ISSN: 2688-9323

Quantitative Trading Decision Based on Dynamic Programming and Gradient Decision Tree Model

-- Taking Gold and Bitcoin Trading as Examples

Xingyuan Huang, Yujie Wang, Tianyi Wang

Minzu University of China, Beijing 100081, China

Abstract

" No fever like gold fever ". Gold has become the world's most popular currency because of its rarity and stability. In recent years, bitcoin has continued to rise in the global markets and has also become a premium trading asset. Starting in 2020, they have become an important safe-haven asset as the COVID-19 outbreak and global economic instability take hold. It is of great value to accurately predict their rise or fall and make the right decisions. After performing data analysis and modeling, we finalized the best strategies and highest returns for gold and bitcoin over 5 years. First, we pre-process the provided data, which includes normalization. Due to the limited data provided, we establish ARIMA and GBRT models to predict past flows to prevent insignificant forecasting effects. Then, we establish a Dynamic Programming model, taking each day as a phase, with boundary conditions [1000, 0, 0], we find that the objective function and the state transfer equation could not be directly represented by simple functions, so we build an innovative decision model and combine the two models as DP-DM. Through exploration, we set the score of buying, by calculating and normalizing it to find the range threshold where 95% of them lie, we determine the following strategy: Buy when the gold score is greater than 0.58 and sell when it is less than 0.3, and buy when the bitcoin score is greater than 0.71 and sell when it is less than 0.56. Next, we use DP-DM to find the best daily strategy and the best final return. If we follow our daily strategy, we end up with $226,648 on 9/10/2021. and the best daily strategy is: 1. Don 't buy anything in the first month. 2. A month later, start to buy gold. Starting from 0.46 gold, 415.34 cash. 3. A year later, start to buy bitcoin and as the price of bitcoin rose further, start to buy more bitcoin and selling the gold we held. Finally, we prove the optimality of the model in various ways, our model is significantly stable and robust to transaction costs.

Keywords

DP-DM; GBRT; Dynamic Programming; The Score of Buying; ARIMA.

1. Introduction

1.1. Background

Develop a model that gives the best daily trading strategy based only on price data up to that day. How much is the initial $1000 investment worth on 9/10/2021 using your model and strategy?

1.2. Overview of Our Works

To determine how daily decisions are made to maximize returns, we build a series of models to solve this problem and derive results for each stage. First, we build ARIMA time series model and GBRT model, and apply them to daily price forecasting. Then, we build a dynamic programming model that treats each day as a stage, determining the objective function and the

ISSN: 2688-9323

state transfer equation, and deriving the daily decision from the results. Thirdly, we test whether these decisions are optimal and check the accuracy of the model. Next, we analyse the sensitivity of the model to transaction costs and discuss the advantages and disadvantages of these models. Finally, we write a memo to market traders presenting our decision scenario and making relevant recommendations.

the entire modelling process is as follows:

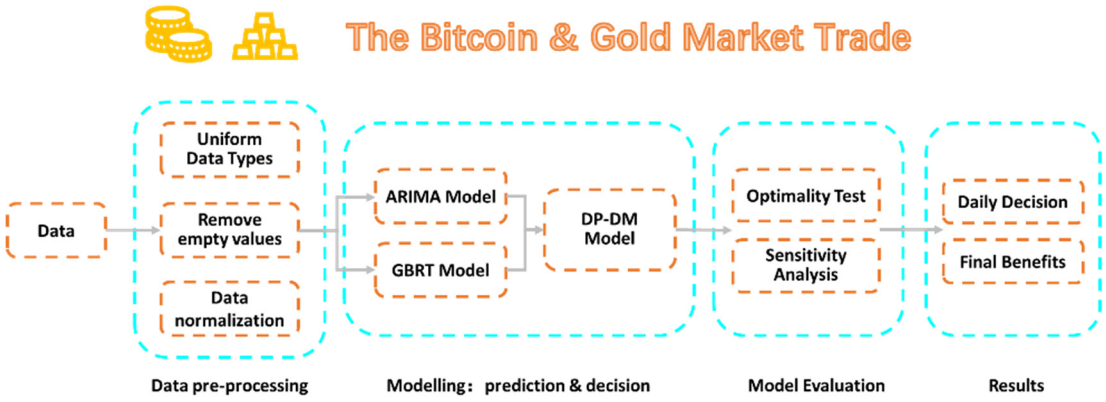

Figure 1. Workflow

2. Assumptions and Reasons

In order to make our model more in line with the facts, we make the following basic assumptions:

Assuming that the data after screening is correct and robust for further analysis.

