

Quantitative Trading Strategy Based on Simplified DPG

Linpei Huang[†], Tengwei Cai[†], Shuai Feng^{*,†}, Zhecheng Xie[†]

Sichuan University, Chengdu, China

* Corresponding Author Email: 2019141520111@stu.scu.edu.cn

[†] These authors contributed equally.

Abstract. In recent years, investment has become more and more popular, and asset management has also received more and more attention. At the same time, with the development of computer science, more and more machine learning or deep learning algorithms can be used for investment management, such as price forecasting, portfolio and quantitative trading strategies. First of all, in the trading market, most of the ups and downs are cyclical, but they are easily affected by factors such as policies and investment fever, which brings great challenges to the establishment of a reasonable price forecasting model. This paper builds a sliding price forecasting model that aims to reduce various noises such as frequent fluctuations. In order to get the best forecast results, we searched the parameters of each sliding forecast, trying to get the forecast closest to the actual price. Second, developing trading strategies based on forecast results is clearly related to goal planning. Therefore, this paper first establishes the objective programming problem, designs the objective function and constraints, and then optimizes the problem. At the same time, the article also considers investor styles, such as the investment boom in gold and bitcoin, and a more cautious approach to buying and selling. Ultimately, this paper designs a flexible feedback model based on style parameters to adjust the trading strategy, by searching for the best parameters of the final model, our trading strategy model converts assets from \$1000 from September 11, 2016 to September 10, 2021 It increased to \$238,869, and the cumulative return reached 23,886.9%.

Keywords: Price prediction, Goal programming, Trading strategy, ARIMA.

1. Introduction

Bitcoin and gold are the two most popular investment products in recent years, which are inseparable from their characteristics. For bitcoin, bitcoin is a virtual currency product based on blockchain technology, which has the characteristics of fast transaction and security. The biggest property is that bitcoins are created as reward for a process known as mining. Due to the limited number of bitcoins, the number is about 21 million, that is, for every additional bitcoin in the market, the total amount will be reduced by one. At the same time, with the speculation of many investors, the price of bitcoin is getting higher and higher. If we keep positions from September 11, 2016, to September 10, 2021, the roughly estimated return on investment is about 7,000%. But the higher the return, the greater the risk. On the contrary, the gold has a stable value, and the price fluctuation is far less than that of bitcoin. When the market is depressed, investing in gold is a better measure to resist risks. Based on the above discussion, how to establish a trading strategy between bitcoin and gold to achieve maximum return is a very challenging and interesting problem. While realizing excess returns, it is also very important for the ability to resist risks.

2. Model Establishment

2.1. Data Preprocessing

This article needs to preprocess the available data, i.e. the price data for bitcoin and gold. By analyzing the characteristics of data, we deal with the missing value and smooth it to make it more suitable for model training data. For the processing of missing values, we found that bitcoin price data is normal, but gold price data is missing. This is because bitcoin can be traded at any time, but gold trading has time restrictions. For example, trading cannot be carried out on holidays. For missing

values, we use the average of gold price on the previous trading day and gold price on the following trading day. In order to make it more standardized, we regularize the date into the format of "Year-Month-Day". Then we use Savitzky-Golay smoothing algorithm [1] to remove noise. The biggest feature of this method is that it can ensure that the shape and width of the signal remain unchanged while denosing. On the contrary, wavelet transform will change the shape and width of the signal and destroy the data to a certain extent, which is also an important reason for choosing S-G smoothing.

(1) Missing value processing:

For gold price data, the price is missing on non-trading days, which will affect the subsequent price series prediction. We expect the value we filled in to minimize the impact on the results of prediction, so we use the mean value of the price on the previous tradable day and the next tradable day as the filling value. Moreover, we set the identification bit to indicate (T_g) whether gold can be traded every day.

(2) Date formatting:

The date series in the price data is a little confusing. In order to facilitate subsequent analysis, this paper uniformly sets the date format to "year-month-day".

(3) Denoising by S-G smoothing:

There are extremely frequent fluctuations in the original price data, which is not conducive to our series prediction. These frequent fluctuations are related to market trading sentiment or market policies, so it is very important to reduce the impact of these factors in order to make a reasonable price prediction. S-G smoothing is a very excellent algorithm, which can reasonably denoise and smooth the data. Unlike wavelet transform, it can well maintain the width and shape of the original signal, which is an important reason why we choose S-G algorithm.

Savitzky-Golay filter [2] is a lowpass data smoothing method based on local least-squares polynomial approximation. It can be viewed as a moving average filter with weight coefficients given as a polynomial of a certain degree. The mathematical description of the S-G filtering process is shown by the formula:

$$s_j^* = \frac{\sum_{i=-m}^{i=m} c_i s_{j+i}}{N}, \quad (1)$$

Where s is the original data, s^* is the smoothed data, c_i is the coefficient for the i -th filtering, N is the number of data points in the filter window and is equal to $2m+1$, where m is the half-width of the smoothing window. j represents the index of the ordinate data in the original data sequence.

