Market traders buy and sell volatile assets frequently, with a goal to maximize their total return. Two such assets are gold and Bitcoin. We started with $1,000 on 9/11/2016 and used a 5-year trading period to trade these two assets to maximize our profits.

Six models are developed to provide each day's trading strategy from the up to that day price information. Final worth of total assets generated by each model is compared. Each consecutive model adapts the optimal ideas based on the computational result from previous ones. Our recommendation model, model 6, yields the final worth of $60,063 (including transaction cost) on the day of 9/10/2021. Meanwhile, the transaction cost does not significantly affect our final worth. The transaction cost percentage is around 32%, which is tolerable considering the high profit. In this model, the idea of Moving Average (MA), different buying stages, self-adapted selling thresholds, and diversification of total assets are applied. Such a strategy based on the past price data up to that day, helped us to determine the prospects of the market, changing the thresholds of profit expectation, and fully utilize the opportunity to switch between high-risk-high-profit asset (Bitcoin) and low-risk-low-profit asset (Gold). Machine learning models are considered, whereas the high volatility of the Bitcoin market might not be successfully predicted in a short time.

Our recommendation is supported by real-life trading strategies and computational results from python algorithms. Meanwhile, our criteria for judging models is not only based on the amount of money earned after five years but also the reproducibility and stability for other investments in future modeling.

Contents

1 Model Assumption

We assume a world where investors fully follow exactly one of the approaches below. Furthermore, no additional trading information is available to any of the investors: We did not take into consideration the amount of daily transactions or political issues that may affect asset price.

2 Models

In this solution, we have developed a total of six models. Each model is learning and adapting from the previous one. Details are shown below.

I. Model 1

Our first model uses all \$1,000 invested in both assets (Bitcoin or Gold) combinations on the first day. Meanwhile, we do not perform any transactions during the entire five years, and we look at the final asset values on the last day. Computational results indicate that the optimal strategy under this scenario would be to invest all the assets into Bitcoins. We would sell all our assets on 9/11, 2021. The maximum asset value would be **\$73,097**, and this value takes into account the amount of transaction fees. Algorithm please refer to the Appendix section, “Model 1”. The percentage of the transaction cost is 2%.

II. Model 2

Our second model would be to define two constant thresholds (in hundredth percentile): one positive threshold for determining when to sell the asset, call it t_1, and another negative threshold for determining when to buy the asset, denote it as t_2. Note that for this model, the two threshold values are defined beforehand and used for the whole five year period, and any transactions would not involve any combination of any assets (Cash, Gold, and Bitcoin). Transactions are only performed based on the previous day returns and the pre-defined thresholds. This method can be illustrated in the following example. Suppose t_1 equals 10 percent and we have already purchased assets, then any daily return of the asset that we invested higher than 10 percent suggests that the expectation for the investment is satisfied, and this would result in the sale of all the assets before the next trading period. On the other hand, suppose $t_2 = -10\%$, if we have not purchased any assets (hold them as cash), then a daily return lower than -10% would suggest the current cost is lower than we expected, so this would result in the purchase of assets at the start of the next trading period. Computational results suggested $(t_1, t_2) = (0.76, -0.04)$ yields the maximum return of **$209,936** for the five years (including transaction costs). Algorithm please please refer to the Appendix section, "Model 2".

This transaction method is entirely dependent on the daily return of the assets for the previous day and arbitrary settings of the threshold t_1 and t_2. If we only consider the amount of earnings, the result is remarkably sound: our final benefit is 200 times greater than the initial asset. However, this method is extremely impractical in real life. We have to exactly define our two thresholds to be those designated values, without a signal bit difference. For example, if we change the threshold to be (0.75, -0.04), the output will decrease significantly to approximately $2,000, which shrinks to 1/100 of the maximum return. Thus, the majority of the investors cannot fully base this theory to perform their investments in practical terms.