Assuming that before you can buy gold or bitcoin again, you need to sell the one you have, which means that you can only have at most one issue of gold and one issue of bitcoin on hand at the end of each day, because we have only two assets in this problem.

Assuming that assets sold on the same day can only be withdrawn on the second day, assets that need to be purchased on the same day can only be purchased through cash holdings, which is in fact the case, and may even take 2 business days and more.

3. Problem 1: Modeling and Making Decisions

3.1. Data Pre-processing

Before analysing the price trends of gold and bitcoin, we need to pre-process the data. From the data provided, we can find the following problems:

Different units between data

We consider normalizing the data, which facilitates the direct application of these data to our model. Here we use the Z-Score method, the normalization formula as follows:

$$y_i = \frac{x_i - \mu}{\sigma}$$

In this equation, μ represents the mean value of data, σ represents the standard deviation.

3.1.1. Prediction Results of ARIMA

Now we can use ARIMA to predict the daily data, and after running the program, we can get the prediction results. We choose the results on four of the days as follows:

ISSN: 2688-9323

Table 1. Results of bitcoin (ARIMA)

Date	Real price	Prediction	Deviation
10/1/16	614.82	607.36	1.21%
10/2/16	612.98	605.81	1.17%
.....			
9/9/21	46078.38	46462.22	0.83%
9/10/21	46368.69	46185.65	0.39%

Table 2. Results of bitcoin (ARIMA)

Date	Real price	Prediction	Deviation
10/3/16	1313.3	1334.8	1.64%
10/4/16	1283.3	1308.78	1.99%
.....			
9/9/21	1788.3	1786	0.13%
9/10/21	1794.6	1788.3	0.35%

3.2. Gradient Boosting Regressor Tree (GBRT)

3.2.1. Establishment of GBRT

GBRT is a member of the integrated learning boosting family, but it is very different from traditional *Adaboost*. Freidman proposed a negative gradient of the loss function to fit an approximation of the current round of losses, and then fit a CART regression tree.[1]

The negative gradient of the loss function of the ith sample of the t-round is expressed as:

$$r_{ti} = -[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i))}]|_{f(x)=f_{t-1(x)}} \tag{1}$$

Utilization $(x_i, r_{ti}), i = 1,2, \cdots m$, we can fit a CART regression tree and get the t regression tree, which corresponds to the leaf node region $R_{tj}, j = 1,2, \ldots, J$. where J is the number of leaf nodes.

For the samples in each leaf node, we find the output value c_{tj} that minimizes the loss function, that is, the best fit leaf node:

$$c_{tj} = \underbrace{\arg min}_{c} \sum_{x_i \in R_{tj}} c_{tj} I, x \in R_{tj} \tag{2}$$

In this way we get the decision tree fitting function for this round as follows:

$$h_t(x) = \sum_{j=1}^{J} c_{tj} I, x \in R_{tj} \tag{3}$$

The expression of the strong learner that is finally obtained in this round is as follows:

$$f_t(x) = f_{t-1}(x) + h_t(x) \tag{4}$$

By fitting the negative gradient of the loss function, we find a general way to fit the loss error. In this way, whether it is a classification problem or a regression problem, we can solve our classification regression problem with GBRT by fitting the negative gradient of its loss function. The difference is only in the negative gradients caused by different loss functions.

