Project Report on

Portfolio Optimization

Submitted by

Atche Sravya - 170001013

B Rushya Sree Reddy - 170001014

Under the guidance of

Dr. Kapil Ahuja

Department of Computer Science and Engineering

Indian Institute of Technology Indore

Autumn 2019

Introduction

One of the problems that most of the investors are struggling with is to use the best combination of risk and return to yield the best diversification of the portfolio. Diversification by itself is not able to solve the problem. A few investors have a specific amount of capital to invest. Some of them need their portfolios to contain specific assets. There is always a need for optimized diversification. This optimization can come true using different methods which can find the best combination of assets to meet the investor's goals.

Portfolio optimization is the process of selecting the best portfolio (asset distribution), out of the set of all portfolios being considered, according to some objective. The objective typically maximizes factors such as expected return, and minimizes costs like financial risk.

Goals

- To study the utility of Markowitz model for portfolio optimization.
- To examine and analyse the relation between risk and return.
- To select optimal portfolio.
- To study and implement interior point method to solve quadratic programming problem in portfolio optimization.

Terminology used:

Let there be N number of assets and the return value of each asset for the previous T number of years is collected.

Mean:

The expected return of each asset is the average of the return values of previous years.

$$r'_i = \left(\sum_{t=1}^{T} r_{i,t}\right) / T$$

$$0 <= t <= T \quad \text{and} \quad 0 <= i <= N$$

Variance:

The variance (σ_i^2) is a measure of how far each value in the data set is from the mea

$$\sigma_i^2 = (\sum^T_{t=1} (r_{i,t} - r_i')^2)/T$$

$$0 <= t <= T$$

$$0 <= i <= N$$

Covariance:

Covariance is a measure of how returns of two assets move together. It is a statistical measure that indicates the interactive risk of an asset relative to others in a portfolio of assets.

$$cov_{i,j} = (\sum^T_{t=1} (r_{i,t} - r_i')(r_{j,t} - r_j')) / (N-1)$$

$$1 <= i,j <= N$$

$$1 <= t <= T$$

Return:

Return of a portfolio of assets is simply the weighted average of the individual asset values held in the portfolio.

$$R_p = (\sum^T_{t=1} (x_i * r_i')) / N$$

x_i = proportion of funds invested in asset i.

r_i' = Expected return of asset i.

Risk:

Risk is determined by both the covariance of return values of each asset with one another and the variance among its own return values.

$$\sigma_p = (\sum^N_{i=1} x_i^2 * \sigma_i^2 + \sum^N_{i=1} \sum^N_{j=1} x_i * x_j * cov_{i,j})^{½}$$

x_i , x_j = proportion of funds invested in assets i , j.

$Cov_{i,j}$ = covariance between assets i and j.

σ_i^2 = variance of asset i.

N = Number of assets.

Optimization using Markowitz method

In this method, we construct a portfolio optimization formulation with the mean variance analysis framework. In statistical terms, investments are described in terms of their expected long-term return rate and their expected short-term volatility(risk).

Markowitz Portfolio selections are obtained by solving the portfolio optimization problems to get maximum total returns, constrained by minimum allowable risk level.

Markowitz Formulation:

Expected return of the portfolio **=** $\sum^{N}_{i=1} x_i * r_i$

Portfolio return variance = $(\sum^{N}_{i=1} x_i^2 * \sigma_i^2 + \sum^{N}_{i=1} \sum^{N}_{j=1} x_i * x_j * cov_{i,j})$

This portfolio variance Q can be represented as the matrix :

$\sigma^2_{1,1}$	$cov_{1,2}$	. . .	$cov_{1,n-1}$	$cov_{1,n}$
$cov_{2,1}$	$\sigma^2_{2,2}$	. . .	$cov_{2,n-1}$	$cov_{2,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$cov_{n-1,1}$	$cov_{n-1,2}$	. . .	$\sigma^2_{n-1,n-1}$	$cov_{n-1,n}$
$cov_{n,1}$	$cov_{n,2}$	. . .	$cov_{n,n-1}$	$\sigma^2_{n,n}$