III. Model 3

Our third model applies the idea of Moving Average_X (MA_X), where X represents the average time period, as well as the Golden Cross and the Dead Cross. A moving average is a calculation used to analyze data points by creating a series of averages of different subsets of the full data set [2]. When a shorter (e.g., 5-day) moving average line crosses over a longer one (e.g., 30-day), from below, while both lines are rising, it is a major buy signal called golden-cross, indicating that the market is in a bull trend; The reverse is called a dead-cross, a sell signal [1]. Generally, larger time periods tend to form stronger lasting breakouts. For example, the daily 50-day moving average crossover up through the 200-day moving average on an index like the S&P 500 is one of the most popular bullish market signals in the real world [1]. However, in our case we are considering daily trading. Day traders commonly use smaller time periods like the 5-period and 10-period moving averages to trade intraday golden cross breakouts.

We calculated MA5, MA10, MA30, respectively. In the real life, when an investor decides to invest in a new asset, one must collect past price data for better investment strategies, which forbids any possible transactions during the first time period. In this case, no transaction is made during the first 30 trading days. We use the Golden Cross as a buy signal, while the Dead Cross as a sell signal. To be more specific, when the MA5 crosses MA10 from the bottom, it is a Golden Cross. This would result in the purchase of assets at the start of the next trading period. Conversely, when the MA5 crosses MA10 from above, it is a Dead Cross, resulting in the sale of all the assets.

Computational Results show that transaction fees made a huge difference between the amount of profits. The maximum output without the transaction fee is $32,096, whereas the output included transaction fee drops to **$587**. It is reasonable since we need to make too many transactions. Thus, this method is very sensitive to the transaction fee rate.

IV. Model 4

Noticing that too many transactions would cause the profit to decrease significantly, we developed our fourth model which reduced the number of transactions using the idea of only Moving Average and expectation threshold. In this model, we purchase assets when the three MA lines (MA5, MA10, and MA30) increase simultaneously. We sell the assets only if it reaches our expected threshold. The expected threshold is defined as a constant beforehand and used for the whole trading period. Algorithm please refer to Appendix “Model 4”.

Same as Model 3, no transactions are made during the first 30 trading days. Computational results give a maximum output of **$52,733** ($78,988 without the transaction fee). Even though this is a high profit, it depends on how volatile a specific asset is. Bitcoin in this case is considered highly volatile, by contrast, the price of the gold is much more stable. However, the profit of buying gold is relatively much lower ($1404 with transaction fee).

V. Model 5

All of our models above are considered concentrated investment. In previous transactions, we used all our resources to purchase one asset type, and sell all the amount at one time. In real life, people usually make investments in different stages based on the performance of the market. People do not make huge transactions when a small price fluctuation occurs. Hence, we decided to invest different fractions of our money in different stages: In the first stage, if MA5 locates above MA10, suggest that the market is in a bull trend, then we would use 20 percent of the cash we owned to buy the specific asset; for the second stage, if MA5 locates above MA10, and at the same time MA10 locates above MA30, which implies the market is in a stronger bull trend, we decide to invest 50 percent of our current cash and purchase in a higher amount. Algorithm please refer to Appendix “Model 5”.

No transactions are made during the first 30 trading days. The result shows a maximum output of **$39695** ($44225 without the transaction fee). Transaction fee in this case is less influential since we only make transactions when a strong bull trend occurs. Moving averages with shorter time lengths tend to fluctuate more often than longer time periods. In our case, when MA5 stands above MA10, the trend may not last long; while MA10 is constantly standing above MA30, the bull trend may be long-lasting, and it is a signal for larger investment.

VI. Model 6

Our final model combines all the ideas of the models above while considering two new aspects: (1) Instead of just considering one asset, we consider combining the two assets (Bitcoin and Gold) together. (2) We consider different selling thresholds based on the performance of the market.

