A Monte Carlo Framework for Portfolio Evaluation

Amir Hadayo

Monte Carlo Methods 048716

Preface

In this paper we discuss the problem of evaluating financial portfolios. The problem of "distributing" wealth or resources among different options is much broader than the one related to financial investments. However, the latter has strong mathematical foundations, thus allowing thorough research.

The forces governing economical markets haven't been determined to full extent yet, so many assumptions need to be made in the process of modeling them. Some of the assumptions are reasonable, some are not. However, solving problems based on not-so-true assumptions might give us insights on how to solve the more realistic problems.

We begin with a description of the financial model we use followed by the mathematical model describing it. When then present some results in the context of optimal choosing of portfolios. After that, the simulation framework for this model is discussed and some simulation results are shown with some interesting insights. We end with conclusions and ideas for future work.

For the purpose of simulation, a new open-source Python library was written with emphasis on performance, modularity and generality for future uses. The code for the library is hosted on GitHub [Hadayo, 2020], and a basic documentation is presented as an appendix to this paper. A quick search on GitHub reveals that most of the repositories on portfolio optimization and evaluation discuss single period or multi-period models, as opposed to the continuous-time model discussed here. Thus the new library might serve as a basic building block for more advanced libraries.

Many of the information in the subsequent sections is adapted from [Korn and Korn, 2001] chapters 2 and 5, and [Shreve, 2004] chapters 3,4 and 5.

Contents

Nomenclature

CRRA	Constant Relative Risk Aversion
Money market account	A risk-free asset.
SDE	Stochastic Differntial Equation
Δt	The time difference between two points in the discretization grid.
$\mu(t)$	The stock's mean rate of return.
$P_0(t)$	Price of the money market account.
$P_1(t)$	Price of the stock
$\phi(t)$	The number of shares in the stock at time t.
$\pi(t)$	The fraction of wealth invested in the stock.
$r(t)$	Instantanous interest rate.
$\sigma(t)$	The stock's volatility.
T	The investing period horizon.
θ	The market price of risk.
$U(x)$	A utility function.
$W(t)$	A Brownian motion (Wiener process).
$X(t)$	The wealth process.

1. The Model

In this section we describe the financial model we follow in the paper and its mathematical description. At the end of the section, some historical notes on the use of Brownian motion to model stock prices are added.

1.1. The Financial Model

We examine a scenario with a single investor that invests his money over a predetermined time period $[0, T]$ where T is called the horizon. We assume a *continuous-time model*, in which the investor can act continuously for every $t \in [0, T]$.

The investor is faced with the problem of splitting his money between two possible assets:

1. The risk-free asset As the name suggests, this asset allows gaining without risk. Of course this implies the gain from this asset is very small. Since no asset has really no-risk involved in it, this is essentially one of the non-realistic assumptions in our model. However, it is quite a reasonable one as there are assets that involve very small amount of risk in them. Such an example is a 3-month U.S Treasury bill[1], that its gains very close to the central bank interest rate. In this paper we will call this asset *the money market account*.

2. The stock This is an asset that is driven by unpredictable forces (Brownian motion will be used for those soon). Investing in it gives a chance to gain more wealth than the risk-free asset, but with the price of the positive probability to gain less as well or even lose. We assume the *small investor hypothesis*, according to it the prices of the stock are not influenced at all by the investor's actions. This assumption doesn't hold for large institutions that trade thousands of stocks. By changing the demand or the supply of a large portion of shares from the same stock they can change its price substantially.

During the trading period, our investor can *re-balance* his holdings, i.e., sell some of the shares in one asset in order to buy more of the other. We also assume the investor begins his trading period with some fixed amount of money, and this money only and his gains and losses during the trading period will serve him to re-balance his portfolio. This last condition is called the *self-financing condition*.

At each time step $t \in [0, T]$, our investor should choose how many shares does he wish to hold from these two assets. We denote the number of shares from the stock at time t as $\phi(t)$. For a given portfolio, we assume whatever left after investing the stock, must be invested in the money market account.

The shares in both assets are assumed to be *perfectly divisible*, so that the investor can hold 1, or 2, or $1/2$ or even π shares if he wishes too. In practice, investors can usually

[1] The risk here is that the U.S government will default and won't pay the money back. For many, this scenario is considered very not likely.

only buy a whole number of shares, but there are brokerages that allow for dividing the amount of shares to certain resolutions.

We also assume a *friction-less market*, which is a market with no transaction costs. Out of all our assumptions, that might be the most unreasonable one but at the same time one of the most simplifying one.

Another reasonable assumption is that our investor actions can depend on the present and past only, i.e., $\phi(t)$ can depend on the information given only up to time t. Information will be described in the next section in the terms of *filtration*.

Shorting An important (and also a realistic) assumption is that the investor can have a *negative* number of shares. When an investor is holding a negative amount of shares of an asset, he is said to *short* the asset[2]. The most simple intuition to this scenario is an *overdraft*. Overdrafts are symbolized with a negative amount of money that is associated with a person, meaning he *owes* that money to the lender.

In the case of the risk-free asset, shorting it means *borrowing* money with the same interest it provides. In the case of the stock, implementing shorting in practice needs to be done by first borrowing the desired amount of shares from someone, probably your broker.

Shorting allows for a much wider spectrum of trading strategies, and very often, optimal portfolios include shorting components.

1.2. The Mathematical Model

We will now give a mathematical description for the financial model described above. Assume a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We denote $P_0(t)$ as the price of the money market account, and $P_1(t)$ as the price of the stock. Let $W(t)$ be a Brownian motion and $\{\mathcal{F}_t\}_{t=0}^{T}$ the filtration associated with this Brownian motion.

The number of shares of the stock $\phi(t)$ is a random process adapted to the filtration $\mathcal{F}_t$, i.e., $\phi(t)$ is $\mathcal{F}_t$-measurable for all $t \in [0, T]$. Thus for every $t \in [0, T]$, $\phi(t)$ can be evaluated after seeing the information up to and including time t.

