Modeling the Stock Market: Simulation and Optimization

Cambron Johnson, Nate Schwindt, Zibo Wang

17 April 2019

1 Abstract

In this study, we consider the problem of selecting stocks to create a portfolio. We detail a simulation of future stock prices that uses drift and volatility parameters obtained from historical data and compare these predicted values to reality. We also experiment with portfolio optimization by choosing the best stocks based on balancing expected rates of return with the risk associated with them. Lastly, we explore a simulated scenario where adjustments are made to the portfolio based on monthly predictions.

2 Introduction

We acquired historical price data for each stock from Yahoo Finance's database, exporting the data to CSV files to obtain a year's worth of daily prices for our desired stocks. We subsequently feed these data to the model to generate expected rate of returns. Optimization of the model outputs the recommended percentages of the portfolio for each stock. After simulating our stock market, we choose optimal risk tolerance values and examine our experimental results for our overall monetary investment after ten months. While the model provides a decent approximation of expected returns, it misrepresents major contributors in portfolio selection, such as diversification and realistic randomness of the stock market. Without adjustment, the model does not account for lowered risk in diversified portfolios, and it approximates the random term as a standard normal variable, which may not be appropriate in reality. We explore these limitations, but we ultimately employ the model as a foundation for portfolio selection.

3 Model

3.1 Stochastic Model for Stock Prices

The model for the stock prices as a function of time is a stochastic differential equation in terms of two parameters - the drift parameter μ and the volatility parameter σ. By applying two assumptions regarding the behavior of the stock price with respect to each parameter individually, a discrete simulation for future prices can be found. The first assumption is that without volatility, the growth is a result of continually compounded interest. Secondly, without deterministic drift, the relative change in a stock's price due to its volatility is normally distributed with variance $\sigma^2 \Delta t$. These assumptions result in the approximation used to model future stock prices

$$P(t + \Delta t) \approx P(t) + \mu P(t)\Delta t + \sigma P(t)\sqrt{\Delta t}\phi. \tag{1}$$

With this equation, we set the starting price $P(0)$ as the price in the latest date in the historical prices data. The volatility parameter σ is the standard deviation of the historical daily return rate, and the drift parameter μ is the average daily return rate. Using $\phi\ N(0, 1)$, we predict the price over a time.

3.2 Portfolio Optimization

The goal of portfolio optimization is selecting stocks that will yield as high of a return with as little risk as possible. When building a model for portfolio optimization, we define a parameter that captures the risk tolerance of the investor, α. α can be varied between 0 and 1, with 0 corresponding to lowest risk tolerance and 1 corresponding to highest risk tolerance in favor of high return potential. As a result, this model assumes that to yield high return, there must be higher risk, and conversely, to maintain low risk, there must be lower return. The weighted risk in the portfolio is

$$w^T C w, \tag{2}$$

and the expected return in the portfolio is

$$rw. \tag{3}$$

In these expressions, w is a vector of the weights invested in each stock, and the weight of an individual stock is defined as the ratio between the price of a share multiplied by the number of shares purchased and the total dollars invested in the portfolio. C is the covariance of the matrix of daily returns, and it represents the risk associated with co-investing in stocks. r is the vector of expected returns, and it is expressed as the mean of the historical daily returns multiplied by the time step (1 day).

The model minimizes the difference between the risk and return, such that the portfolio chooses stocks that give high enough return at a low enough risk. Adjusting the difference to be weighted based on the investor's risk tolerance gives the optimization problem

$$\min[(1 - \alpha)w^T C w - \alpha r w]. \tag{4}$$

We use the quadprog function in MATLAB to optimize the portfolio selection (similar to the one seen in Section 4.3). After applying the optimization, the appropriate percentage of stocks is determined by considering the weight distribution of the portfolio. An example optimization was performed, and the corresponding weight distributions are given in Figure 1 below.

Historical		1 Month		2 Month		3 Month		4 Month	
ATVI	0.12	ATVI	0.12	BILI	0.20	CCCL	0.14	ASHR	0.01
BILI	0.02	BILI	0.02	CCCL	0.14	MCD	0.02	BILI	0.20
CCCL	0.04	CCCL	0.03	MCD	0.01	MSFT	0.09	CCCL	0.14
CEA	0.03	CEA	0.03	MSFT	0.20	TSLA	0.13	LFC	0.03
GOOGL	0.04	GOOGL	0.04	NFLX	0.07	VZ	0.19	MCD	0.20
MA	0.04	MA	0.04	VZ	0.19	WMT	0.20	MMM	0.01
NFLX	0.20	NFLX	0.20	WWE	0.20	WWE	0.20	VZ	0.01
NTDOY	0.05	NTDOY	0.05	All others	0.01	All others	0.02	WMT	0.20
NTES	0.07	NTES	0.07					WWE	0.20
SINA	0.03	SINA	0.03					All others	0.02
SPOT	0.06	SPOT	0.06						
TM	0.01	TM	0.01						
TSLA	0.20	TSLA	0.20						
WMT	0.09	WMT	0.10						
All others	0.01	All others	0.01						

Figure 1: The weight distributions of an optimized portfolio at time = 0 and after 1, 2, 3, and 4 months of simulated prices. A new optimization was run after each month had simulated prices, basing the simulation on the previous 12 months.