To combine two assets together, we need to consider the trading time of the two markets. Bitcoin can be traded every day, whereas gold is only traded on designated days when the market is open. We need to combine the gold and bitcoin datasets together to see whether we can make any transactions on a specific day. Based on the results of previous models, we know that Bitcoin has a higher possibility of earning more money. In order to maximize our profit, we decide to closely follow up price fluctuations in both markets. When the Bitcoin market is in bear markets, we sell all owned Bitcoin shares and purchase gold depending on whether the gold market is in a bull trend or not. Conversely, If the Bitcoin market is turning good, we sell all gold shares to purchase Bitcoin, due to the volatile nature of Bitcoin, small fluctuations may result in large price shifts. If the gold market is in a bull trend, we can buy gold shares. Otherwise, we just hold this money until the Bitcoin market becomes good. Using this strategy, we can maximize our profit rather than just spending money in the bitcoin market.

We use the MA10, MA30 line to determine the expected threshold. No transactions are made during the first 30 trading days. When MA10 intersects with MA30 from below, this implies the long term market is optimistic and we will sell the shares when the condition of the expected threshold is satisfied. When MA10 intersects with MA30 from above, this implies the long term market is not optimistic and we will sell the shares to prevent further money loss. This strategy perfectly captures better and more appropriate timing. In this final model, we refer to the price line, MA5, MA10 and MA30 of both Bitcoin and gold. First, we use the price of gold and MA5 of gold to decide whether a specific day is appropriate for gold transactions: If value intersects with MA5 from below, it is a buy signal for gold; if value intersects with MA5 from above, it is a bad day to make gold transactions. Similar to Model 5, we applied the idea of partial investment. In terms of Bitcoins, if MA5 stands stable above MA10, we would use 20 percent of the cash we owned to buy the specific asset. If MA10 stands stable above MA30,we decide to invest 50 percent of our current cash and purchase a higher amount. We also define a changing threshold to determine the selling point of Bitcoin. If MA10 intersects MA30 from below, then we think the long term market is promising and the threshold will be adjusted to 300%. If MA10 intersects MA30 from above, the long term market is pessimistic and the threshold decreases to 150%. If the ratio falls below the threshold, we decide to sell the bitcoin to prevent further loss. Then until the next bull trend for Bitcoin, we observe whether the gold market is open and is appropriate for transactions.

Computational results suggested that this method yields the return of **$60,063** ($79323 without the transaction fee), which is the current best strategy. Algorithm please refer to the Appendix section, "Model 6".

3 Comparison

Of all the models, we recommend a rational investor using model 6. Depending on how radical the investor is, the combination of model 5 and model 6 are suggested. Compared to the final model, no transactions are performed during the entire five years in the first model. Even though a huge amount of profit is gained, it is not applicable to rational investors. Also, the huge amount of profit is based on the abnormal behavior of the Bitcoin market, so it is hard to apply this model to any other assets. The second model is also not applicable compared to the models that we recommend. Without data support or the ability to foresee the future, the exact thresholds are nearly impossible to set beforehand. The third model is not recommended due to the high transaction costs. To be more specific, the profit shrinks from $32,096 to $587 due to the high frequency of transactions. For the fourth model, it gives a high profit of $52,733. However, it is hugely dependent on how volatile a specific asset is. For the last two models we recommended, various factors are taken into consideration from the real world stock market operational strategies. We considered different investment stages based on the idea of "Do not put all eggs in one basket". This idea also leads to the diversification of investment into two assets.

4 Transaction Cost

Overall, we noticed that with the increase in transaction times, the transaction costs increased significantly. For various models we listed above, the percentage of transaction costs ranged from 11.4% to 5367%. We conclude that in order to maximize the profit, we should try to decrease the transaction times. For the model that we recommend, the transaction cost is around 32%, which is relatively low compared to the rest models.

5 Memorandum

In our strategy, we mainly use the average price of the asset in the past 5 days, 10 days and 30 days to determine the buying operation and the selling operation. Our recommended model can be relied on because the method of using the moving average has already been widely used in the financial markets for many years. We tested our model on data from 9/11/2016 to 9/10/2021, and the result is convincing. Specifically, we start with $1000 on 9/11/2016, and end with $60,063 on 9/10/2021. Thus, the ratio of profit is 5906%.