1.2.1. The Assets Processes

Let $r(t)$ denote the instantaneous interest rate, usually measured in [years^{-1}]. We assume $r(t)$ is adapted to the filtration as well. The interest rate can be a constant or time-varying and even a random process, but it must be positive almost surely. The money market account dynamics are modeled by the Stochastic Differential Equation (SDE)

$$\mathrm{d}P_0(t) = P_0(t)\, r(t)\, \mathrm{d}t \tag{1}$$

[2]The contrary to short an asset is to long it, but the latter is rarely used as the former.

As always, SDEs should be interpreted in their integral form, which has a defined mathematical meaning. In the case of the equation above, given the nonrandom initial condition $P_0(t) = P_0$, the evolution of $P_0(t)$ is described by the equation

$$P_0(t) = P_0 + \int_0^t P_0(t)\, r(t)\,\mathrm{d}t \tag{2}$$

This equation can be easily solved by computing the differential of $\ln(P_0(t))$ using Itô's formula and integrating to get the explicit solution

$$P_0(t) = P_0 \exp\left(\int_0^t r(t)\,\mathrm{d}t\right) \tag{3}$$

Since the interest rate is always positive, we see $P_0(t)$ is a monotonically increasing function, thus our future gains are guaranteed. If $r(t)$ is known, they can even be calculated exactly. This kind of increasing is called *continuous compounding*.

In the case of the stock, we assume its price $P_1(t)$ follows the SDE of a *Generalized Geometric Brownian Motion*

$$\mathrm{d}P_1(t) = P_1(t)\,\mu(t)\,\mathrm{d}t + P_1(t)\,\sigma(t)\,\mathrm{d}W(t) \tag{4}$$

where $\mu(t)$ is the *mean rate of return* of the stock (usually measured in $[\text{years}^{-1}]$) and $\sigma(t)$ is the stock's *volatility* (usually measured in $\left[\text{years}^{-1/2}\right]$). Both of these functions can be constant or time varying, or even random process that are adapted to the filtration. As before, equation 4 should be interpreted in its integral form, which is (given the non-random initial condition for $P_1(0)$)

$$P_1(t) = P_1(0) + \int_0^t P_1(t)\,\mu(t)\,\mathrm{d}t + \int_0^t P_1(t)\,\sigma(t)\,\mathrm{d}W(t) \tag{5}$$

where the right integral is the *Itô Integral* with respect to the Brownian motion. This SDE can also be solved by using Itô's formula to calculate the differential of $\ln(P_1(t))$ and integrating to give the explicit form

$$P_1(t) = P_1(0) \exp\left(\int_0^t \left(\mu(t) - \frac{1}{2}\sigma^2(t)\right)\mathrm{d}t + \int_0^t \sigma(t)\,\mathrm{d}W(t)\right) \tag{6}$$

In the following we will mainly use a market model with $r(t), \mu(t)$ and $\sigma(t)$ that are constant. The stock model with constant μ and σ is the famous *Black-Scholes Model* from [Black and Scholes, 1973].

1.2.2. The Wealth Process

We now define the *wealth process* $X(t)$. This process measures the amount of money our investor has during the trading period. As we said in the previous section, we assume the investor starts with a fixed, nonrandom amount of wealth $X(0)$. Motivated by the self-financing condition, the dynamics of the wealth process are captured by the SDE

$$\mathrm{d}X(t) = \phi(t)\,\mathrm{d}P_1(t) + \left(X(t) - \phi(t)\,P_1(t)\right) r(t)\,\mathrm{d}t \tag{7}$$

This equation determines the source of the change in the wealth value over a small time interval. According to this equation there are two reasons for the change of wealth. The first one is due to the change in the stock price $\mathrm{d}P_1(t)$. Since our investor holds $\phi(t)$ shares of the stock at time t, the change in wealth due to the change in the price of the stock is exactly the left summand.

For the second summand, recall from the previous section that all the wealth that *is not* invested in stocks, is invested in the money market account. This amount, at time t, is exactly $X(t) - \phi(t)\,P_1(t)$, thus the right summand is actually equation 1 where the amount invested is $X(t) - \phi(t)\,P_1(t)$ and not $P_0(t)$.

Instead of tracking the number of shares the investor has of the stock at every moment $\phi(t)$, a more used notation for the trading strategy is in terms of relative wealth. Specifically, we denote $\pi(t)$ as the relative amount of wealth invested in the stock out of all the wealth the investor has, or mathematically

$$\pi(t) = \frac{\phi(t) \cdot P_1(t)}{X(t)} \tag{8}$$

We see that for each point in time, this equation gives a one-to-one relation between $\phi(t)$ and $\pi(t)$.

1.2.3. Notes: Modeling Stock Prices with Brownian Motion

Brownian motion has many desirable properties for the continuous model of stock prices. First, it has no tendency to rise or fall, that is reflected in its *martingale* property

$$\mathbb{E}\left[W(t) \mid \mathcal{F}_s\right] = W(s) \quad \forall 0 \le s \le t \le T$$

and also $\mathbb{E}[W(t)] = 0$ for all $0 \le t$. Thus Brownian motion has a very "random, unbiased" characteristic: it has no tendency to rise or fall, and at every moment, our best estimate of its future value is its present value.

Brownian motion became popular after Einstein's celebrated paper on the mechanics of small particles in 1905. But in fact, Brownian motion was used 5 years earlier, In the PhD thesis of Louis Bachelier, "Théorie de la spéculation". Bachelier suggested to model stock prices by Brownian motions with drifts, that is

$$P(t) = \mu t + \sigma W(t)$$

So the stock price has a deterministic component (μt) and a random component ($\sigma W(t)$). Assuming prices behave like above, Bachelier suggested methods for pricing options. The main drawback of Bachelier's model, is that stock prices have positive probability to reach negative values, which of course doesn't correspond with reality. His work was not well received in the mathematical community, especially because it was one of the first works to incorporate advanced mathematics in the finance domain. Nonetheless, Bachelier's work is considered the cornerstone of financial mathematics.

