Portfolio Optimisation for US based equity Instruments using Monte Carlo Simulation

Ayan Mukherjee, Gargi Mondal, Ashish Kumar Singh

Department of Computer Science & Engineering
Kalinga Institute of Industrial Technology

Key Words : Modern Portfolio Theory(MPT), Monte Carlo Simulation, Portfolio Optimization, Equity Instruments, Risk Profile

Abstract

The Modern Portfolio Theory (MPT) or mean-variance analysis is a mathematical modelling technique that is used for creating portfolios which aims to maximize the return for a given amount of risk. In this paper we analyse the possibility of constructing optimised portfolios in accordance to the modern portfolio theory for US based equity instruments. Given a Portfolio having 'n' number of stocks, with each stock 'i' having weight of 'w_i' we compute the mean and risk (standard deviation) and optimise our portfolio by optimizing the weights 'w_i' for the equity instruments using Monte Carlo simulation.

Introduction

The modern portfolio theory was devised by Henry Markowitz in 1952, He was later awarded the 1990 Nobel Prize in Economics.

The Mordern Portfolio Theory is based on the assumption that the nature of investors is that of being risk averse, i.e given 2 portfolios that offer same exected return, an investor shall prefer the one with minimal risk. Thus, an investor is willing to take a higher risk only if the same is compensated by a higher expected return. Conversely, it can be said that an investor who wants higher expected returns must be willing to except a higher risk. The exact trade-off between Risk and Expected Returns is not same for all investors. Different investors will evaluate the trade-off in differently based on individual risk aversion characteristics. The implication being a rational investor will not invest in a portfolio if there exists a another portfolio with a more favourable risk-expected return profile.

Under the model:

- Portfolio return is the proportion-weighted combination of the constituent assets' returns.
- Portfolio volatility is a function of the correlations ρ_{ij} of the component assets, for all asset pairs (i, j).

In general:

Expected return:

$$\mathrm{E}(R_p) = \sum_i w_i \, \mathrm{E}(R_i)$$

Where R_p is the return on the portfolio, R_i is the return on asset *i* and w_i is the weighting of component asset i (that is, the proportion of asset "i" in the portfolio).

Portfolio return variance: It is a measure of risk in this model. A higher variance indicates a higher risk for the asset class and the portfolio.

$$\sigma_p^2 = \sum_i w_i^2 \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \rho_{ij},$$

where σ is the (sample) standard deviation of the periodic returns on an asset, and ρ_{ij} is the correlation coefficient between the returns on assets *i* and *j*. Alternatively the expression can be written as:

$$\sigma_p^2 = \sum_i \sum_j w_i w_j \sigma_i \sigma_j \rho_{ij}$$

where ρ_{ij} for *i* = *j* , or

$$\sigma_p^2 = \sum_i \sum_j w_i w_j \sigma_{ij}$$

where σ_{ij} is the (sample) covariance of the periodic returns on the two assets, or alternatively denoted as $\sigma(i,j)$, *cov*$_{ij}$ *or cov(i,j)* .

Portfolio return volatility (standard deviation):

$$\sigma_p = \sqrt{\sigma_p^2}$$

Sharpe Ratio : It is the measure of the return of an investment in relation to the risk-free rate(Treasury rate) & its risk profile. In general a higher value of SR indicates a better & more lucrative investment. It can be said that when comparing 2 portfolios with similar risk profiles, all all else equal it would be better to invest in the portfolio having a higher Sharpe Ratio.

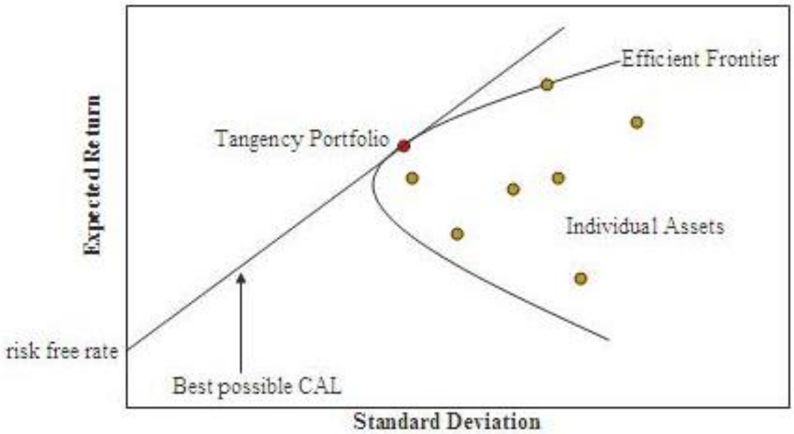
$$\frac{R_p - R_f}{\sigma_p}$$

R_p is the return of portfolio
R_f is the risk free rate

σ_p Standard deviation of the portfolio's excess return.

