ASSIGNMENT 2

Linear Programming in Investment Portfolio Creation

IME611A

Platform Used - Python (Scipy Library)

4/6/2020

Submitted by - Aman Jangid (170082)
Submitted to - Prof Shankar Prawesh, Prof Suman Saurabh

Problem Statement-

I want to maximize the return by taking consideration of risk index of each client. Some Client are risk loving and some are risk aversion so I want to show them portfolio based on their risk taking.

There are some constraints in risk and bounds in weights which we have to consider:

Given →

weights

W_1	Growth Fund	20 to 40%
W_2	Balanced Fund	20 to 50%
W_3	Government Fund	At least 30%

So, $\boxed{0.20 \leq W_1 \leq 0.40}$ $\boxed{0.20 \leq W_2 \leq 0.50}$, $\boxed{W_3 \geq 0.30}$

And $W_1 + W_3 + W_2 = 1$

	Risk Assestment
Growth fund	0.2
Balanced Fund	0.08
Government Fund	0.01

So, $0.2W_1 + 0.08W_2 + 0.01W_3 >= 0.05$

So, Till Now we have 2 constraints & bounds.

Const1 → $W_1 + W_2 + W_3 = 1$

Const 2 → $0.2W_1 + 0.08W_2 + 0.01W_3 >= 0.05$

Bounds: $0.20 \leq W_1 \leq 0.40$, $0.20 \leq W_2 \leq 0.50$, $W_3 \geq 0.30$

Objective of probleem →

Maximizing Portfolio return.

So, Portfolio return $= W_1R_1 + W_2R_2 + W_3R_3$.

$\text{Max}\,(W_1R_1 + W_2R_2 + W_3R_3)$

Methodology –

In this assignment I use python scipy library (scipy.optimize) for solving this linear programming problem. I use Sequential least square Algorithm method for optimization. The objective and constraints are shown in python code with comments. I have also plotted a graph between max. return and risk index which proves to be a very good tool for showing different clients. I have obtained the weights in each asset and maximum portfolio return shown in python code.

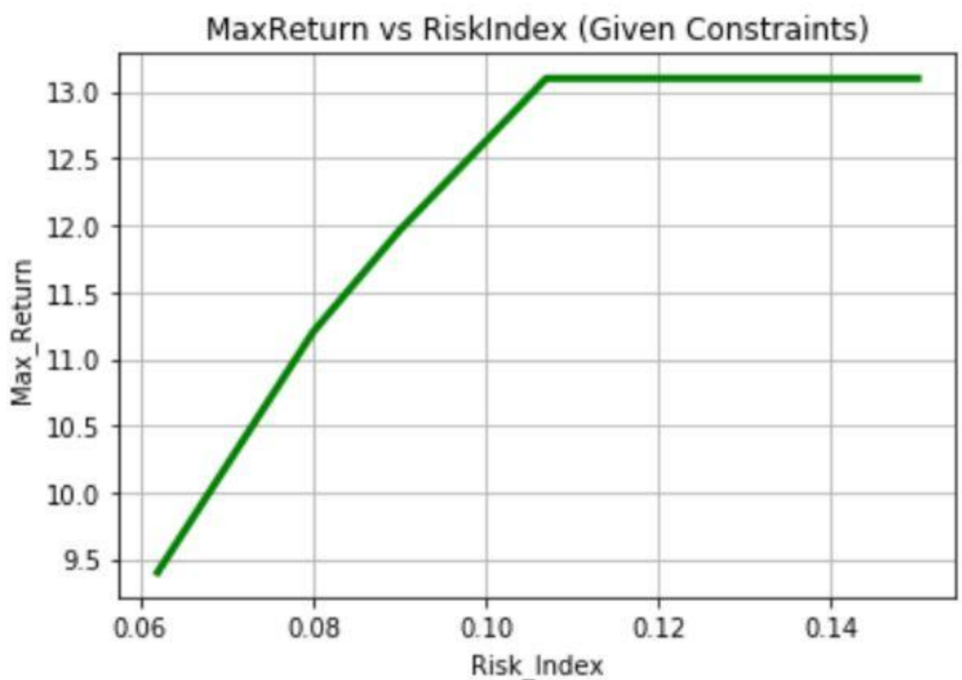
As higher the return is, more the risk is so, based on the clients risk taking capacity, I can show him different portfolio options which I calculated by only changing one constraint in my code i.e. risk index.

For more safer clients we will more focus on government bonds as they have very low risk and for risk loving more focus on growth funds.