3.2.2. Prediction Results of GBRT

Now we can use GBRT to predict the daily data, and after running the program, we can get the prediction results, and choose four of the days as follows:

ISSN: 2688-9323

Table 3. Results of bitcoin (GBRT)

Date	Real price	Prediction	Deviation
10/1/16	614.82	660.33	**6.22%**
10/2/16	612.98	660.33	**8.31%**
.....			
9/9/21	46078.38	46568.07	**1.06%**
9/10/21	46368.69	46568.07	**0.43%**

Table 4. Results of bitcoin (GBRT)

Date	Real price	Prediction	Deviation
10/3/16	1313.3	1312.8	**0.04%**
10/4/16	1283.3	1283.0	**0.02%**
.....			
9/9/21	1788.3	1795.0	**0.38%**
9/10/21	1794.6	1795.0	**0.02%**

3.3. Dynamic Programming-decision Model (DP-DM)

3.3.1. Establishment of DP

The basic idea of dynamic programming is to decompose a large-scale and difficult to solve original problem into multiple simple sub-problems, but these sub-problems are not independent of each other. Dynamic programming algorithms are optimal for implementing a problem in a certain situation, and the subproblems they decompose are optimal. On the other hand, the sub-problems it decomposes overlap each other, saving the solved problems in a table, which can reflect the efficiency of the algorithm as the scale of the problem increases.

The optimal strategy of a process has the property of constituting an optimal strategy for the process of taking the state formed by the first decision as the initial state, regardless of its initial state and the initial decision. That is to say, the essence of this principle is that the multi-stage decision-making process has the nature of making the next optimal decision only from the current state and the optimization requirements of the system, regardless of the past process. These are the principles of dynamic programming.

The steps to apply the dynamic programming algorithm are shown below:

Step 1. Since we knew the state of the first day, we decided to do dynamic programming from beginning to end. And we need to give the best daily trading strategy, we will take each day of the five years as a phase: $K = \{k_1, k_2, \cdots, k_{1826}\}$, $k_i, i = 1,2, \cdots, 1826$ means the i^{th} stage, which is the i^{th} day.

Step 2. For day i, what we have is the triple $[C, G, B]$, that is the amount of cash, gold, and bitcoin that we have at the beginning of day i. Thus, the triplet $[C, G, B]$ is our state variable s_i.

And the decision of each day is the amount of gold we need to buy or sell x_i and the amount of bitcoin y_i, so x_i and y_i are the decision variables.

Step 3. The optimization function is how we buy and sell gold and Bitcoin every day so that we have the greatest return on this day. So, we need to consider the practical question of how we should buy and sell. On the question of how to buy and sell, we developed a decision model, and the optimization function the state transfer equation are in 3.4.2.

Step 4. The boundary condition is the initial state S_0 at the beginning of the first day. According to the requirements in the title, we can know that the boundary condition is a triplet $[1000, 0, 0]$. According to above *functions* and *variables* and the *initial state*, so we can solve dynamic programming problems.

3.3.2. Establishment of Decision Model

(1) Symbol Explanation

(2) Rules of Buying and Selling

Considering whether to buy or sell, we create the score of buying. Using Bitcoin as an example, the score is calculated as:

$$bt_n = w_1 \times df_n^{bp} + w_2 \times f_n^{sb} + \frac{1}{f_n^{sr}} - \delta$$

ISSN: 2688-9323

Table 5. Notations

Notation	Meaning	Notation	Meaning
df_n^{sb}	Price of bitcoin	df_n^{sg}	Price of gold
α_b	Commission rates for bitcoin	α_g	Commission rates for gold
bt_n	Scores of buying bitcoin	$gold_n$	Scores of buying gold
df_n^{bt}	Current bitcoin profits	df_n^{gold}	Current gold profits
df_n^{bp}	The rise of bitcoin	df_n^{gp}	The rise of gold
df_n^{getb}	Share of bitcoin holdings	df_n^{getg}	Share of gold holdings
df_n^{getc}	Cash holdings	$df_n^{dealday}$	Whether or not the trading day
df_n^{all}	Total Assets		

Among them, $f_n^{sb}$is the score of bull market, $f_n^{sr}$is the score of risk, δ means residuals, w_1,w_2 means weight.

After repeated validation, we believe that w_1, w_2 are optimal to take 10 and 5 respectively. Since bulls are the stocks with big uptrends and generally choose to buy in bull markets, we consider using scores of bull market instead of bear market's. After normalizing the scores, the following is the procedure for processing the buy scores.