The Efficient Frontier :

This plot measure risk vs returns and is used to select the most optimum portfolio to invest into after considering the risk profile and the characteristics of the investor. The efficient frontier is essential the part of the curve in the first and second quadrants depending on the objective and investor ability/characteristics.



The Capital Allocation Line (CAL) is essentially a tangent to the efficient frontier. The point of intersection between the tangent and the frontier is considered to be the optimal investment which has maximum returns for a given risk profile, under normal conditions

The MPT is instrumental for investors keen on diversifying their portfolios. As a matter of fact the growth of ETFs (Exchange Traded Funds) made the theory more relevant by allowing investors to have an eaiser access to different asset classes.

Equity Investors manage risk using MPT by putting a small portion of their portfolios in Govt. EFTs. This way the variance of the portfolio becomes significantly lower since govt. bonds have a –ve cor-relation with equity instruments. This ensures that there's no large impact on the expected return because of this loss reducing effect.

Methodology

Annual Returns and Standard Deviation

In this paper we have considered the daily returns and standard deviation for 1 month of stock data. However, in practice, institutions work with annual returns and standard deviation. The formulae for converting daily returns and standard deviation to an annual basis are as shown (assuming 252 trading days in a year):

Annual Return = Daily Return * 252 Annual Standard Deviation = Daily Standard Deviation * 252

In this paper we have considered a portfolio consisting of four stocks in banking/financial services sector, namely:

- Bank of America (BAC)
- Goldman Sachs (GS)
- JP Morgan Chase & Co (JPM)
- Morgan Stanley (MS).

Initially random weights are assigned to the above 4 stocks keeping the sum of weights to be 1. The expected return and standard deviation of the portfolio is then calculated and stored.
Monte Carlo Simulations is then used on the portfolio to get optimal weights for each equity instrument in our portfolio in python.

Monte Carlo Simulation

Monte Carlo simulations define a method of computation that uses a large number of random samples to obtain results. They are often used in physical and mathematical problems and are most useful when it is difficult or impossible to use other mathematical methods. Monte Carlo methods are mainly used in three distinct problem classes: optimization, numerical integration, and generating draws from probability distributions. In this simulation, random weights are assigned to the equity instruments keeping the sum of weights to be 1. The expected return and S.D is then calculated for every combination of these weights and stored. Then weights are then again changed, assigned randomly and the process is repeated.
The number of iterations depends on the error that the investor is willing to accept. Higher the number of iterations, higher will be the accuracy of the optimization but at the cost of computation and time.
In this paper we have considered 10000 such iterations, out of these 10000 results of Expected return and risk profile generated, portfolio optimization can be achieved by identifying the portfolio which satisfies any of these 3 conditions:

1. A portfolio which has the minimum risk for the desired level of expected return.
2. A portfolio which gives the maximum expected return at the desired level of risk.
3. A portfolio which has the maximum return to risk ratio or Sharpe ratio.

Portfolio Optimsation process in Python

The stock price data was fetched from yahoo and saved under DataFrame name '*data'*.

```
stock =['BAC','GS','JPM', 'MS']
data = web.DataReader(stock, data_source = "yahoo", start = '12/31/2017',end= '12/31/2017')['Adj Close']
```

following which the same was sorted and then converted into returns and will save this under the name '*stock_ret*'.

