Risk Management

A Quantitative Approach To Organise And Develop New Ways To Monitor And Hedge Positions On The Market. Decision-Making Process Based On The Monte Carlo Simulation And Different Sampling Methods.

Introduction

Over the last few years, numerous types of methods and algorithms have been developed to reach a **high level** of risk management in the financial markets. Various meanings are assigned to the general concept of Risk Management, from Portfolio Risk Management to Credit Risk Management, going through Liquidity Risk Management and many others. This paper aims to give a **quantitative approach** in the *selection* and *allocation* of assets in a portfolio, more specifically, the type of assets considered will be stocks, bonds, currencies, commodities, indexes, options, forwards and futures. The IDE used for the research was *CLion*, specific development software for C/C++ programs. The software is based on *C++ 20* and the libraries used are: (1) Yahoo Finance and (2) Nlohmann JSON.

(1) is needed for the Financial Market Data, all the information such as: *Open*, *High*, *Low*, *Close* and *Adjusted Close* prices on a daily basis are easily fetched through the API

(2) is needed to read and write JSON files, these have been used to read the information regarding the portfolio we want to simulate and to store the results coming from the simulation process

Before using C++ as main programming language, the preliminary analysis has been performed through Python 3.9, for the simulation, and MATLAB, for the visualisation. Particular attention have to be given to the **sampling methods**, the extraction of the returns of a specific asset from the dataset have been done in different ways, as an example: considering a uniform probability distribution for all the samples or by giving more attention to the last returns (increasing probability distribution). Actually, many different sampling methods could be taken in consideration, they could also be compared in order to understand which of those better describe the reality. Towards the end of the paper, you will find the explanation regarding a sampling method based on **Markov Chain**.

Structure

First of all, it is necessary to prepare all the *preliminary analysis* relatively to the assets we want to buy or sell for our portfolio. Once the choices have been taken relatively to the portfolio composition (e.g., *UCG.MI* and *BAMI.MI*), we have to establish the time horizon (e.g., Days), the number of combinations (of coefficients, e.g., for a portfolio composed by two assets, we obtain a matrix $C \in \mathbb{R}^{M,N}$, with N indicating the number of combinations and M representing the number of assets, each row sums up to 1), the number of simulation for each combination of coefficient (It will be indicated with the letter K). As an example, let's suppose we want simulate the performance of a portfolio composed by two stocks, with their specific returns datasets (Generally, we consider a dataset of 1-year at least). Firstly, the program generates a matrix $A \in \mathbb{R}^{M,N}$, with N indicating the number of combinations of coefficients and M indicating the number of assets. Afterwards, we generate a tensor $B \in \mathbb{R}^{K,M,D}$, with K indicating the number of simulations for combination of coefficient, M indicating the number of assets and lastly, D representing the number of days considered for the simulation. We then multiply the Matrix A by the tensor B, obtaining K portfolio returns for each combination of coefficients and for each day, in a simpler way, we obtain a new tensor $T \in \mathbb{R}^{N,D,K}$. We could simply state that we *"concatenated"* K matrices belonging to $\mathbb{R}^{D,N}$. As a result, we add up 1 to all the entries in order to obtain the *yield* tensor and then we cumulate the product through all the columns of all the K matrices. In order to simulate the portfolios in a more realistic manner, we could also multiply this tensor element-by-element by an initial investment capital $C \in \mathbb{R}_{>0}$. Lastly, we compute the percentage of success (all the values greater than the C initial capital value increase the success ratio) for each entries of the K matrices. The outcome will simply be a matrix $M \in \mathbb{R}^{D,N}$, where each entry will represent the probability of success of a specific portfolio $i = \{1,\dots,N\}$ in a specific day $j = \{1,\dots,D\}$.