4 Results

4.1 Comparison with Reality

To assess the credibility of our model, we compared six months of actual data with simulated prices given by the model. Since we obtained a year's worth of data for each stock, we input the first six months of data and predict the prices over the next six months. Each time running the simulation gives a different result, but none were close to matching the actual data. Figure 2 provides an example of one run of the model comparison with actual prices on the left and simulated prices on the right. In general, the simulated prices remained more constant than the actual prices over the six month span, which supports the claim that the model does not appropriately account for realistic random effects.

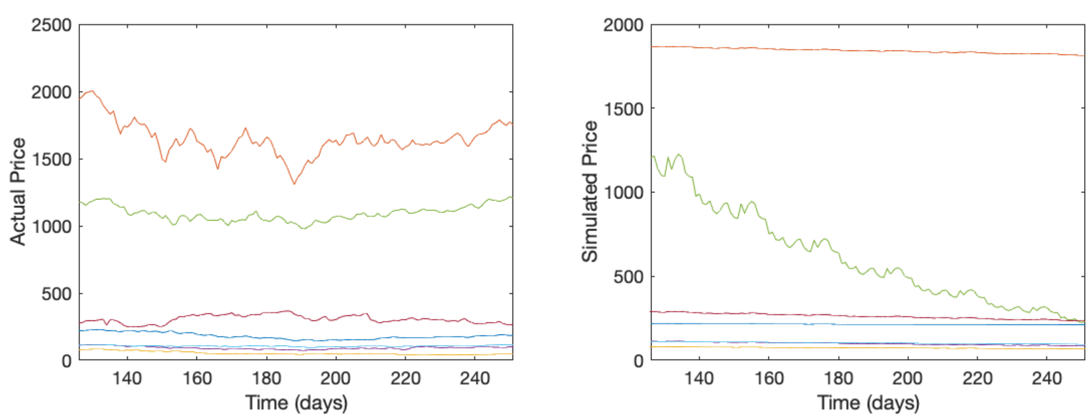

Figure 2: Actual Prices of Stocks vs Simulated Prices of Stocks

4.2 Simulating Expected Rate of Return and Portfolio Selection

In order to ensure diversification, we restricted the stock weights to maximum percentages of 10%, 20%, and 50%. Three different expected return values were generated for the team's chosen stocks. Figure 3 displays the expected rate of return with a 10% restriction over a range of risks, while Figures 4 and 5 display 20% and 50% caps, respectively.