It wasn't until 1973 that the problem of modeling stock prices had a breakthrough. Fisher Black and Myron Scholes proposed the Geometric Brownian Motion model for stock prices, which is equation 4 with constant parameters. By looking at the solution in 6 we can see the stock price is surely positive. Also it has the nice features of the exponent function, where its change is proportional to its price. This is a desired properties as stock with a high price tend to change in much larger quantities compared to stock with a low price.

2. Optimizing Portfolios

We can now define a variant of the Portfolio Optimization problem. We consider our investor that starts at time 0 with $\$X(0)$ and invests it in the money market account and the stock over the time period T. The goal we are aiming for is to maximize some known function of the terminal wealth $U(X(T))$. This is of course a random variable, and its distribution depends on $r(t), \mu(t), \sigma(t)$ and $\phi(t)$. Thus maximizing this function of the terminal wealth is not well defined, and a more meaningful goal will be to maximize its expected value $\mathbb{E}[U(X(T))]$. It is known for more than a decade, that the maximization of wealth is not a good measure of peoples' desires. A good example is the St. Petersburg Paradox that was first analyzed by Daniel Bernoulli in 1738.

We now discuss the properties of a reasonable utility function, and then give some results on the Portfolio Optimization problem.

2.1. Utility Functions

The utility function, as described in [Markovitz, 1959] "simply represents the degree to which the individual is willing to take risks for outcomes presented by the wealth". More formally

Definition 1. Utility Functions

A function $U : (0, \infty) \to \mathbb{R}$ that satisfies the following conditions:

1. Strictly concave and continuously differentiable.
2. $\lim_{x \to 0^+} U'(x) = \infty$ and $\lim_{x \to \infty} U'(x) = 0$.

The first condition on the derivative reflects the hate to lose money. As the wealth gets closer to zero, every dollar lost implies a huge loss in utility. The second condition, on the other hand, reflects the indifference in wining a great amount of money. For most of us, gaining $\$1,000,000$ or gaining $\$1,000,001$ will result in the same joy. This is the property represented by this limit. Some examples for classic utility functions satisfying those conditions are: $\ln(x), \sqrt{x}$ and x^α where $0 < \alpha < 1$.

It is important to know that utility functions are defined up to affine transformations. Hence an important feature of a utility function is its *Risk Aversion* that is defined to be $-\frac{U''(x)}{U'(x)}$. The higher the curvature of U, the higher (and positive) the risk aversion. If U is convex, then the utility signifies "risk-loving" and the risk aversion will be negative.

Since risk aversion has units of $[\$^{-1}]$, a different measure used to distinguish utility function is the *Relative Risk Aversion* that is defined to be the risk aversions times the wealth x. It has similar properties to the risk aversion measure, but is unit-less and thus more universal. An important class of utility functions is the one with Constant Relative Risk Aversion, or more compactly CRRA. For instance the utility $U(x) = \ln x$ has relative risk aversion of 1 for all $x \in (0, \infty)$. Also the utility $U(x) = \sqrt{x}$ has relative risk aversion of $1/2$ for all $x \in (0, \infty)$.

2.2. The Portfolio Optimization Problem

Since the market parameters $r(t), \mu(t), \sigma(t)$ cannot be altered by our investor's will, The search space for maximization of the expected utility is over the trading strategies, $\phi(t)$ or $\pi(t)$. Thus the Portfolio Optimization problem we will be dealing with is:

Definition 2. The Portfolio Optimization Problem
Given a utility function $U(x)$ satisfying definition 1, market parameters $r(t), \mu(t), \sigma(t)$ and a horizon T, find the random process $\{\pi(t)\}_{t=0}^{T}$ that solves

$$\max_{\pi(t)} \mathbb{E}\left[U\left(X\left(T\right)\right)\right] \tag{9}$$

For obvious reasons, there are many trading strategies $\pi(t)$ that are by no means good strategies, thus sometimes the search space is narrowed to the following: Given a positive initial wealth $X(0) = x > 0$, search over all the adapted strategies $\pi(t)$ such that $X(t) \geq 0$ almost surely for all $t \in [0, T]$.

2.3. Solutions to the Portfolio Optimization Problem

The first serious attempt to solve the discussed problem was given by [Merton, 1969]. In this paper, Merton suggests a method based on the theory of stochastic control.

The second known method for solving the problem is widely called the *Martingale Method*. The origin of the name is in the Martingale Representation Theorem, and in the fact the discounted wealth process is a martingale under the risk-neutral measure. Loosely speaking, this method first determines the optimal terminal value, i.e., solving for the random variable B[3] that solves

$$\max_{B} \mathbb{E}\left[U\left(B\right)\right]$$

Then, it constructs a trading strategy $\pi(t)$ such that $X(T) = B$ almost surely. The existence of this trading strategy is justified by the Martingale Representation Theorem.

Finding an explicit for $\pi(t)$ given B is the hard problem. We will present one closed form solution based on [Karatzas et al., 1987]. Assuming equations 1,4 hold with constant market parameters r, μ, σ and and assume U is a CRRA utility function with R denoting its relative risk aversion measure computed by $-\frac{U''(x)x}{U'(x)}$. Also denote the *market price of risk* $\theta = \frac{\mu - r}{\sigma}$. Then the optimal trading strategy at time t is given by

$$\pi(t) = \frac{\theta}{R\sigma} \tag{10}$$

In section 4 we will demonstrate quantitatively this strategy's supremacy. Note that in this case the optimal portfolio is achieved by keeping the amount of money in the stock

[3]Not all RVs B are possible of course, and the search space is dictated by a theorem that specifies all the reasonable target terminal wealth values.

at a constant ratio compared to the total wealth. This trading strategy is not so simple as it looks since it demands to re-balance our portfolio at every given moment so that the fraction of wealth in the stock will be constant. (As opposed to constant $\phi(t)$ which symbols no transactions during the trading period.)

3. Simulating Portfolios with Monte Carlo

In this section we will discuss how to evaluate the value of $\mathbb{E}\left[U\left(X\left(T\right)\right)\right]$ given all the related parameters under the model discussed in 1.2. We will begin by speaking broadly on simulating SDEs, and then show explicit formula for simulating the SDEs important to portfolio evaluation.