However, this astonishing success is because of the abnormal market of bitcoin during 2020 and 2021. This method can be generalized to other assets with careful adjustments for different market behaviors. If the Bitcoin market is in the bear market, then this strategy will not give such a high return on profits. Our recommendation from model 6 might be more suited for radical investors because most of the cash is spent on the volatile Bitcoin market. For conservative investors, we advise to spend about 20-40% of money on method 6 depending on how much risk you are willing to take. For the rest of the money, we recommend investing in the gold market using model 5.

6 Reference

Hayes, Adam. “What Is a Golden Cross?” Investopedia. Investopedia, February 8, 2022. https://www.investopedia.com/terms/g/goldencross.asp.

Fernando, Jason. “Moving Average (MA) Definition.” Investopedia. Investopedia, February 8, 2022. https://www.investopedia.com/terms/m/movingaverage.asp.

7 Appendix: Python code for models

[1]:
```
import pandas as pd
import numpy as np
%matplotlib widget
import matplotlib.pyplot as plt
from tqdm import tqdm
from datetime import datetime
from datetime import timedelta
```

[2]:
```
df_bitcoin = pd.read_csv("../dataset/BCHAIN-MKPRU.csv")
df_gold = pd.read_csv("../dataset/LBMA-GOLD.csv")
```

[3]:
```
"""
Model 1:
    No transactions during five years
    Test different combinations of investment ratio
"""
cash = 1000
share = [0, 0]
best = [0, 0, 0]
for i in range(101):
    share[0] = cash*0.98 * (i/100) / df_bitcoin.iloc[0]['Value']
    share[1] = cash*0.99 * (1-(i/100)) / df_gold.iloc[0]['USD (PM)']
    final_asset = df_bitcoin.iloc[-1]['Value'] * share[0] + df_gold.iloc[-1]['USD (PM)'] * share[1]
    if (final_asset > best[0]):
        best[0] = final_asset
        best[1] = i/100
        best[2] = 1-i/100
best
```

[3]:
```
[73097.91072146707, 1.0, 0.0]
```

[4]:
```
money = 1000 * 0.99
for index in df_gold.index:
    row = df_gold.loc[index]
    percent = row['Return']
    money = money * (1+percent)
```

[5]:
```
money * 0.99
```

[5]:
```
1327.8630982938464
```

[6]:
```
"""
A wrapper function that calculates the final amount of assets given different thresholds as inputs
(For Model 2)
"""
def method2(df, maxthres, minthres):
    money = 1000 * 0.98
    cur = 0
    status = 1
    for index in df.index:
        row = df.loc[index]
        percent = row['Return']
        price = row['Value']
        cur += percent
        if status == 1:
            if cur > maxthres:
                status = 0
                money = money * (1+percent) * 0.98
                cur = 0
            else:
                money = money * (1+percent)
```

```
        if status == 0:
            if cur < minthres:
                status = 1
                cur = 0
    return money
```

[7]:
```
"""
Model 2:
    Define two constant thresholds:
    - one positive threshold for determining when to sell
    - one negative threshold for determining when to buy
"""
max_money = 0
for i in range(0, 100):
    for j in range(-20, 0):
        cur = method2(df_bitcoin, i/100,j/100)
        if cur > max_money:
            max_money = cur
            max = i
            min = j
```

[8]:
```
money, max, min
```

[8]:
```
(1341.275856862471, 76, -4)
```

[9]:
```
"""
Model 3:
    Moving Average, Golden Cross and Dead Cross
"""

money = 1000
status = 0
for index in df_bitcoin.index:
    if index != 1825:
        row = df_bitcoin.loc[index]
        rowcompare = df_bitcoin.loc[index+1]
        value = rowcompare['Value']
        x1 = row['MA5']
        x2 = rowcompare['MA5']
        y1 = row['MA10']
        y2 = rowcompare['MA10']
        if x1 > y1 and x2 < y2 and status == 1:    # Dead Cross, signal for sale
            status = 0
            money = share * value * 0.98
        elif x1 < y1 and x2 > y2 and status == 0:  # Golden Cross, signal for buy
            status = 1
            share = money / value * 0.98
```