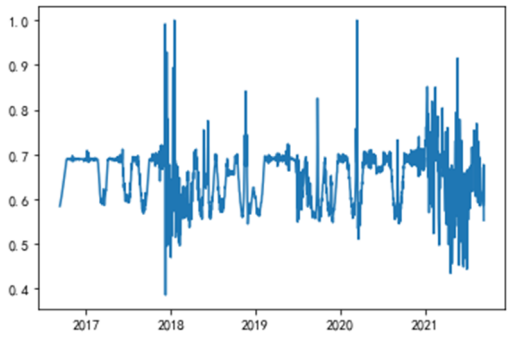

Figure 2. Bitcoin buying score

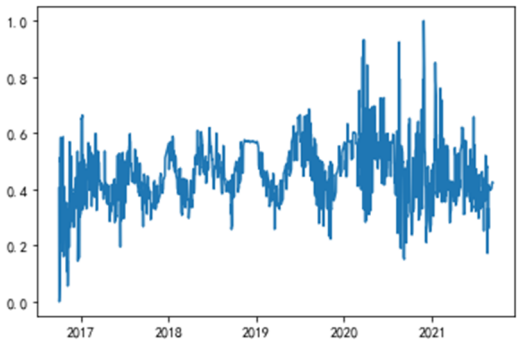

Figure 3. Gold buying score

We can conclude that 95% of the data for the bitcoin buy score falls in the range of 0.55 to 0.71. Similarly, the range for gold is 0.3 to 0.58, so we get the following strategy.

Buy when the gold score is greater than 0.58 and sell when it is less than 0.3, and buy when the bitcoin score is greater than 0.71 and sell when it is less than 0.56.

Therefore, when bitcoin and gold can be bought at the same time, if: $gold_n - 0.58 > (bt_n - 0.71) \times 2$, that means the score of gold is lower than bitcoin's, and the gold has a huge upside, so we decide to purchase gold. Conversely, purchase bitcoin.

Additionally, take gold as an example, the number of purchases and sales can be expressed as:

$$df_n^{getg} = df_{n-1}^{getg} + df_{n-1}^{getc} \cdot gold_n \cdot \frac{(1-\alpha_g)}{df_{n-1}^{sg}}$$

$$df_n^{getc} = df_{n-1}^{getc} - df_{n-1}^{getg} \cdot gold_n$$

3.3.3. Results of DP-DM

Calculated by the above algorithm, we can predict the price changes of gold and bitcoin.

We programmed the algorithm to get the results. The following graphs show the curves of bitcoin, gold, cash and total assets as a function of date:

ISSN: 2688-9323

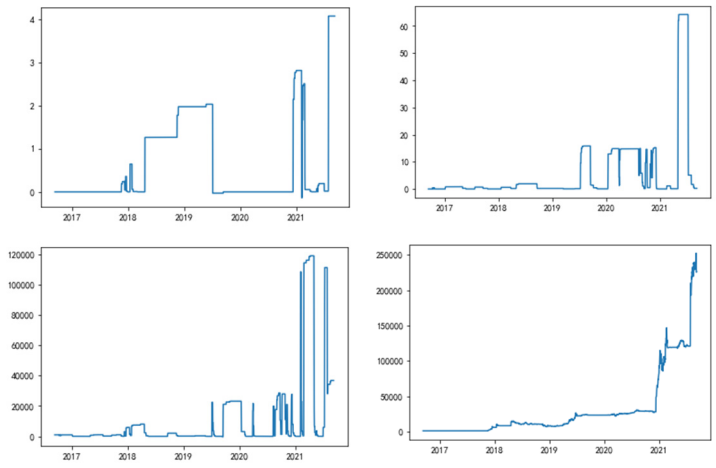

Figure 4. The curves of bitcoin, gold, cash and total assets

The following table shows the daily decisions and total revenue:

Table 6. Daily decisions & total assets

Date	Gold holdings	Bitcoin holdings	Cash holdings	Total assets
9/11/2016	0	0	1000	1000
9/12/2016	0	0	1000	1000
.....				
10/10/2016	0.459831239	0	415.3408369	994.5
10/11/2016	0.167975464	0	779.2572611	989.81
.....				
6/27/2019	0.131002188	2.036620729	0.001846733	26522
6/28/2019	0.131003047	2.036620729	0.000630047	22858
.....				
9/9/2021	0.217492962	4.085563555	36815.87771	225461
9/10/2021	0.217492962	4.085563555	36815.87771	226648

As we can see, after the last day, the total assets we have are 226648 dollars. Compared to the original $1,000, it is more than 220 times higher, which shows that our decision is a relatively good decision, and we will introduce our decision is the optimal decision in the next section.

In the first month, we did not buy and sell, because in the initial time we could not judge the law of market changes very well. After more than one month, we started buying gold, which was more expensive at the time. Then one year later, the price of Bitcoin started to rise, we started buying bitcoin and as the price of bitcoin rose further, we started buying more bitcoin and selling the gold we held, and finally reached an excellent result of $220,000 in total assets.

It can be found that our buying rules follow the principle of price fluctuations in the market and follow our own scoring system, which is reasonable and very efficient.

4. Problem 2: Proof of Optimality

We will prove that our strategy is optimal in the following ways:

- **We have high accuracy of forecasts.**

ISSN: 2688-9323

Our model has a small percentage of error in predicting the next day's gold and bitcoin prices. The average error for both ARIMA and GBRT for Bitcoin is within 3%, 2.81% and 2.91%, respectively. The average error for both forecasts of gold price is within 1%, 0.71% and 0.58%, respectively.

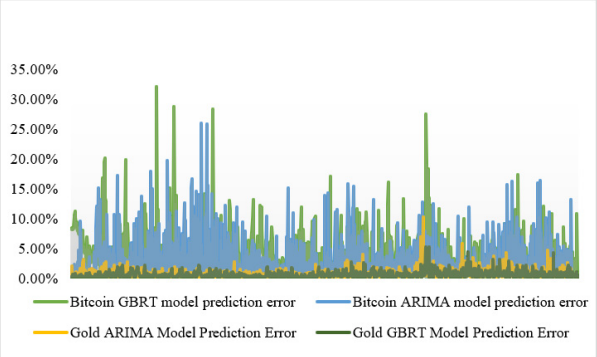

Figure 5. Prediction error percentage plot

Due to the high accuracy of our forecasts, we provide a good reference for buying and selling gold and bitcoin.

- **We accurately determine the risk of investment.**

As the price of Bitcoin continues to rise, the price of Bitcoin has risen as much as 70 times, so the risk of buying Bitcoin is increasing, as reflected in our charts. As a very stable commodity, the price of gold has risen steadily for five years, so the risk of buying gold has remained at a relatively stable level, which is also reflected in our charts. The purchase risk of gold and bitcoin is a key consideration in our decision making. It is our relatively accurate judgment of investment risk, in line with the objective laws of change, that ensures the accuracy of our decisions, and we have good reason to believe that our decisions are optimal.

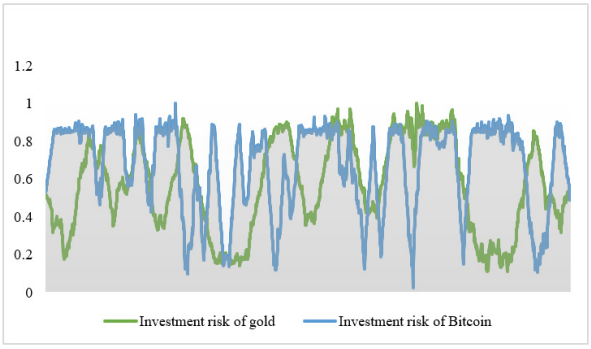

Figure 6. Investment risk of bitcoin & gold

- **We gave the correct gold and bitcoin purchase scores.**

Gold and Bitcoin purchase scores are an important part of our decision-making model in Dynamic Planning, which helps us determine the buy-sell strategy for each day. The right score helps us make the right judgment. The threshold we set for buying and selling is in line with objective laws and in line with the changing situation of the market, which helps us to better buy and sell to ensure that every day's trading is optimal.