3.1. Simulating SDEs

Consider the general SDE given by

$$\mathrm{d}S\left(t\right)=\alpha\left(t,S\left(t\right)\right)\mathrm{d}t+\beta\left(t,S\left(t\right)\right)\mathrm{d}W\left(t\right) \tag{11}$$

together with the nonrandom initial condition

$$S\left(0\right)=S_0 \tag{12}$$

where $W\left(t\right)$ is a Brownian motion.

Under mild conditions on the functions α and β, the problem specified by 11 and 12 has a unique solution $S\left(t\right)$. A random process which is a solution to problem like this is usually called a *diffusion process*, and β is called the coefficient of diffusion.

We would like devise a method in this section to sample paths of this process $S\left(t\right)$. The most common way is to use the celebrated *Euler's Method*. For this method, we consider a time period we are interested in the value of S, $[0,T]$, and discretize it to M points $\{t_n\}_{n=0}^{M-1}$ such that

$$0=t_0<t_1<\cdots<t_{M-1}=T.$$

The points are usually chosen such that $t_{n+1}-t_n\triangleq\Delta t$ is constant for all $0\leq n<M-1$. We approximate equation 11 for every $0\leq n<M-1$ as follows

$$\mathrm{d}S\left(t\right)\approx S\left(t_{n+1}\right)-S\left(t_n\right)$$

and

$$\begin{aligned}\alpha\left(t,S\left(t\right)\right)\mathrm{d}t+\beta\left(t,S\left(t\right)\right)\mathrm{d}W\left(t\right)\approx&\\ \alpha\left(t_n,S\left(t_n\right)\right)\Delta t+\beta\left(t_n,S\left(t_n\right)\right)&\left(W\left(t_{n+1}\right)-W\left(t_n\right)\right)\end{aligned}$$

Using the more compact notation $S\left(t_n\right)\triangleq S_n$ we get the approximated SDE

$$S_{n+1}-S_n=\alpha\left(t_n,S_n\right)\Delta t+\beta\left(t_n,S_n\right)\left(W\left(t_{n+1}\right)-W\left(t_n\right)\right) \tag{13}$$

together with the initial condition on S_0.

Since α and β are known functions and Δt is chosen by us, we are left to determine how to deal with the term $\left(W\left(t_{n+1}\right)-W\left(t_n\right)\right)$. Fortunately, by the definition of the Brownian motion, we know that the differences between any two points $0\leq s<t$, $W\left(t\right)-W\left(s\right)$ are normally distributed with mean 0 and variance $t-s$, and are also independent from

differences in different times. Thus the differences $W(t_1) - W(t_0), W(t_2) - W(t_1), \ldots$ can be substituted by *independent* normal random variables with mean 0 and variance Δt.

To conclude, in order to simulate a path given by the dynamics in 11 and the initial condition 12 over the interval $[0, T]$, we divide the interval into M points $t_0, \ldots, t_{M-1}$, and sample $M-1$ standard normal random variables $\{\varepsilon_n\}_{n=1}^{M-1}$. We set S_0 by the initial condition, and compute $S_1, S_2, \ldots, S_{M-1}$ by the recursive formula

$$S_{n+1} = S_n \alpha(t_n, S_n) \Delta t + \beta(t_n, S_n) \sqrt{\Delta t} \cdot \varepsilon_{n+1} \tag{14}$$

This scheme is also known in the literature as the *Euler-Maruyama scheme*.

3.2. Simulating the Assets and the Wealth

Armed with this knowledge, we can proceed to estimate paths of the stock price, the money market account and the wealth process. In the following we will use similar notation as used above

$$r(t_n) = r_n \quad \mu(t_n) = \mu_n \quad \sigma(t_n) = \sigma_n$$

The Money Market Account For the market money account, it is common to choose the initial condition $P_0(0) = 1$. Then, its SDE (equation 1) can be approximated as

$$P_{0,n+1} = P_{0,n} + r_n P_{0,n} \Delta t = P_{0,n}(1 + r_n \Delta t) \tag{15}$$

This specific recursion can be solved with the initial condition, yielding the relation

$$P_{0,n} = \prod_{k=0}^{n-1} (1 + r_k \Delta t)$$

Note that if we are given the sequence $\{r_n\}$ in advance, we can calculate the whole path for P_0 using a simple cumulative product operation as NumPy's `cumprod`.

The Stock For the stock price, its SDE (equation 4) can be approximated as

$$\begin{aligned} P_{1,n+1} &= P_{1,n} + \mu_n P_{1,n} \Delta t + \sigma_n P_{1,n} \sqrt{\Delta t} \varepsilon_{n+1} & (16) \\ &= P_{1,n} \left(1 + \mu_n \Delta t + \sigma_n \sqrt{\Delta t} \varepsilon_{n+1}\right) & (17) \end{aligned}$$

where $\{\varepsilon_n\}_{n=1}^{M-1}$ is a sequence of independent standard normal RVs. This recursion can also be solved with the initial condition

$$P_{1,n} = P_1(0) \prod_{k=0}^{n-1} \left(1 + \mu_k \Delta t + \sigma_k \sqrt{\Delta t} \varepsilon_{k+1}\right)$$

So given the sequences $\{\mu_n\}$ and $\{\sigma_n\}$ in advance (we can simulate $\{\varepsilon_n\}$ in advance obviously), we can calculate the whole path for P_1 using cumulative product as in the money market account case.