Quadratic programming Problem

The general quadratic programming problem formulation contains a quadratic objective function and linear equality and inequality constraints.

$$\text{Min } f(x) = \tfrac{1}{2} x^T Q x + d^T x$$

Subjected to $A_{eq}x = b_{eq}$ (linear equality constraints)

$Ax <= b$ (linear inequality constraints)

$l <= x <= b$ (lower and upper bounds of x)

The **Markowitz optimisation problem** can be represented as

$\min \sigma_p^2 = x^T Q x$ objective function

subject to $\sum_{i=1}^{N} x_i * r_i = r_p$ return constraint

$\sum_{i=1}^{N} x_i = 1$ budget constraint

$0 \leq x_i \leq 1, \forall i = 1 \text{ to } N$ long-only constraint

Where,

Q = the covariance matrix for the returns on the assets in the portfolio(which is a semi positive definite matrix)

x = Vector of portfolio weights.

r_i = Return value of asset i.

R_p = Return value of the portfolio.

Comparing Markowitz formulation of portfolio optimization and Quadratic programming problem, we can say that the techniques to solve Quadratic programming problem can also be used for Markowitz formulation of portfolio optimization.

Interior Point Optimisation for Quadratic Programming

Consider a Non linear programming problem

$$\min_{x} \ f(x)$$

subjected to $g_i(x) >= 0$ $\quad i=1,2,3.....,m1$

$h_i(x) = 0$ $\quad j=1,2,3,......,m2$

$x >= 0$

In integer point programming we iteratively approach the optimal solution from the interior of the feasible set.

We use the **barrier functions** to force the iterates to remain in the feasible region.

Barrier Method:

For the barrier method algorithm, there a few approximations that must be made. Given a problem we must reformulate it to implicitly include the inequalities in the objective function. We can do this by creating a function that greatly increases the objective if a constraint is not met.

A well known barrier function is the logarithmic barrier function.

$$\mathbf{B(x,\mu) = f(x) - \mu\left(\sum^{m1}_{i=1} \log(g_i(x)) + \sum^{N}_{i=1} \log x_i \right)}$$

μ = Barrier parameter.

Here, logarithmic functions $\log(g_i(x))$ and $\log x_i$ are defined at points x for which $g_i(x)>0$ and $x_k>0$ $\forall$i=1 to m and $\forall$k=1 to n

Significance of barrier function:

As μ approaches 0 , the approximation becomes closer to the function f(x). Also, for any value of μ, if any of the constraints is violated, the value of the barrier approaches infinity.

The quadratic programming problem is

$$\mathbf{Min_x\ B(x,\mu)}$$

$$\mathbf{Subject\ to\ h_j(x) = 0,\ j = 0,1,\ldots,m2}$$

To find an optimal solution x_μ of $(QP)_\mu$ for a fixed value of the barrier parameter μ

Lagrange function of QP_μ

$$L_\mu(x,\lambda) = f(x) - \mu\left(\sum^{m1}_{i=1} \log(g_i(x)) + \sum^{N}_{i=1} \log x_i \right) - \sum^{m2}_{i=1} \lambda_j h_j(x)$$

for a given μ, a vector x_μ is a minimum point of QP_μ if there is a Lagrange parameter λ_μ such that, the pair (x_μ, λ_μ) satisfies:

(KKT conditions)

$$\nabla_\lambda L_\mu(x, \lambda) = 0$$

$$\nabla_x L_\mu(x, \lambda) = 0$$

$\nabla_\lambda L_\mu(x, \lambda) = -h(x)$

$$\nabla_x L_\mu(x, \lambda) = \nabla f(x) - \mu \left(\sum^{m1}_{i=1} (1/g_i(x)) (\nabla g_i(x)) + \sum^{m1}_{i=1} (1/x_i) (e_l) \right) + \sum^{m}_{j=1} \lambda_j \nabla h_j(x)$$