[10]:
```
money
```

[10]:
```
587.7797327819005
```

[11]:
```
"""
Model 4:
    Buy Signal appears only when MA5, MA10, MA30 increase simul
    Sell all assets when reached expectation threshold
"""

money = 1000
status = 0
cur = 0
for index in df_bitcoin.index:
    if index != 1825:
        row = df_bitcoin.loc[index]
        rowcompare = df_bitcoin.loc[index+1]
        value = rowcompare['Value']
```

```
        x1 = row['Value']
        x2 = rowcompare['Value']
        y1 = row['MA5']
        y2 = rowcompare['MA5']
        a1 = row['MA10']
        a2 = rowcompare['MA10']
        b1 = row['MA30']
        b2 = rowcompare['MA30']
        if status == 0 and y2 > y1 and a2 > a1 and b2 > b1:      # Buy Siganl
            status = 1
            share = money / value * 0.98
            buyprice = value
        if status == 1 and value / buyprice > 1.5:               # Sell Signal
            status = 0
            money = share * value * 0.98
            cur = 0
```

[12]:
```
money
```

[12]: 52733.181739128115

[13]:
```
"""
Model 5:
    Different Buying Stages
"""
cash = 1000
money_spend = 0
asset = 0
percent = 10
for index in df_bitcoin.index:
    if index != 1825:
        cur = df_bitcoin.loc[index]
        nex = df_bitcoin.loc[index+1]
        cur_value = cur['Value']
        cur_ma5 = cur['MA5']
        cur_ma10 = cur['MA10']
        cur_ma30 = cur['MA30']
        nex_value = nex['Value']
        nex_ma5 = nex['MA5']
        nex_ma10 = nex['MA10']
        nex_ma30 = nex['MA30']

        if cur_ma5 > cur_ma10 and nex_ma5 < nex_ma10:
            percent = 20
        if cur_ma10 > cur_ma30 and nex_ma10 < nex_ma30:
            percent = 50

        if money_spend < cash * 0.8 and cur_ma5 > cur_ma10 and nex_ma5 > nex_ma10:       # signal for buy
            asset += (cash * 0.20) * 0.98 / nex_value
            money_spend += cash * 0.20
        if money_spend < cash * 0.5 and cur_ma5 > cur_ma10 and cur_ma10 > cur_ma30 and nex_ma10 > nex_ma30: ␣
↪    # signal for bigger buy
            asset += (cash * 0.50) * 0.98 / nex_value
            money_spend += cash * 0.50
        if (money_spend != 0) and (nex_value * asset / money_spend) > 1.50:
            cash = (cash - money_spend) + asset * nex_value * 0.98
            money_spend = 0
            asset = 0
total = cash - money_spend + asset * nex_value
```

[14]:
```
total
```

[14]: 38816.763280649044

[16]:
```
df_bitcoin = pd.read_csv("../dataset/BCHAIN-MKPRU_model6.csv")
df_gold = pd.read_csv("../dataset/LBMA-GOLD_model6.csv")
```

[17]:
```
df_gold.columns = ['Date', 'Value', 'Return', 'MA5', 'MA10']
df_gold = df_gold[29:]
df_bitcoin = df_bitcoin[29:]

for index in df_gold.index:
    row = df_gold.loc[index]
    df_gold.loc[index, "Date"] = datetime.strptime(row['Date'], "%m/%d/%y")
for index in df_bitcoin.index:
    row = df_bitcoin.loc[index]
    df_bitcoin.loc[index, "Date"] = datetime.strptime(row['Date'], "%m/%d/%y")
```