A particular sampling method

Let's consider a stock F with its specific dataset $\mathscr{D}$ on a specific time frame, instead of using a sampling approach based on a uniform probability distribution or based on an increasing probability distribution (useful when we want to give more importance to the last returns), we will consider a sampling method based on a *directed graph*, more specifically, we will develop the *conditional probabilities*, in order to obtain a more realistic perspective of the financial market. Firstly, we find the **minimum** and the **maximum** return of a stock in a specific period of time and, afterwards we create a partition S, so a finite list of the form $x_0, x_1, \dots, x_n$, where (the size of the each interval could vary):

$$a = x_0 < x_1 < \dots < x_n = b \quad a, b \in \mathbb{R}, a < b$$

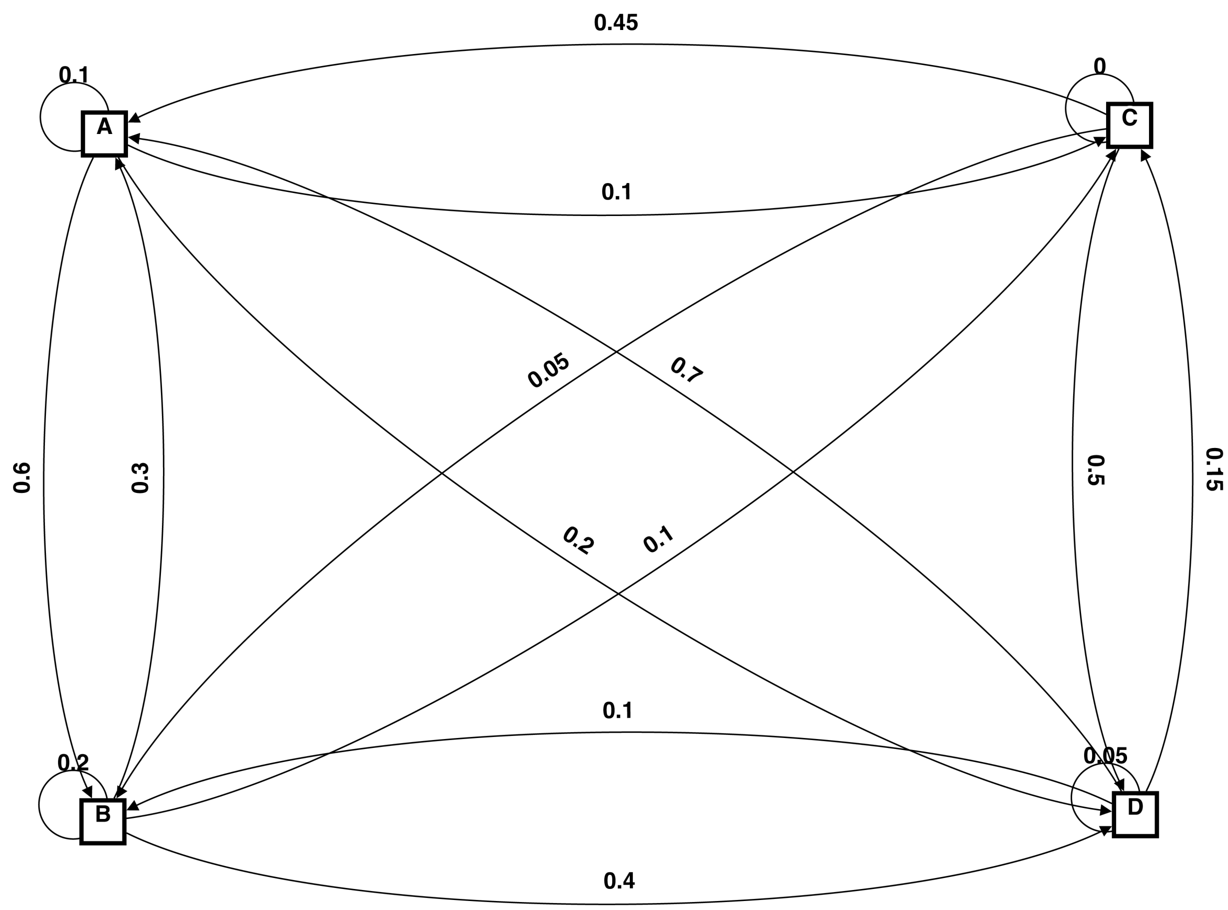
We use the partition S of the form $x_0, x_1, \dots, x_n$ of $[a, b]$ as a union of closed subintervals:

$$[a, b] = [x_0, x_1] \cup [x_1, x_2] \cup \dots \cup [x_{n-1}, x_n]$$

Let's now build a *weighted directed graph* with N nodes, corresponding to each subinterval in the partition, and N^2 directed edges, where the weights represents the probability for a return in a specific node (interval) to go to another node. E.g., If at time $t = 0$ the return belongs to the interval B, the probability at time $t = 1$, to reach a return in the interval A, C, D or again B is respectively equals to 0.3, 0.1, 0.4 and 0.2. We could also represents a *directed graph* as a matrix $G \in \mathbb{R}^{SxS}$ and give some consideration regarding that:

$$G = \begin{bmatrix} 0.1 & 0.6 & 0.2 & 0.1 \\ 0.3 & 0.2 & 0.1 & 0.4 \\ 0.45 & 0.05 & 0 & 0.5 \\ 0.7 & 0.1 & 0.15 & 0.05 \end{bmatrix}$$

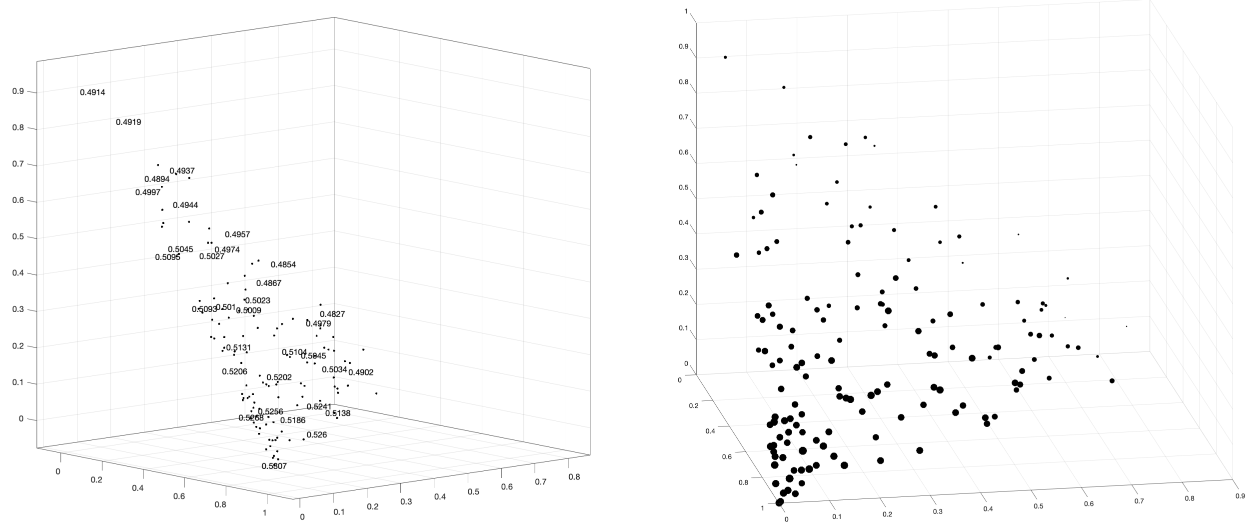
In particular, the matrix above is a **Markov Matrix** since, the sum of the entries of each row in the matrix is equal to 1 and all the entries are *non-negative real numbers.* This method have been implemented with some difficulties since the initial structure of the program didn't completely fit this type of sampling method. Nonetheless, this was really useful to obtain more reliable results and to understand how the return distributions worked compared with the other two approaches. The use of this sampling method could be helpful in understanding how actually a specific financial asset changes over specific time periods. They could be selected according to particular macroeconomics context, this would allow us to describe the financial market evolution in a discrete-time. As mentioned before, the time horizon relatively to the portfolio allocation often go from 7 days to 21 days roughly, the amount of days simply depends on the assets selection.



The figure above represents the weighted directed graph needed to describe the financial asset returns.