The Wealth Process The wealth process satisfy the SDE given by 7, which we will write now again for further development

$$\mathrm{d}X\left(t\right)=\phi\left(t\right)\mathrm{d}P_{1}\left(t\right)+\left(X\left(t\right)-\phi\left(t\right)P_{1}\left(t\right)\right)r\left(t\right)\mathrm{d}t$$

This equation is approximated by the recursive formula

$$X_{n+1}=X_{n}+\phi_{n}\left(P_{1,n+1}-P_{1,n}\right)+\left(X_{n}-\phi_{n}P_{1,n}\right)r_{n}\Delta t$$

We can substitute

$$\phi_{n}=\pi_{n}\frac{X_{n}}{P_{1,n}}$$

and get an equivalent formula

$$X_{n+1}=X_{n}+X_{n}\pi_{n}\frac{P_{1,n+1}-P_{1,n}}{P_{1,n}}+\left(X_{n}-\pi_{n}X_{n}\right)r_{n}\Delta t \tag{18}$$

$$=X_{n}\left(1+\pi_{n}\frac{P_{1,n+1}-P_{1,n}}{P_{1,n}}+\left(1-\pi_{n}\right)r_{n}\Delta t\right) \tag{19}$$

Thus again, given the initial wealth X_0 we can write an explicit formula for the approximated wealth process

$$X_{n}=X_{0}\prod_{k=0}^{n-1}\left(1+\pi_{k}\frac{P_{1,k+1}-P_{1,k}}{P_{1,k}}+\left(1-\pi_{k}\right)r_{k}\Delta t\right)$$

and given $\{\pi_n\}$ and $\{P_{1,n}\}$ and $\{r_n\}$ we can calculate the whole path for X_n with a cumulative product operation. This time however we should note that $\pi\left(t\right)$, in general, can depend on the value of the portfolio at time t, thus assuming we have $\{\pi_n\}$ before calculating $\{X_n\}$ might not be feasible. For the case of constant market and CRRA, the optimal $\pi\left(t\right)$ is constant and thus doesn't depend on the portfolio value.

3.3. Monte Carlo Portfolio Evaluation

Now that we know how to sample assets prices paths and how to compute the wealth process out of them, we can devise a general Monte Carlo algorithm for estimating $\mathbb{E}\left(U\left(X\left(T\right)\right)\right)$.

1. First determine the splitting of the interval $[0,T]$ to M points.
2. Fix the number of samples that will be used for the estimation N.
3. Initialize sum $=0$.
4. For N iterations:
    a) Sample paths of $P_0\left(t\right)$ and $P_1\left(t\right)$ of length M according to 15,17.
    b) Compute the path of $\pi\left(t\right)$.
    c) Compute the path of $X\left(t\right)$ according to 19.
    d) sum $=$ sum $+\,U\left(X\left(T\right)\right)$.
5. Return $\frac{\text{sum}}{N}$.

4. Results

In this section we show some results achieved with this simulation method. The equations from the previous section are implemented in the PortOpt library discussed in the appendix.
A constant market model is considered with typical parameters

$$r = 0.01 \left[\text{years}^{-1}\right] \qquad \mu = 0.1 \left[\text{years}^{-1}\right] \qquad \sigma = 0.2 \left[\text{years}^{-\frac{1}{2}}\right]$$

The investing period is over a year $T = 1$ [years] with $\Delta t = \frac{1}{365}$ [years], i.e, a day.
Two utility functions will be examined:

1. $U(x) = \ln x$ and
2. $U(x) = x$ – not a valid utility according to the definition.

We assume, for simplicity that both the money market account and the stock have initial price of \$1. Also our investor starts with initial wealth of \$1. This helps to determine the final relative gain easily

$$\text{Relative Gain} = \frac{X(T) - X(0)}{X(0)} = X(T) - 1$$

3 trading strategies will be examined:

1. Constant Strategy For this strategy we fix $\pi(t) \equiv \pi$. For the sake of comparison we will check $\pi = \{0.5, 1, 4\}$ corresponding to: a) 50% investment in the stock, b) 100% investment in the stock and c) 400% investment in the stock[4].

2. Best So Far Strategy This strategy picks at every moment the best performing asset so far (in terms of relative gain), that is

$$\pi_n = \begin{cases} 1 & P_{1,n} \geq P_{0,n} \\ 0 & \text{else} \end{cases}$$

(Remember that for both assets initial price is 1, so the inequality above is actually

$$\frac{P_{1,n} - P_{1,0}}{P_{1,0}} \geq \frac{P_{0,n} - P_{0,0}}{P_{0,0}}$$

The intuition behind this strategy is that it is very immune against bad scenarios where the stock performs poorly. In those cases the strategy will be to invest all the wealth in the money market account and thus ensure steady gain.

[4] Allowed by borrowing money from the money market account

3. Optimal Oracle Strategy We will use equation 10, that uses a constant portfolio as well. However, for the current market, the market price of risk[5]

$$\theta = \frac{\mu - r}{\sigma} = 0.45 \left[\text{years}^{-\frac{1}{2}}\right]$$

We will use the relative risk aversion of the logarithm function, which is exactly $R = 1$. Thus the optimal portfolio is given by

$$\pi = \frac{\theta}{R\sigma} = 2.25$$

Note that $\pi > 1$, that is, under these market conditions, the optimal strategy is to *borrow* money from the money market account and invest it in the stock as well[6].

An important point should be made about the optimal strategy – it is based on the market parameters r, μ and σ. In practice, the interest rate r is usually known, but μ and σ are not. Thus a "real" strategy should also estimate those parameters based on current and past data. The problem of parameter estimation in diffusion process is intricate and will not be discussed in this work. For this reason we called this strategy an "oracle" as it knows the unknown parameters.

$N = 100,000$ iterations were done and the results are presented in the tables 1 and 2.

	Const. 50%	Const. 100%	Const. 400%	Best So Far	Opt. CRRA = 1
mean	1.0568	1.1058	**1.4521**	1.0693	1.2386
std	0.0003	0.0007	0.0043	0.0006	0.0019
relative error	0.0003	0.0006	0.0030	0.0006	0.0015

Table 1: Expected terminal utility for the $U(x) = x$ utility

For the $U(x) = x$ utility, the strategy that brought the highest gains is the Constant 400% stock strategy with average gains of 45%, almost two times more than the second place – the optimal CRRA strategy – that brought average gains of 23%.

[5] A good intuition for the market price of risk is the *advantage* of the stock over the money market account. We first calculate how much μ is greater than r, but divide it by σ. So if there is a lot of volatilty, the market price of risk will be smaller.

[6] It is also interesting to note that when $\mu < r$, then $\theta < 0$ and thus $\pi < 0$ (as $\sigma > 0$ surely). That is we short the stock (sell borrowed stocks) and invest this money in the money market account.