Results -

1. For a hypothetical client risk index>=0.05
   Weights = [0.4, 0.3, 0.3]
   If the clients risks taking capacity is greater than 0.05 then maximum return (assuming forecasted return work) comes by investing 40% in growth fund, 30% balanced fund and 30% Government bonds. (results obtained by python code).
   Maximum Portfolio Return = 0.4*20 + 0.3*12 + 0.3*5 = 13.1% (shown in python code).

2. Plot showing on changing risk index how maximum portfolio return changes (following given constraints). An increase and decrease in the risk changes the investment choices.



This graph is consider as a good tool for showing different clients, as each client has his own risk taking capacity.

In the graph as risk is increasing max return increases but after a certain risk index max. return is same this is because of our constraints of weights that we can invest in asset in some range weight.

Graph shown in python code.

3. This model will provide benefits to other clients –

- We are using more diversified portfolio by bounding weights in each asset.
- We are showing Max. Return vs risk index to the clients so that he can choose his own choice of portfolio.
- By changing weights bounds or risk index in our model we can obtain more investment choices for our clients.

Python Code –

```
In [287]: #importing Libraries.

          import numpy as np
          from scipy.optimize import minimize
          from matplotlib import pyplot as plt
```

```
In [277]: #Main Objective Function i.e Potfolio_Return to whom we want to maximize.(Sign=-1) Becoz We want to maximize using minimize tool

          def objective(weight,Return = [0.20,0.12,0.05],sign=-1):
              Portfolio_return = []
              for i in range(len(weight)):
                  Portfolio_return.append(weight[i]*Return[i])
              return sign*sum(Portfolio_return)
```

```
In [278]: #Const1 - Sum of all weights equal to 1

          def constraint1(weight):
              return (sum(weight) - 1)
```

```
In [293]: #Const2 - Total Risk Should be greater than 0.05(given in question).We can change for checking variation of max_return with risk.

          def constraint2(weight,risk_assestment = [0.2,0.08,0.01],sign=1):
              total_risk = []
              for i in range(len(weight)):
                  total_risk.append(weight[i]*risk_assestment[i])
              return sign*(sum(total_risk) - 0.05)
```

```
In [294]: #Weights Intial Assumption.

          weight = [0.30,0.30,0.40]
          Return = [0.20,0.12,0.05]
```

```
In [295]: #Bounds of each weight.

          w1 = (0.20,0.40)
          w2 = (0.20,0.50)
          w3 = (0.3,0.6)
```



```
In [296]: #All constraints and bounds together.

          con1 = {'type':'eq','fun':constraint1}
          con2 = {'type':'ineq','fun':constraint2}
          cons = [con1,con2]
          bound = [w1,w2,w3]
```

```
In [297]: #Optimization of Weigths for maximizing return using Sequential Least Square Algo(SLSQP).

          solution = minimize(objective,weight,method='SLSQP',bounds=bound,constraints=cons)
```

```
In [300]: #New Weights after optimization.

          weight = solution.x
          weight

Out[300]: array([0.4, 0.3, 0.3])
```

```
In [299]: #Maximum Portfolio Return given above constriants.

          Pt_Return = 0
          for i in range(len(weight)):
              Pt_Return = Pt_Return + weight[i]*Return[i]
          Pt_Return*100

Out[299]: 13.099999999999998
```

```
In [ ]:   #Obtaining max_return by changing risk_Index as different client prefer different risk, good to show graph to clients.

          '''IF
           Risk_Index     Max_Return(%)
             =0.062          9.40
             <=0.07         10.20
             <=0.08         11.20
             <=0.09         11.96
             <=0.10         12.63
             <=0.107        13.1
             <=0.15         13.1
          '''
          '''Maximum return we can extract from these constraints is 13.01 (Weight = [0.4,0.3,0.3])'''
```

In [301]:
```
#Risk_Index and thier respective max_return.
Risk_Index = [0.062,0.07,0.08,0.09,0.10,0.107,.15]
Max_Return = [9.40,10.20,11.20,11.96,12.63,13.1,13.1]
```

In [302]:
```
#Plot showing RiskIndex that upto certain risk what is my max_return (assumming my forcast return and bounds of my weight).
plt.plot(Risk_Index,Max_Return,color="green",linewidth=3)
plt.title("MaxReturn vs RiskIndex (Given Constraints)")
plt.xlabel("Risk_Index")
plt.grid(True)
plt.ylabel("Max_Return")
plt.show()
```