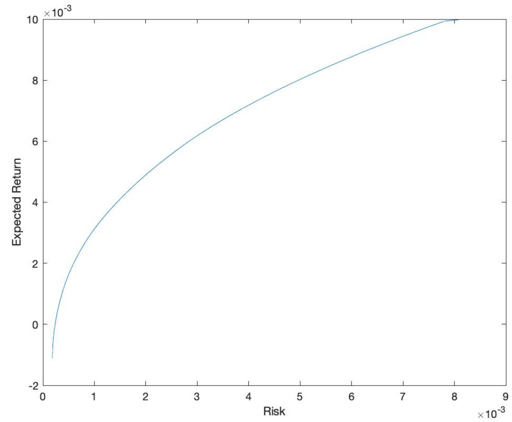

Figure 3: Expected Rate of Return with 10 Percent Cap

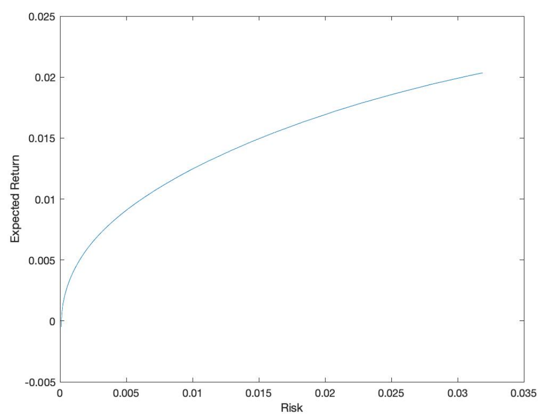

Figure 4: Expected Rate of Return with 20 Percent Cap

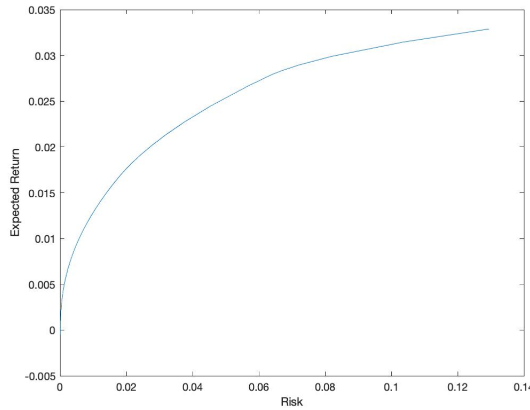

Figure 5: Expected Rate of Return with 50 Percent Cap

As the cap for maximum portfolio percentage is increased, the expected return also increases. The model tends to pick stocks like Google and Tesla when the portfolio percentage restriction is higher. This trend is reasonable since Google and Tesla are growing, profitable companies with expensive stocks and high potential return. The model will put as much weight on these stocks as is allowed, which is a result of the model neglecting the importance of diversification without the deliberate restrictions.

The point of diminishing returns, where risk and return are equal, is the point where slope of the return versus risk plot is one. This point represents the optimal risk tolerance value for the scenario. We expect increasing amounts of risk to increase our expected returns. Likewise, the law of diminishing returns states that, at some point, adding one more unit of risk will yield lower incremental returns. The optimal risk tolerance α turns out to be about 0.707 for our example simulation.

4.3 Simulating the Stock Market

Following MATLAB script simulates stock market, and optimizes the portfolio each month based on prices from the previous year's history:

```
clear all
clc
%% 10 months simulation
M(1) = 1000; % initial money

%% read adj. close price
ds = tabularTextDatastore('\','FileExtensions','.csv');
% path of csv file
T = readall(ds);
s=44; % num of stock
for i=1:s
      P(:,i)=T((i-1)*251+1:i*251,4);
end
```

```
P=table2array(P);

%% optimize portfolio for first month
R=(P(2:end,:)-P(1:end-1,:))./P(1:end-1,:);
r=mean(R);
C=cov(R);
alpha=0.2;
[w(:,1),optVal(1)]=
quadprog((1-alpha)*2*C,-alpha*r,[],[],
        ones(1,s),[1],zeros(s,1),0.2*ones(s,1));
S(:,1) = (M(1)*w(:,1))./(P(251,:)');
% money spend on each stock for first month

%% simulation stock market and optimize portfolio for following months
dt=1/251;
phi=randn(30,1);
for i=1:44
    sigma(i)=std(R(:,i));
end
for k=1:10
    for j=1:30
        P(251+j+30*(k-1),:)=
        P(251+j-1+30*(k-1),:)+r.*P(251+j-1+30*(k-1),:)*dt
        +sigma.*P(251+j-1+30*(k-1),:)*sqrt(dt)*phi(j);
    end
    R=(P(2+30*(k-1):end,:)-P(1+30*(k-1):end-1,:))./P(1+30*(k-1):end-1,:);
    r=mean(R);
    C=cov(R);
    [w(:,k+1),optVal(k+1)]=
    quadprog((1-alpha)*2*C,-alpha*r,[],[],
            ones(1,s),[1],zeros(s,1),0.2*ones(s,1));
    M(k+1)=S(:,k)'*P(end,:)';
    S(:,k+1) = (M(k)*w(:,end))./(P(end,:)');
end
```

For maximum portfolio percentage values of 20%, with risk tolerance $\alpha = 0.2$, a typical simulation would have results like:

Table 1: Monetary Investment Example over 10-Month Span with $\alpha = 0.2$

month	0	1	2	3	4
money	1000	974.46	974.78	965.35	964.21
5	6	7	8	9	10
952.45	954.63	943.24	944.56	934.63	937.37

As discussed in Section 4.2, the optimal risk tolerance for this example is approximately