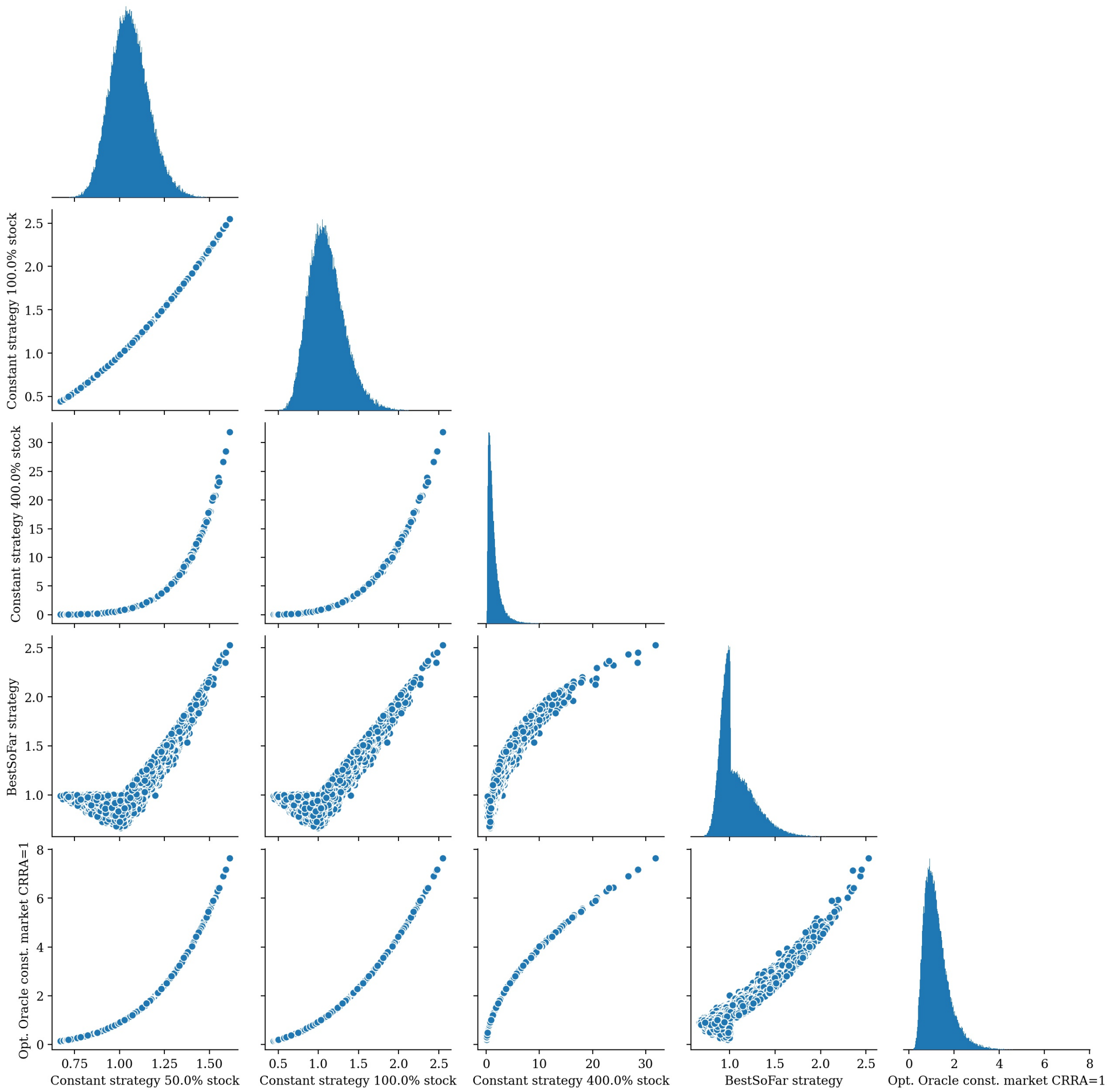

Figure 1: Pair plot of the expected terminal utility for the $U(x) = x$ utility

Figure 1 shows a *pair plot* for the terminal utilities. In each diagonal cell a histogram for a strategy is shown and for non-diagonal cells 2D scatter plots are shown. All the strategies show positive correlations. That makes sense since they are all trying to maximize (in some way) the terminal wealth, thus when one of the strategies has a good day, so are the others.

	Const. 50%	Const. 100%	Const. 400%	Best So Far	Opt. CRRA = 1
mean	0.0502	0.0805	0.0523	0.0523	**0.1125**
std	0.0003	0.0006	0.0025	0.0005	0.0014
relative error	0.0063	0.0079	0.0484	0.0101	0.0127

Table 2: Expected terminal utility for the ln utility

In the case of the ln utility, we can see the supremacy of the optimal strategy, that achieves on average more than twice the utility than most of the other strategies. This time we see the constant 400% strategy performs even worse than the constant 100% strategy in terms of utility, as it is more risky.