ISSN: 2688-9323

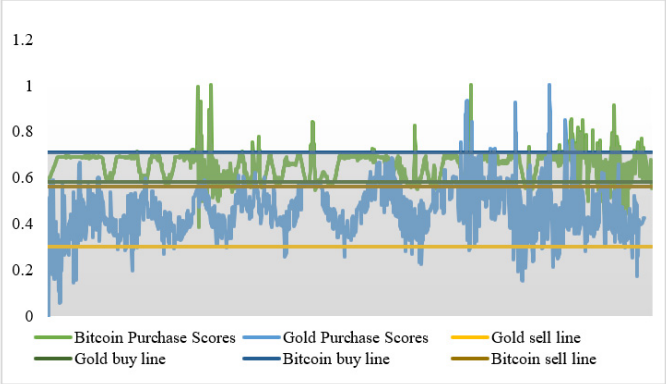

Figure 7. Purchase scores

- **We take into account bull and bear markets**

We make decisions according to the laws of the bull and bear market, and according to the market's reaction as one of the criteria for judging our decisions, it also makes our decisions more accurate and makes our final returns better.

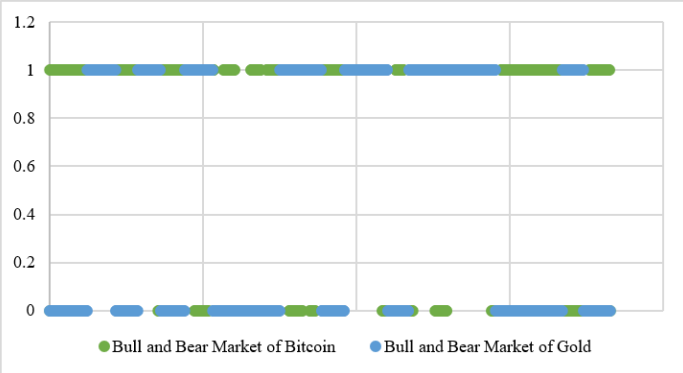

Figure 8. Distribution of bull and bear market

5. Conclusion

We combine DP with a decision model of our own creation, called DP-DM. The model innovatively uses the framework of dynamic programming to address the limitations of simple dynamic programming models where the state transition equations and optimization functions are functions of a single mathematical expression.

We get the best decision for each day, by using the DP-DM model and our prediction data. For example, the best decision for 5/1/2018 is [913.60, 1.2348, 1.2653]. Ultimately, our total assets with the initial $1,000 in cash became $226,648 on 9/10/2021, a 236-fold increase.

We show that our strategy is optimal and prove robustness of our model. By continuously changing the parameters, we find that none of the total returns are as high as the results obtained from this model, and the model has better stability and robustness to transaction costs.

References

[1] Cheng Ming. Research on the forecasting method of international gold price[D]. Shandong University, 2020. DOI: 10.27272/d.cnki. gshdu.2020.003329.

[2] Verano Dwi Asa, Husnawati, Ermatita. Implementation of Autoregressive Integrated Moving Average Model to Forecast Raw Material Stock in The Digital Printing Industry[J]. Journal of Information Technology and Computer Science,2020,5(1).

ISSN: 2688-9323

[3] Juan A. Ortega, Eugenio Losada, Roberto Besteiro, Tamara Arango, Maria J. Ginzo Villamayor, Ramon Velo, Maria D. Fernandez, Manuel R. Rodriguez. Validation of an AutoRegressive Integrated Moving Average model for the prediction of animal zone temperature in a weaned piglet building[J]. Biosystems Engineering,2018,174.

[4] Box G.E.P, Jenkins G.M. Time series Analysis: Forecasting and Control[M]. San Francisco: Holden Day,1976.

[5] Friedman J H. Greedy Function Approximation: A Gradient Boosting Machine[J]. The Annals of Statistics, 2001, 29(5):1189-1232.

[6] Jerome H. Friedman, Greedy Function Approximation: A Gradient Boosting Machine, The Annals of Statistics, Vol.29, No.5 (Oct., 2001), Pages 1189-1232.