So we need to solve the following equations :

$$-h(x) = 0$$

$$\nabla f(x) - \mu \left(\sum^{m1}_{i=1} (1/g_i(x)) (\nabla g_i(x)) + \sum^{m1}_{i=1} (1/x_i) (e_l) \right) + \sum^{m}_{j=1} \lambda_j \nabla h_j(x) = 0$$

This system is solved using **Newton's method.**

So, if

$$F_\mu(x, \lambda)^T = \left[\ h(x) \qquad \nabla f(x) - \mu \left(\sum^{m1}_{i=1} (1/g_i(x)) (\nabla g_i(x)) + \sum^{m1}_{i=1} (1/x_i) (e_l) \right) + \sum^{m}_{j=1} \lambda_j \nabla h_j(x)\ \right]$$

Algorithm

Step 0 : choose $(x0,\lambda)$

Step 1 :

1. find $(p_x^k , p^k_\lambda) = d$ by solving the linear system

$$J_{F\mu} (x^k, \lambda^k)\, d = -F_\mu(x^k, \lambda^k)$$

Since in Newton's method search direction p^k is given by $-Hess^{-1}(f(x)).grad(f(x))$

2. Determine a step length α_k
3. Set $x^{k+1} = x^k + \alpha^k p^k_x$ and $\lambda^{k+1} = x^k + \alpha^k p^k_\lambda$

Stop : if convergence is achieved

else repeat **step 1**.

Matrix $J_{F\mu} (x_k, \lambda_k)$ is given by

$J_{F\mu} (x_k, \lambda_k)$ =

$J_h(x)$	0
$H(x) - \mu\ (\sum^{m1}_{i=1} (1/g_i(x)) (\nabla g_i(x) \nabla g_i(x)^T + G_i(x)) - \sum^{m1}_{i=1} (1/x_i^2) (e_l)\ ] + \sum^{m2}_{j=1} \lambda_j \nabla F_j(x)$	$[J_h(x)]^T$

Where $H(x)$ is the Hessian matrix of $f(x)$

$J_h(x)$ is the Jacobian matrix of $h(x)^T$

$G_h(x)$ is the Hessian matrix of $g_i(x)$

$F_j(x)$ is the Hessian matrix of $h_j(x)$

- For each give μ, the above algorithm can provide a minimal point x_μ for the problem QP_μ.
- Choose a sequence $\{\mu_k \}$ of decreasing, sufficiently small non-negative barrier parameter values to obtain associated sequence $\{x_\mu^k \}$ optimal solutions of QP_μ^k .
- The solutions x_μ^k converge to a solution x^* of NLP

i.e. $\mathbf{\lim_{\mu \to 0} x_\mu = x^*}$ **.**

Primal-Dual Interior point Algorithm:

Consider a non linear programming problem:

$$\min_{x} \; f(x)$$

$$\text{subjected to} \quad g_i(x) >= 0 \qquad i = 1 \text{ to } m$$

We now introduce slack variables to turn all inequalities into non-negativities:

$s^T = [\, s_1 \quad s_2 \quad s_3 \quad ..... \quad s_m \,]$

$$\min_{x} \; f(x)$$

$$\text{subjected to} \quad g(x) - s = 0$$

$$s >= 0$$

Using the logarithmic barrier function for this problem we get

$$\min_{x} \; f(x) - \mu \sum_{i=1}^{m} \log(s_i)$$

$$\text{subjected to} \quad g(x) - s = 0$$

The lagrange function for this optimization problem is given by

$$L_\mu(x,\lambda) = f(x) - \mu \sum_{i=1}^{m} \log(s_i) - \sum_{i=1}^{m} \lambda_j \,(\, g_i(x) - s_i \,)$$

$$= f(x) - \mu \sum_{i=1}^{m} \log(s_i) - \lambda^T (\, g(x) - s)$$

Now using KKT conditions and setting all the derivatives to zero:

$$\nabla_x L(x, \lambda) = \nabla f(x) - \nabla g(x)^T \lambda = 0$$

$$\nabla_\lambda L(x, \lambda) = -\mu S^{-1} e + \lambda = 0 = \mu e - S \lambda e$$

$$\nabla_s L(x, \lambda) = g(x) - s = 0$$

$e^T = [1\ 1\ 1\ 1 \ldots.\ 1]$ size = m

$$S = \begin{matrix} 1/s_1 & 0 & 0 & \ldots\ldots\ldots & 0 \\ 0 & 1/s_2 & 0 & & 0 \\ : & : & : & & : \\ 0 & 0 & 0 & & 1/s_m \end{matrix}$$