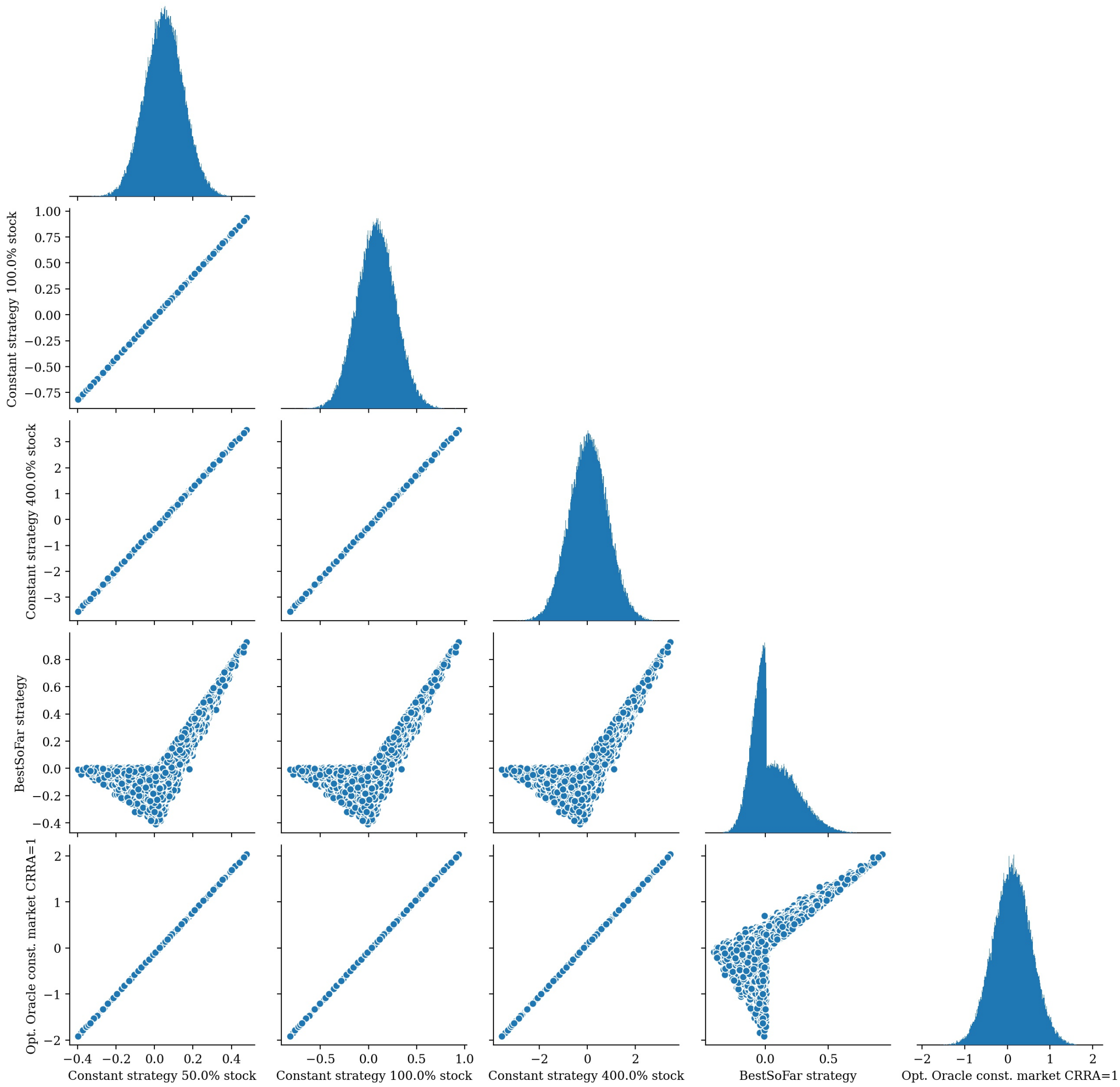

Figure 2: Pair plot of the expected terminal utility for the ln utility

Figure 2 shows the pair plot for the expected terminal utility. Interesting to see is that now terminal utilities are very correlated under the ln utility.

5. Conclusions and Future Work

In this paper we have considered the continuous-time model of capital markets containing a risk-less asset and a stock. In this setting we defined the problem of portfolio optimization. We showed how the expected terminal utility of a portfolio can be approximated using Monte Carlo methods and provided a novel library for doing such computations efficiently. Finally we have demonstrated some experiments to confirm known results on optimal portfolios.

But our story is yet to be concluded. There are many trivial and non-trivial extensions that can be added to this work.

5.1. Trivial Extensions

One trivial extension is to consider a market with *multiple* stocks. The stocks prices can be modeled by a more general SDE, e.g.

$$\mathrm{d}P_i\left(t\right) = P_i(t) \cdot \left(\mu_i\left(t\right)\mathrm{d}t + \sum_{j=1}^{d} \sigma_{ij}\left(t\right)\mathrm{d}W_j\left(t\right)\right)$$

where $W\left(t\right) \triangleq \begin{pmatrix} W_1\left(t\right) & W_2\left(t\right) & \ldots & W_d\left(t\right) \end{pmatrix}$ is a d-dimensional Brownian motion. Note that through the matrix $\sigma_{ij}\left(t\right)$ the stock prices can be correlated.

The Portfolio Optimization problem can be extended too by expanding the goal not just to maximize the expected terminal utility, but also by allowing our investor to *consume* some of its wealth during the investing period. This is usually modeled by an instantaneous consumption process $c\left(t\right)$, that is plugged into the wealth process SDE in the following manner

$$\mathrm{d}X\left(t\right) = \phi\left(t\right)\mathrm{d}P_1\left(t\right) + \left(X\left(t\right) - \phi\left(t\right)P_1\left(t\right)\right)r\left(t\right)\mathrm{d}t - c\left(t\right)\mathrm{d}t$$

The optimal portfolio problem is now to maximize

$$\mathbb{E}\left[\int_0^T U_1\left(t, c\left(t\right)\right)\mathrm{d}t + U_2\left(X\left(T\right)\right)\right]$$

where U_2 is a utility function like we have defined in section 2.1 and U_1 is a time-dependent utility function, i.e. for every $t \in [0, T]$ $U_1\left(t, \cdot\right)$ is a utility function in the normal definition. The search space is now over the trading strategies and consumption processes $\left(\pi\left(t\right), c\left(t\right)\right)$.

5.2. Non-Trivial Extensions

One assumption we made that was very unrealistic yet very simplifying is of a friction-less market – a market with no transaction costs. In practice, every trade has a cost (beside the money for the assets) and even the simplest strategies (recall the constant π strategy) are actually trading at every time instance. Such frequent trading can lead to huge transaction costs that can eventually ruin the investor. Models that account for these costs are facing a crucial trade-off at every moment, between achieving a better position by re-balancing the holdings, or saving money by avoiding another transaction. The methods for solving the Optimal Portfolio problem with transaction costs is discussed more thoroughly in chapter 5 of [Korn, 1997].

An important problem has visited us shortly when we discussed the optimal trading strategy in the previous section. The optimal strategy was built upon the fact that the trader *knows* the market parameters μ and σ. Of course in a real situation that is not the case at all. Hence the trader needs to estimate these parameters from current stock prices and historical prices as well. The problem of parameter estimation in stochastic differential equations is a discipline of its own and thus wasn't discussed in this paper. A future direction for this work is to incorporate methods of parameter estimation in order to develop more realistic tools that can be used in practice.