Date	BAC	GS	JPM	MS
2017-11-30	28.17	247.64	103.98	51.61
2017-12-01	28.10	248.95	104.25	51.95
2017-12-04	29.06	250.65	106.40	52.68
2017-12-05	28.93	248.33	105.17	52.01
2017-12-06	28.64	245.95	104.39	51.69
2017-12-07	28.78	248.56	104.08	52.35
2017-12-08	29.05	250.35	105.38	52.89
2017-12-11	28.94	250.13	105.07	52.77
2017-12-12	29.32	257.68	106.30	53.85
2017-12-13	28.84	255.56	104.96	53.18
2017-12-14	28.73	255.48	104.12	52.64
2017-12-15	29.04	257.17	105.59	53.10
2017-12-18	29.48	260.02	106.41	53.24
2017-12-19	29.45	256.48	105.96	52.93
2017-12-20	29.48	255.18	105.59	52.51
2017-12-21	29.82	261.01	107.27	52.88
2017-12-22	29.88	258.97	106.89	52.72
2017-12-26	29.78	257.72	106.47	52.47
2017-12-27	29.73	255.95	106.66	52.57
2017-12-28	29.80	256.50	107.23	52.65
2017-12-29	29.52	254.76	106.39	52.47

#compute stock returns and print the returns in percentage format

```
stock_ret = data.pct_change()
print(stock_ret.round(4)*100)
```

The mean return of all stocks and the covariance matrix was then computed.

#Calculate mean returns and covariances of all four the stocks

```
mean_returns = stock_ret.mean()
cov_matrix = stock_ret.cov()
print (mean_returns)
print (cov_matrix)
```

```
BAC     0.002405
GS      0.001470
JPM     0.001195
MS      0.000870
dtype: float64
```

	BAC	GS	JPM	MS
BAC	0.000124	0.000081	0.000095	0.000078
GS	0.000081	0.000128	0.000081	0.000086
JPM	0.000095	0.000081	0.000094	0.000075
MS	0.000078	0.000086	0.000075	0.000087

The results of each iteration was stored in an array. The number of columns in the array will change with the number of stocks in the portfolio, since the weights of all the stocks are to be stored.

The len function has been used while defining the array. The number of rows in the array is equal to the number of *iterations.*

#Set the number of iterations to 10000 and define an array to hold the simulation results; initially set to all zeros

```
num_iterations = 10000
simulation_res = np.zeros((4+len(stock)-1,num_iterations))
for i in range(num_iterations):

#Select random weights and normalize to set the sum to 1
        weights = np.array(np.random.random(4))
        weights /= np.sum(weights)

#Calculate the return and standard deviation for every step
        portfolio_return = np.sum(mean_returns * weights)
        portfolio_std_dev = np.sqrt(np.dot(weights.T,np.dot(cov_matrix, weights)))
```

```
#Store all the results in a defined array
        simulation_res[0,i] = portfolio_return
        simulation_res[1,i] = portfolio_std_dev

#Calculate Sharpe ratio and store it in the array
        simulation_res[2,i] = simulation_res[0,i] / simulation_res[1,i]

#Save the weights in the array
        for j in range(len(weights)):
            simulation_res[j+3,i] = weights[j]
```

The output was saved in a *'pandas data frame'* for easy analysis and plotting.

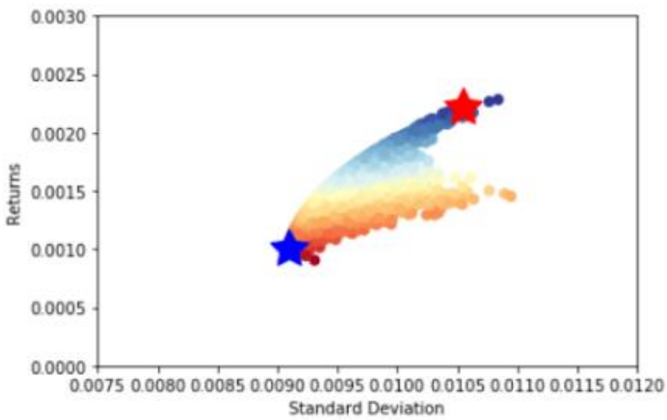
	ret	stdev	sharpe	BAC	GS	JPM	MS
0	0.001131	0.009189	0.123067	0.027700	0.054452	0.571253	0.346595
1	0.001397	0.009474	0.147459	0.142838	0.329737	0.338161	0.189263
2	0.001675	0.009681	0.173042	0.355449	0.293747	0.256548	0.094256
3	0.001503	0.009491	0.158379	0.239681	0.258662	0.338860	0.162797
4	0.001528	0.009553	0.159959	0.239959	0.304541	0.329268	0.126231