S is an mxm diagonal matrix

Now using Newton's method to determine the search directions p_x, p_s, p_y

Since in Newton's method search direction p^k is given by -Hess^{-1}(f(x)).grad(f(x))

$$\left(\nabla^2 f(x) - \sum^m_{i=1} \lambda_i \nabla^2 g_i(x) \right) p_x - \nabla^T g(x)\ p_\lambda = -\left(\nabla f(x) - \nabla g(x)^T \lambda \right)$$

$$\lambda p_s + S p_y = -(\mu e - S \lambda e)$$

$$\nabla g(x)\ p_x - I p_s = -(g(x) - s)$$

By solving these equations we get the search directions p_x, p_s, p_y

Algorithm:

Step 0 : choose (x_0, s_0, λ_0)

Step 1 :

1. find $(p_x^k, p_s^k, p^k_\lambda)$ by solving the equations given above
2. Determine a step length α_k
3. Set $x^{k+1} = x^k + \alpha^k p^k_x$

   $s^{k+1} = s^k + \alpha^k p^k_s$

   $\lambda^{k+1} = x^k + \alpha^k p^k_\lambda$

Stop : if convergence is achieved

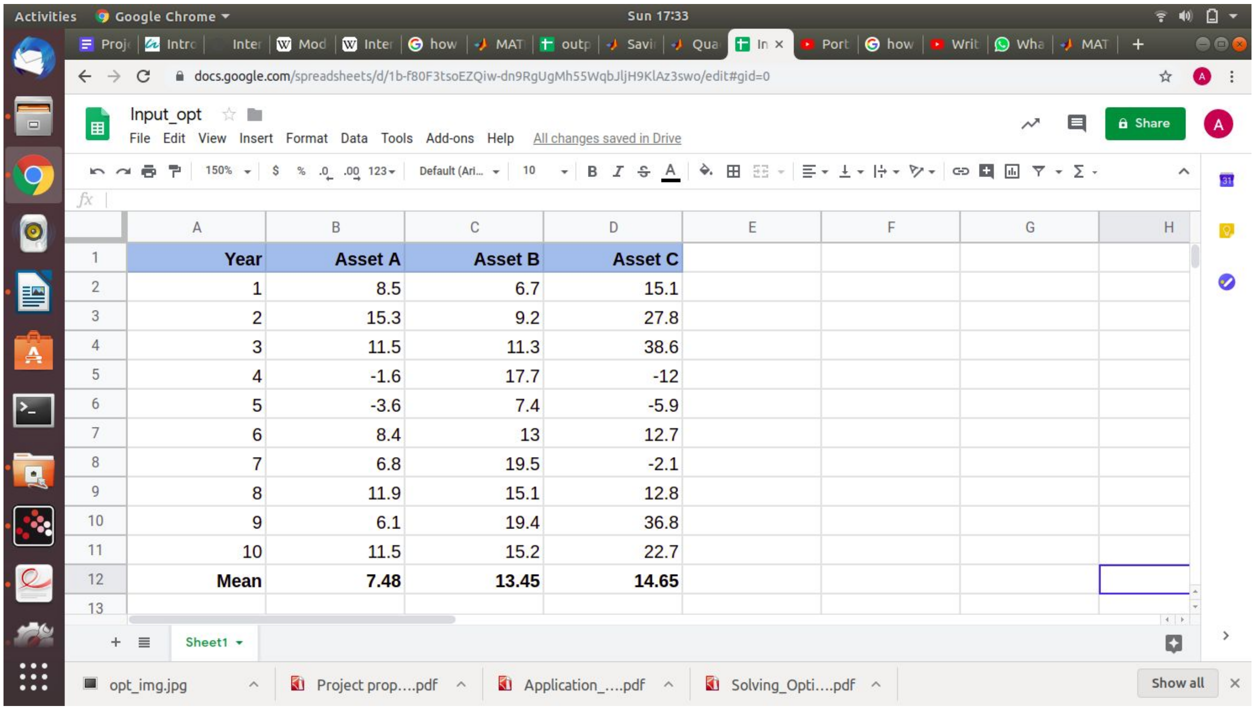
else repeat **step 1**.