[19]:
```
for index in df_gold.index:
    if index != 29:
        row = df_gold.loc[index-1]
        rowcompare = df_gold.loc[index]
        x1 = row['Value']
        x2 = rowcompare['Value']
        y1 = row['MA5']
        y2 = rowcompare['MA5']

        if x1 < y1 and x2 > y2:
            df_gold.loc[index, "decide"] = 1
        elif x1 > y1 and x2 < y2:
            df_gold.loc[index, "decide"] = -1
        else:
            df_gold.loc[index, "decide"] = 0
```

[20]:
```
df_gold = df_gold[1:]
df = df_bitcoin.merge(df_gold, how='outer',on='Date')
golddate = df_gold['Date'].values
```

[21]:
```
"""
Final Model: Model 6
    Combining all the ideas above.
    The idea of Moving Average (MA), different buying stages, self-adapted selling thresholds,
    and diversification of total assets are applied.
"""
trans = pd.DataFrame(columns = ['Date', 'money_owned', 'money_spend', "bitcoin_value", "type"])
money = 1000
cash = 1000
status = 0
goldstatus = 0
thres = 1.5
sellprice = 1000
money_spend = 0
bitcoin = 0
share = 0
for index in df.index:
    if index <= 1766:
        row = df.loc[index]
        rowcompare = df.loc[index+1]
        value = rowcompare['Value_x']
        date = rowcompare['Date']
        x1 = row['Value_x']
        x2 = rowcompare['Value_x']
        y1 = row['MA5_x']
        y2 = rowcompare['MA5_x']
        a1 = row['MA10_x']
        a2 = rowcompare['MA10_x']
        b1 = row['MA30']
        b2 = rowcompare['MA30']
        if a1 > b1 and a2 < b2:
            thres = 1.5
        if a1 < b1 and a2 > b2:
            thres = 3

        # Buying and Selling Bitcoins
```

```
        if goldstatus == 0 or goldstatus == -1:
            if money_spend < money * 0.8 and x1 > y1 and x2 > y2:
                if goldstatus == -1:
                    while date not in golddate:
                        date -= timedelta(days=1)
                    money = share * df_gold[df_gold['Date'] == date].iloc[0]['Value'] * 0.99
                    goldstatus = 0
                bitcoin += (money * 0.20) / value * 0.98
                money_spend += money * 0.20
                dict = {"Date":date, "money_owned":money, "money_spend":money_spend, "bitcoin_value":value,␣
↪"type":"buy small"}
                trans = trans.append(dict, ignore_index=True)

            if money_spend < money * 0.5 and y1 > a1 and y2 > a2:
                if goldstatus == -1:
                    while date not in golddate:
                        date -= timedelta(days=1)
                    money = share * df_gold[df_gold['Date'] == date].iloc[0]['Value'] * 0.99
                    goldstatus = 0
                bitcoin += (money * 0.50) / value * 0.98
                money_spend += money * 0.50
                dict = {"Date":date, "money_owned":money, "money_spend":money_spend, "bitcoin_value":value,␣
↪"type":"buy large"}
                trans = trans.append(dict, ignore_index=True)

            if (money_spend != 0) and (value * bitcoin / money_spend) > thres:
                money = (money-money_spend) + bitcoin * value * 0.98
                money_spend = 0
                bitcoin = 0
                goldstatus = 1
                dict = {"Date":date, "money_owned":money, "money_spend":money_spend, "bitcoin_value":value,␣
↪"type":"sell"}
                trans = trans.append(dict, ignore_index=True)

        # Buying and Selling Golds
        if (goldstatus == 1) and (date in golddate):
            if rowcompare['decide'] == 1:
                share = money / rowcompare['Value_y'] * 0.99
                goldstatus = -1

        if (goldstatus == -1) and (date in golddate):
            if rowcompare['decide'] == -1:
                money = share * rowcompare['Value_y'] * 0.99
                goldstatus = 0
```

[22]:
```
total = money + bitcoin * value
total
```

[22]: 60063.71003857569