	ret	stdev	sharpe	BAC	GS	JPM	MS
9995	0.001602	0.009583	0.167130	0.316478	0.230561	0.330747	0.122214
9996	0.001333	0.009402	0.141731	0.115071	0.307778	0.311639	0.265512
9997	0.001571	0.009606	0.163501	0.277284	0.353731	0.192777	0.176208
9998	0.001517	0.009439	0.160689	0.312987	0.104245	0.319382	0.263386
9999	0.001602	0.009772	0.163964	0.296160	0.453444	0.017170	0.233226

```
The portfolio for max Sharpe Ratio:
ret          0.002224
stdev        0.010542
sharpe       0.210910
BAC          0.822040
GS           0.147847
JPM          0.009119
MS           0.020994
Name: 9294, dtype: float64
The portfolio for min risk:
ret          0.001010
stdev        0.009090
sharpe       0.111160
BAC          0.008306
GS           0.019566
JPM          0.356755
MS           0.615373
Name: 2260, dtype: float64
```

The above figures show the output where s some rows of the simulation results are shown. Portfolios having maximum Sharpe ratio or minimum risk was then computed.

The output was then plotted using the matplotlib library as the relevant points can be highlighted as shown



In the output, the red star shows the portfolio with the maximum Sharpe ratio and the blue star depicts the point with the minimum standard deviation.

From the above curve, we can get the composition for the required optimal portfolio based on any of the three conditions as discussed above. The portfolio with maximum return for a given risk or a portfolio with minimum risk for a given return can be selected or the portfolio with maximum Sharpe ratio can be selected.

Conclusion

The literature around portfolio optimization is rich and vast. There are a wide variety of variations and improvements upon the basic methods and a lot of active research that goes around it. In this paper we have used Monte Carlo simulation to optimise our portfolio that consisted of US based equity instruments from the banking sector, the same can be applied not only to equity instruments from wide array of sectors but also to different asset classes so as to diversify investor portfolio and improve the Risk to Expected Return Trade-Off.

References:

1. *Markowitz, H.M. (March 1952).* "Portfolio Selection". *The Journal of Finance. 7 (1): 77–91.* doi*:*10.2307/297597
2. *Markowitz, Harry (February 2000). Mean-Variance Analysis in Portfolio Choice and Capital Markets. Wiley.* ISBN 978-1-883-24975-5.
3. *Low, R.K.Y.; Faff, R.; Aas, K. (2016).* "Enhancing mean–variance portfolio selection by modeling distributional asymmetries" (PDF). *Journal of Economics and Business. 85: 49–72.* doi*:*10.1016/j.jeconbus.2016.01.003
4. Risk Manager Journal (2006), "New Approaches for Portfolio Optimization: Parting with the Bell Curve — Interview with Prof. Svetlozar Rachev and Prof.Stefan Mittnik
5. Doganoglu, Toker; Hartz, Christoph; Mittnik, Stefan (2007). "Portfolio Optimization When Risk Factors Are Conditionally Varying and Heavy Tailed
6. Maccheroni, F., Marinacci, M., Rustichini, A., Taboga, M. (2009). Portfolio selection with monotone mean-variance preferences. Mathematical Finance, 19(3), 487-521.
7. rechuk, B., Molyboha, A., Zabarankin, M. (2012). Mean-deviation analysis in the theory of choice, Risk Analysis: An International Journal, 32(8), 1277-1292.
8. Hubbard, Douglas; Samuelson, Douglas A. (October 2009). "Modeling Without Measurements". OR/MS Today: 28–3
9. Kroese, D. P.; Brereton, T.; Taimre, T.; Botev, Z. I. (2014). "Why the Monte Carlo method is so important today". WIREs Comput Stat. 6 (6): 386–392. doi:10.1002/wics.1314. S2CID 18521840
10. *Hastings, W. K. (1970-04-01).* "Monte Carlo sampling methods using Markov chains and their applications". *Biometrika. 57 (1): 97–109.* Bibcode*:*1970Bimka..57...97H. doi*:*10.1093/biomet/57.1.97. ISSN 0006-3444. S2CID 21204149
11. Hetherington, Jack, H. (1984). "Observations on the statistical iteration of matrices". Phys. Rev. A. 30 (2713): 2713–2719. Bibcode:1984PhRvA..30.2713H. doi:10.1103/PhysRevA.30.271