Implementation of Matlab code for Portfolio Optimisation problem

(Code has been uploaded in the following link https://github.com/atche333/portfolio_optimisation) **:**

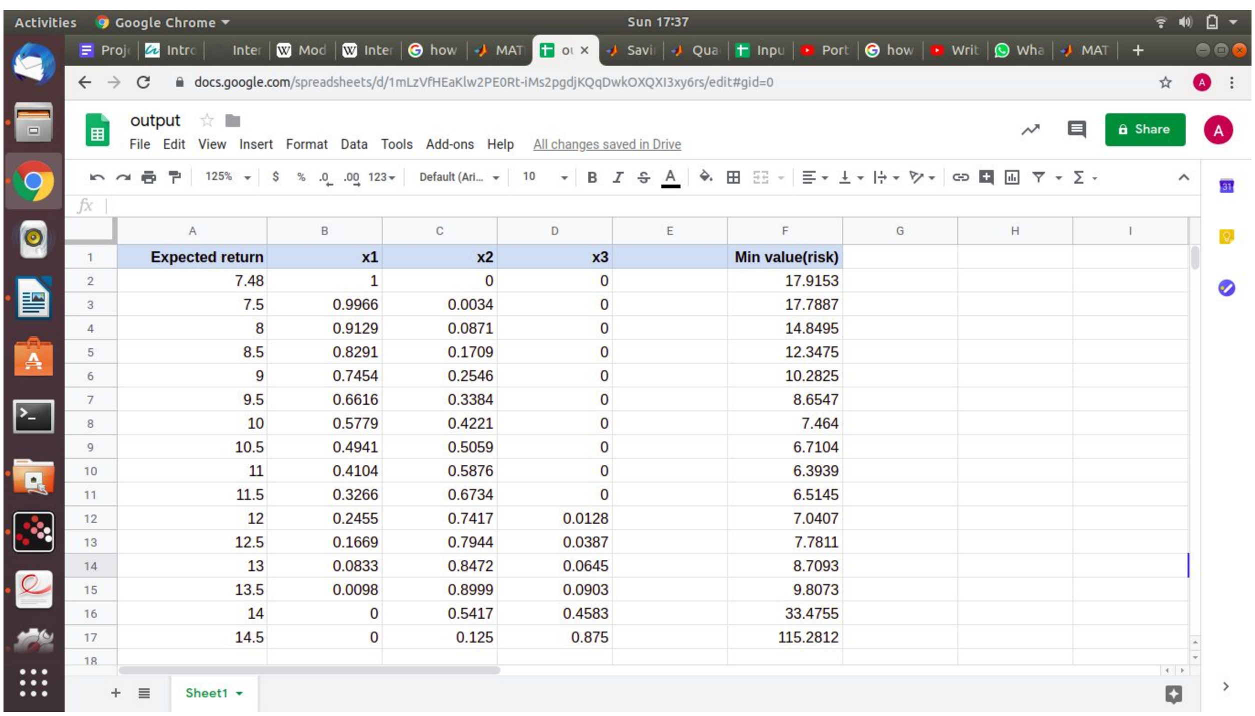
- We calculate mean, variance of asset values from the data read from excel sheet(input as shown in the
- below figure).
- Risk Function to be minimised for portfolio optimisation is formulated, which is a quadratic programming problem with two equality, one inequality and bound constraints.
- This quadratic programming problem is solved using the function quadprog() which takes covariance matrix, constraints and bounds as parameters.
- The quadprog() function uses the interior point method for solving the optimisation problem.
- Function returns weights(proportions) of the assets to be taken to get minimum risk and also the minimum risk value for expected return (as shown in the below figure).
- The closeness of the returned value with the optimal minimisation point is checked by the KKT error computed using the KKT conditions as mentioned in the code.