0.707, and we found this value using the script:

```
clear all
clc

%% read adj. close price
ds = tabularTextDatastore('\','FileExtensions','.csv');%path of csv file
T = readall(ds);
s=45;
for i=1:s
    P(:,i)=T((i-1)*251+1:i*251,4);
end
P=table2array(P);
R=(P(2:end,:)-P(1:end-1,:))./P(1:end-1,:);
r=mean(R);
C=cov(R);

%% find the point of diminishing return
alpha=[0:0.01:1].^2;
for i=1:length(alpha)
    [w(:,i),optVal(i)]=
    quadprog((1-alpha(i))*2*C,-alpha(i)*r,[],[],
            ones(1,s),[1],zeros(s,1),0.2*ones(s,1));
end
for i=1:length(alpha)
    wcw(i)=w(:,i)'*C*w(:,i); % risk
end
dw=wcw./(r*w); % (r*w) is the expected return
plot(alpha,dw)
```

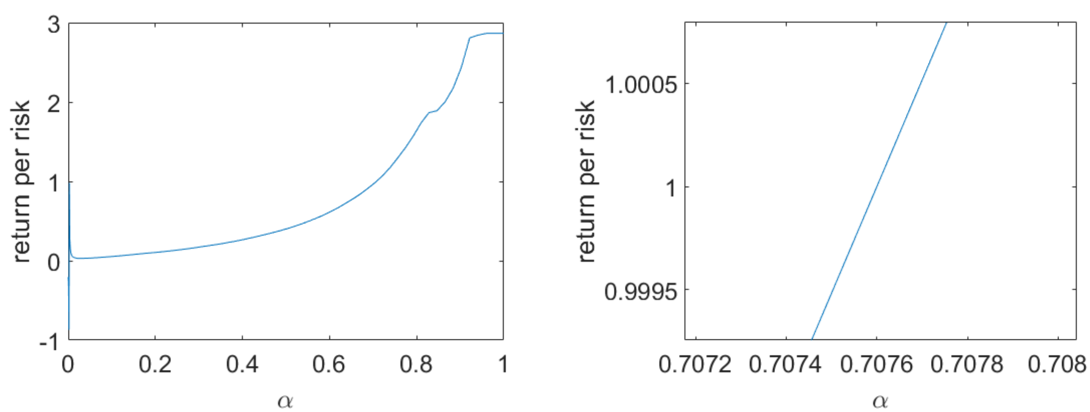

Figure 6: Return per risk vs. risk tolerance α

After adjusting the risk tolerance accordingly, the resulting monetary values for this example drastically improved as shown below. However, when running the simulation, occasionally the final total would be much lower, approximately \$700 since we increased the risk. Similarly, the final total had the potential to exceed $\$10,000$. Increasing the risk tolerance from 0.2 to 0.707 dramatically affects the potential return predicted by the model.

Table 2: Monetary Investment Example over 10-Month Span with $\alpha = 0.707$

month	0	1	2	3	4
money	1000	1202.76	1206.01	1476.07	1609.71
5	6	7	8	9	10
1971.21	1971.21	2703.38	2965.75	3708.76	4069.11

To take a 5% transaction cost into account, we assume that at end of every month, we sell all our stocks and purchase new stocks according to the new recommended portfolio. Instead of starting with all the money we receive from selling our stocks, we change `M(k+1)=S(:,k)'*P(end,:)';` to `M(k+1)=0.95*S(:,k)'*P(end,:)';`. After the adjustment, we are more likely to lose money. The final total for the 10-month simulation typically ends around \$700.

5 Stocks Used

The following table displays all of the stocks chosen for use in our simulations:

Cambron	Apple (AAPL) Microsoft (MSFT) Verizon (VZ) Nintendo (NTDOY) Nvidia (NVDA) Sony (SNE) Seagate Technology (STX) Spotify (SPOT) Amazon (AMZN) Western Digital Corporation (WDC) Tesla (TSLA) Electronic Arts (EA) Activision Blizzard (ATVI) Alphabet (GOOGL) AT&T (T)
Nate	Nokia (NOK) Xtrackers Harvest CSI 300 China A-Shares ETF (ASHR) The Royal Bank of Scotland Group plc (RBS) Papa John's International, Inc. (PZZA) McDonald's Corporation (MCD) Kasikornbank Public Company Limited (KPCPY) World Wrestling Entertainment, Inc. (WWE) The Walt Disney Company (DIS) Netflix, Inc. (NFLX) Honeywell International Inc. (HON) 3M Company (MMM) Walmart Inc. (WMT) Mastercard Incorporated (MA) Toyota Motor Corporation (TM) Target Corporation (TGT)
Zibo	Alibaba (BABA) Aluminum Corporation of China (ACH) Baidu (BIDU) Bilibili (BILI) China Ceramics (CCCL) China Eastern Airlines (CEA) China Life Insurance (LFC) China Southern Airlines (ZNH) Marriott (MAR) NetEase (NTES) New Oriental Education (EDU) Sina (SINA) Sohu (SOHU) Weibo (WB) 51job (JPBS)