A. The PortOpt Library

All the results and simulations in this report where created with the `portpot` library. The `portopt` library (from the term `portfolio-optimization`) is a dedicated Python library available on GitHub (Hadayo [2020]) that was written specifically for this project, but allows for descent modifications such that it can work for different scenarios. Since this library is dedicated to simulations, a big effort was put into its performance.

We will now describe the native objects of this library and explain how to use them together in order to create the simulations seen in this work. An examples of usage can be found in the examples directory of the repository in the file `ex1.py`.

The library is aimed to supply two ways of working with it. The first way we call the *online* mode, that may be used for real-time trading. All the objects beside the `Simulator` object can work in an online fashion. In the online mode, every time step we need to call the `step` method in the object we are updating with their needed data.

The second way is the *batch* mode, and it is the preferred way for doing simulations as it as roughly 100 times faster then the online mode. In the batch mode we supply the objects (or sample from them) *all* the data over the desired trading period. In this fashion, optimized vector operations can be applied and performance is increased compared to the online way. To work in batch mode, one needs to use the `sample_path` or `compute_path` function, depending on the object, and supply the needed information.

In this library, the initial stock price, the initial money market account value and the initial wealth of the trader is assumed to be $1.

A.1. Simulating the Market

In order to simulate a market (a money market account and a stock), we are using two objects: the `MarketModel` and the `MarketSimulator`.

A.1.1. The Market Model

The `MarketModel` is responsible for supplying the market parameters processes $r(t), \mu(t)$ and $\sigma(t)$. It can work in two modes: online and batch.

The online mode is invoked when using the `step` function, and the object returns the parameters for the time step we are currently at.

The batch mode is invoked when using the `sample_path` function, and the object returns a path for the three processes. This function also receives the horizon T in order to determine the number of points in the path.

The library has right now one market model and it's the `ConstantMarketModel` which implements a market with constant parameters over time.

A.1.2. The Market Simulator

The MarketSimulator is responsible for simulating the assets processes, i.e. the money market account and the stock price. Every MarketSimulator has a MarketModel that supplies the required parameters. The MarketSimulator simply uses this parameters in the dynamics equations describes in section 3.2.

The MarketSimulator implements the online mode and the batch mode of operation via the step and the sample_path functions respectively.

A.2. Simulating Traders

Simulating traders has a similar structure to the market simulation problem. A trader is simulated using two objects: a Strategy and a Trader object.

A.2.1. The Strategy

The Strategy is responsible for deciding the fraction of the wealth that should be invested in the stock for every time instance, $\pi(t)$. It implements as well the two modes of operation with the step function for online mode and and the compute_path for batch mode[7]. In the two modes the strategy gets an information variable, that holds the visible information for the trader: the asset prices and the interest rate. In addition, it may choose to get a secret_information variable that holds information that is not available to a real trader like $\mu(t)$ and $\sigma(t)$. This option is used in order to test "Oracle" strategies.

If the user decides his strategy should receive secret information it should declare that in the step and compute_path functions (as in the ConstantCRRAOracleStrategy class). Else, it should just put *args in his functions declaration for compatibility with other modules.

The library has right now 3 implemented strategies:

1. ConstantStrategy – which has constant, predetermined π for all times.
2. BestSoFarStrategy – which chooses at each moment the asset with the highest relative gain so far.
3. ConstantCRRAOracleStrategy – which implement the optimal strategy for a constant market with a CRRA utility.

A.2.2. The Trader

The Trader is responsible for simulating the wealth process. It also keeps track of the position in the assets (how many shares in each asset, $\phi(t)$ for the stock). Each Trader has a Strategy the he acts according to. It implements the two modes of operation

[7]Note that strategies may depend on the wealth process, which is not the case for the three given strategies. Thus the Strategy might compute the wealth process on the way by itself.

with the `step` function for online mode and and the `compute_path` for batch mode. It uses the equations from section 3.2 to implement the dynamics of the wealth process.

A.3. Evaluating Portfolios

The `Simulator` is responsible for estimating the expected terminal utility for traders. It supports simulation of a single market model and a single utility function with many different traders.

The `Simulator` is initialized with a list of `Traders`, a `MarketSimulator`, the desired utility function and a directory path for log files (if desired). It has two important functions: `sample` and `simulate`.

The `sample` function samples a path from the `MarketSimulator` and computes a wealth process path for each of the traders. This function returns the final portfolio values (not the utilities!) for each trader.

The `simulate` function calls the `sample` function many times (as specified by the users) in order to create a dataset for estimation of the expected value. It then calls the `print_results` method which prints the results to the user and also saves them to the log file if specified.

The results printed are the average terminal utility for each trader (which is an estimator for the expected value), the standard deviation of the estimator and the relative error of the estimator.

Another function for displaying results is the `display_histograms` function, that prints a *pair plot*[8] of the terminal utilities to the user.

[8] Search for Seaborn Pairplot for more information.

References

Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, 1973.

Amir Hadayo. Portopt, a python library for simulating portofolios. https://github.com/Hadayo/MC-Portfolio-Optimization, 2020.

Ioannis Karatzas, John P Lehoczky, and Steven E Shreve. Optimal portfolio and consumption decisions for a "small investor" on a finite horizon. *SIAM journal on control and optimization*, 25(6):1557–1586, 1987.

Ralf Korn. *Optimal portfolios: stochastic models for optimal investment and risk management in continuous time*. World Scientific, 1997.

Ralf Korn and Elke Korn. *Option pricing and portfolio optimization: modern methods of financial mathematics*, volume 31. American Mathematical Soc., 2001.

Harry M Markovitz. *Portfolio selection: Efficient diversification of investments*. John Wiley, 1959.

Robert C Merton. Lifetime portfolio selection under uncertainty: The continuous-time case. *The review of Economics and Statistics*, pages 247–257, 1969.

Steven E Shreve. *Stochastic calculus for finance II: Continuous-time models*, volume 11. Springer Science & Business Media, 2004.