Input :



Year	Asset A	Asset B	Asset C
1	8.5	6.7	15.1
2	15.3	9.2	27.8
3	11.5	11.3	38.6
4	-1.6	17.7	-12
5	-3.6	7.4	-5.9
6	8.4	13	12.7
7	6.8	19.5	-2.1
8	11.9	15.1	12.8
9	6.1	19.4	36.8
10	11.5	15.2	22.7
Mean	**7.48**	**13.45**	**14.65**

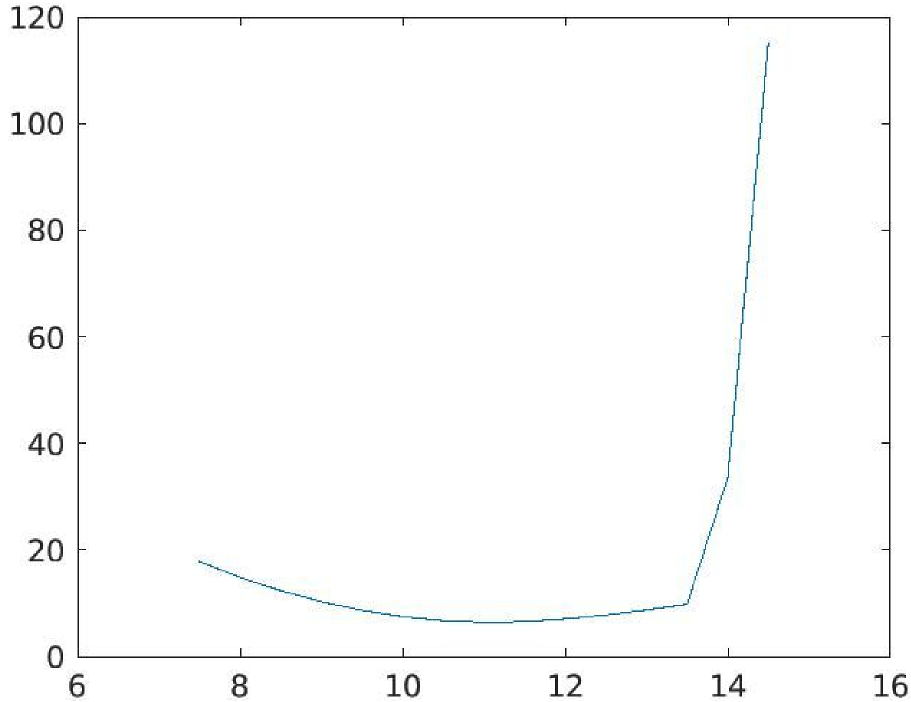
Output :



Expected return	x1	x2	x3		Min value(risk)
7.48	1	0	0		17.9153
7.5	0.9966	0.0034	0		17.7887
8	0.9129	0.0871	0		14.8495
8.5	0.8291	0.1709	0		12.3475
9	0.7454	0.2546	0		10.2825
9.5	0.6616	0.3384	0		8.6547
10	0.5779	0.4221	0		7.464
10.5	0.4941	0.5059	0		6.7104
11	0.4104	0.5876	0		6.3939
11.5	0.3266	0.6734	0		6.5145
12	0.2455	0.7417	0.0128		7.0407
12.5	0.1669	0.7944	0.0387		7.7811
13	0.0833	0.8472	0.0645		8.7093
13.5	0.0098	0.8999	0.0903		9.8073
14	0	0.5417	0.4583		33.4755
14.5	0	0.125	0.875		115.2812

Graph of expected return v/s minimum risk

(Min risk)



(Expected return)

Conclusion :

We have studied the barrier method and Primal-Dual Interior point method, and applied this to solve the portfolio optimisation problem.

From the results obtained for this set of data taken(input), the risk factor initially decreases as the expected return value is increased and after a certain point (when there is no optimal solution possible) a drastic increase in the risk factor with expected return value can be observed.