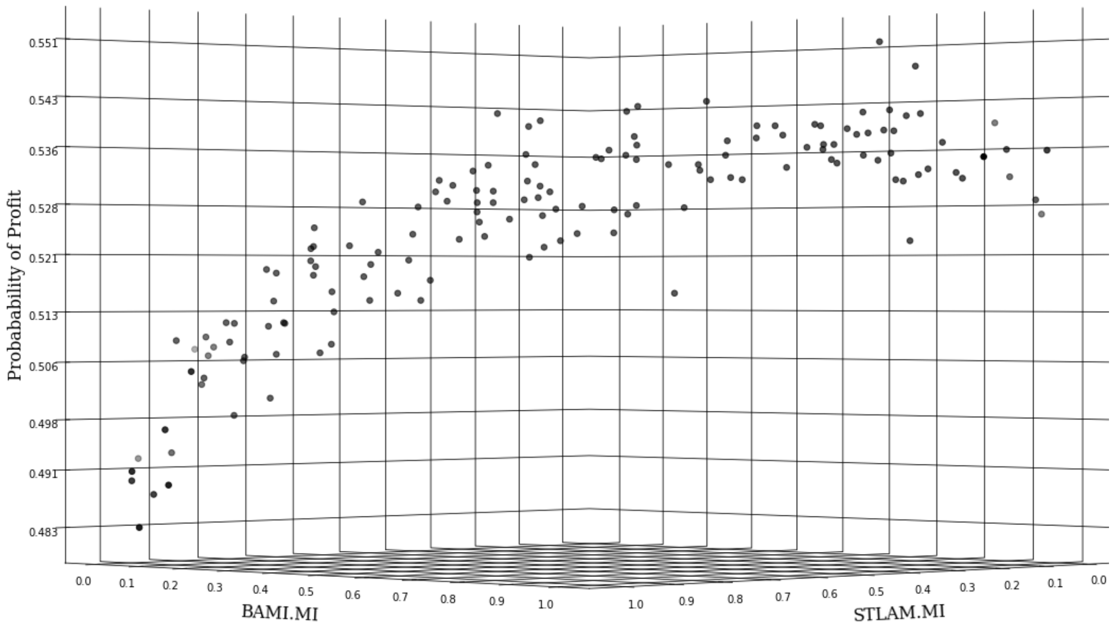
Plots

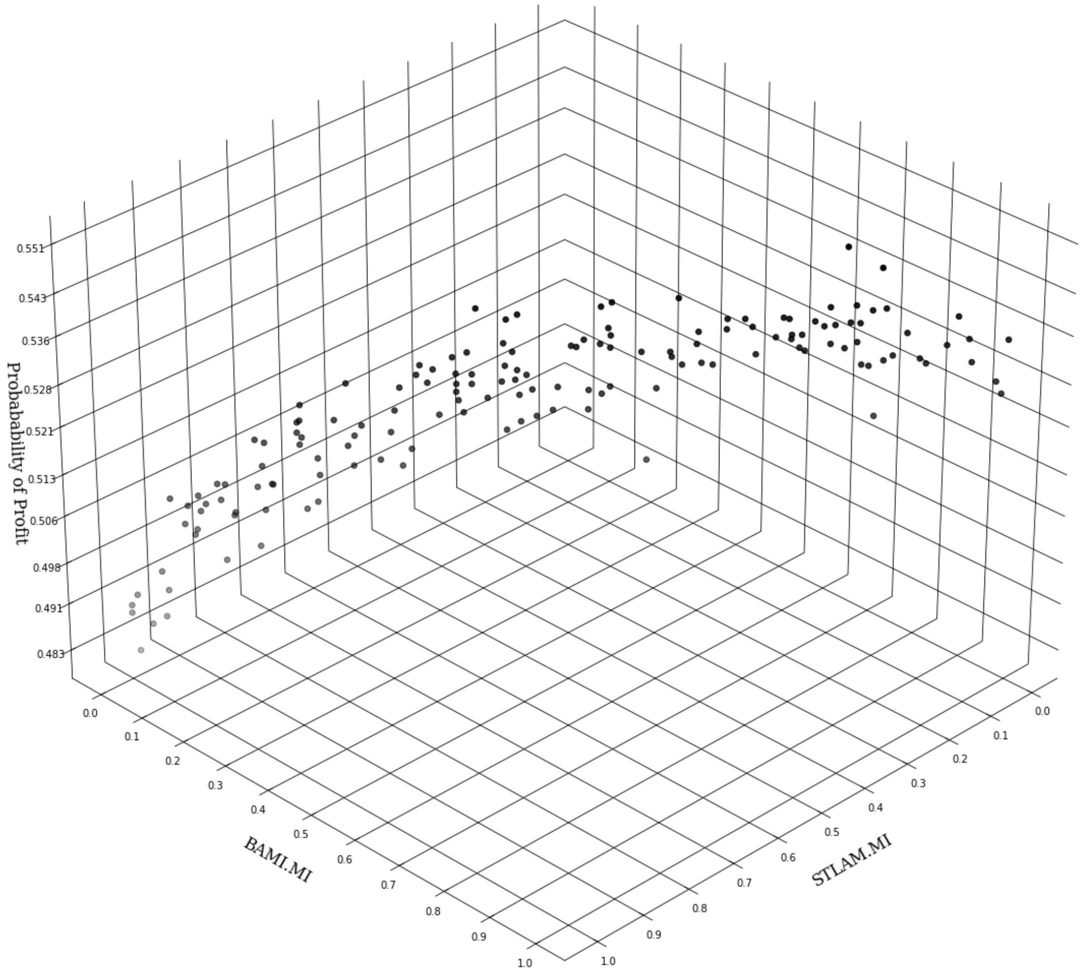
Once the simulation over the convex linear combinations of the chosen assets was performed, each result was plotted on a 3D scatter plot (samples portfolios were composed by two stocks). Obviously, the previous mentioned plotting methodology is not adaptable for a number of assets $N > 2$. For this reason, the use of **simplex geometry** could give us a better understanding and a different overview of the profitability of each portfolio combinations. As an example, if we would like to perform a simulation on three different stocks, we should visualise 3 different axes and all the results will rely on a *triangle*, each point representing the portfolio combination, in the figure below you will find a simulation of 150 portfolio combinations composed by three different stocks: UniCredit (UCG.MI), Moncler (MONC.MI) and Stellantis (STLAM.MI). For each portfolio combination, 10000 simulations have been performed (obviously the number of simulation could be greater) with a time horizon of 3 days. On the X, Y and Z axis, respectively figure coefficients relatively to UniCredit, Moncler and Stellantis.



