Maximize **E**(r)

Subject to

$\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w} = s$ (variance should be s)

$\mathbf{w}^T\mathbf{I} = 1$ (weights should sum up to 1)

This can be summarized in one Lagrange equation as:

$\max_{\alpha 1, \alpha 2}\min_{\mathbf{w}} L(w, \alpha_1, \alpha_2) = \mathbf{w}^T\mathbf{R} - \alpha_1(\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w} - s) - \alpha_2(\mathbf{w}^T\mathbf{I} - 1)$

Since the gradients are not linear equations, the solution becomes rather tedious to compute. We proceed forward with gradient descent to find the optimal solution.

Solution:

We start by finding gradients of the Lagrange w.r.t the variables to be optimized:

$\nabla_w L(w, \alpha_1, \alpha_2) = \mathbf{R} - 2\alpha_1 \boldsymbol{\Sigma}\mathbf{w} - \boldsymbol{\alpha}_2$ ($\boldsymbol{\alpha}_2$: vector with all values equal to α_2)

$\nabla_{\alpha 1} L(w, \alpha_1 \alpha_2) = \mathbf{w}^T \boldsymbol{\Sigma}\mathbf{w} - s$

$\nabla_{\alpha 2} L(w, \alpha_1, \alpha_2) = \mathbf{w}^T\mathbf{I} - 1$

Now, the problem formulation requires us to simultaneously minimize and maximize different variables. Every iteration of the gradient descent, we can find w by equating gradient to 0 and convert it into a maximization problem instead.

$\nabla_w L(w, \alpha_1, \alpha_2) = 0$

$\Rightarrow \mathbf{R} - 2\alpha_1 \boldsymbol{\Sigma}\mathbf{w} - \boldsymbol{\alpha}_2 = 0$

$\Rightarrow w = (2\alpha_1 \boldsymbol{\Sigma})^{-1}(R - \boldsymbol{\alpha}_2)$

Thus, the problem now is simply calculating **w** every iteration then applying gradient ascent on the Lagrange multipliers α_1 and α_2, simultaneously:

$\alpha_1 := \alpha_1 + \in (\mathbf{w}^T \boldsymbol{\Sigma}\mathbf{w} - s)$

$\alpha_2 := \alpha_2 + \in (\mathbf{w}^T\mathbf{I} - 1)$

The final algorithm is listed below.

The algorithm:

-Randomly select lambda1 and lambda2 (corresponding to α_1 and α_2)

-Choose the number of iteration (for n=5, around 100k is like a good number)

-Choose epsilon, the learning rate (around 1e-4)

-loop for number of iterators:

w = (2 * lambda1 * cov)$^{-1}$(R – lambda2)

lambda1 := lambda1 + epsilon*(w^T*cov*w – s)

lambda2 := lambda2 + epsilon* (sum(w) – 1)

if w^T*cov*w ≈ s and sum(w) ≈ 1

break

end loop