The figures above respectively represent a 3D Text Scatter and a standard 3D Scatter Plot, in particular, the dots in the right-hand figure are proportional to the probability of success.

Thanks to the simplex geometry, we could also visualise portfolios composed by 4 different financial instruments through a *tetrahedron*. For what concerns, the visualisation of portfolio composed by two stocks, the next figures show a 3D Scatter Plot representing 150 portfolio combinations of the following stocks: Banco BPM (BAMI.MI) and Stellantis (STLAM.MI). For each portfolio combination 10000 simulations have provided with a time horizon of 3 days. To obtain a quick understanding of the outcomes we are observing: for each portfolio combination, we are simulating K number of times (K indicates the number of simulations) the performance of the portfolio on a specific time horizon. The probability of success is computed dividing the number of times the portfolio performed positively by the number of simulation set (K).





It is important to highlight that for both the simulation samples above, the sampling method was based on the uniform probability distribution of returns. Different results would be obtained using *different* sampling methods.

Conclusion

As shown before, there a lot of different sampling methods that could implemented, we could consider the idea to implement some sort of *attention coefficients*, this would be useful to give more attention to specific turbulent macroeconomics events, periods where the uncertainty could prevail. Another helpful analysis could regard the distribution of returns for each portfolio, how it could be shifted, minimised or stretched by hedging it (E.g., a bond would shift the distribution over the right, thanks to the guaranteed positive returns). Another important consideration could be done relatively to the time evolution of the returns portfolio distribution, analysing the performance on N different combinations and how each portfolio behaves in time subintervals (Stochastic Process described in a discrete time). Lastly, a more *complex* and *dynamical* approach could be considered, developing an allocation strategy based on martingales, in this way, the system would become a decision-maker algorithm. Before trying to pursue this objective would be better to find much more reliable sampling methods, as an example, we could consider the possibility to extract samples coming from other assets which are positively and negatively correlated $|\rho_{AB}| \in [0.85,1]$ (with A and B indicating two different assets). Other important aspects to consider regard the use of mixtures of assets distributions, these could be extremely useful to model portfolio through algorithms, more specifically, assuming that all the assets are normally distributed we could try to build portfolios with specific distribution (E.g., The *Expectation–maximisation Algorithm*), otherwise, we could also model portfolio as mixtures of non-gaussian distributed assets through *neural networks.*

Mario Nicolò De Matteis
mario.dematteis@studbocconi.it
+39 338 180 7445
Milan, Italy