The S-G filter is adopting a polynomial in a sliding window to fit the original signal piece-by-piece depending on the least-squares estimation algorithm. The polynomial can be modeled as:

$$f_k(i) = b_0 + b_1 i + b_2 i^2 + \dots + b_k i^k = \sum_{n=0}^k b_n i^n, \quad i \leq 2m. \quad (2)$$

The least-squares criterion expressed by Eq. 3 is applied to obtain the unknown filtering coefficients b_n .

$$\frac{\partial}{\partial b_n} \left[\sum_{i=-m}^m (f_k(i) - s_i)^2 \right] = 0 \quad (3)$$

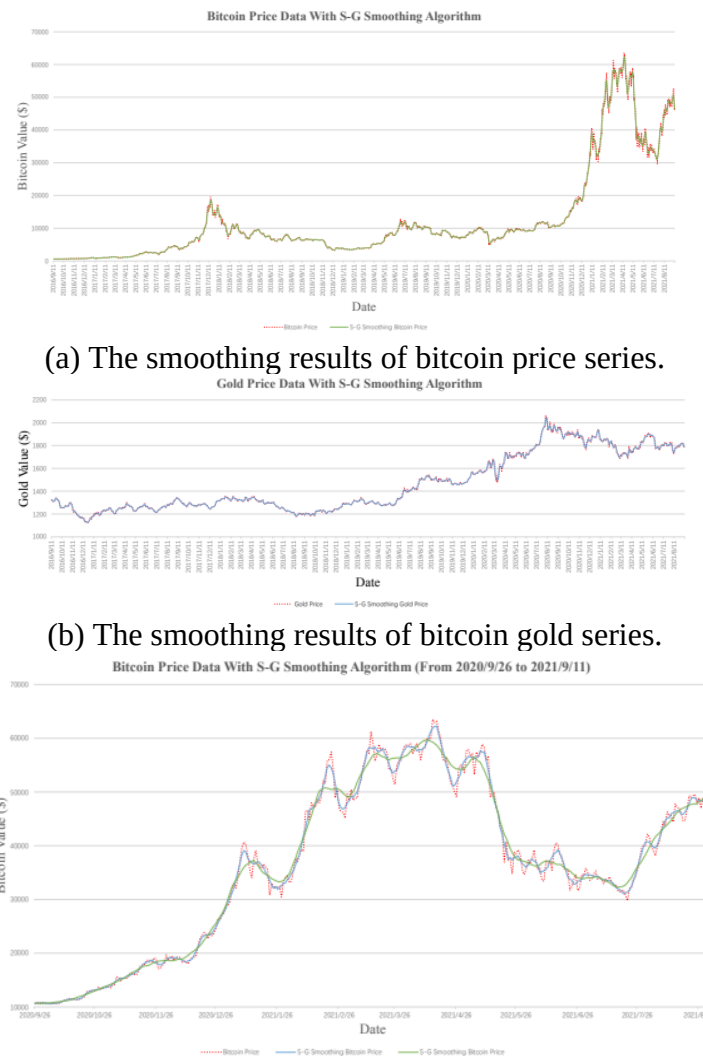
This leads to $k+1$ simultaneous equations for computing the unknown coefficients b_n . We evaluate Eq.2 at $i=0$ and can only obtain an expression for b_0 . Then, we compute the n -th differential of Eq.2 at $i=0$ to get b_n . The obtained coefficients $b_0, b_1, \dots, b_k$ are equivalent to the desired coefficients c_i , $i \in [-m, m]$. So, we deduce an expression from Eq.2 as

$$f_k^n(0) = \sum_{i=-m}^m c_i^n s_i \quad (4)$$

Where n denotes the derivative order, c_i^n and s_i are the convolution weight and the value of i -th point, respectively.

S-G algorithm requires two key parameters: the window size and the polynomial degree. If the window length is too long, it will produce some loss of valid signals, whereas if the window length is too short, it cannot denoise well; if the polynomial degree is too high, it may yield redundant data and produce new noise, and if the polynomial degree is too low, it may yield over-smoothing and signal distortion.

In this paper, window-size and polynomial-degree are set to 5 and 2. We compared the smoothing results of different parameters, as shown in Fig 1.



(c) The partial smoothing result of bitcoin price series. We can see that the selection of window-size is very important to the smoothing effect. When window-size is 15, it is too smooth and many important pairs of peaks are lost. When window-size is 5, it is more appropriate.

Figure 1. The smoothing results of price series

2.2. Prediction Model

The predicted price can only use the data before the day, so this article uses the sliding window prediction. After each training and prediction, there is a second iteration until all predictions are done.

For the selection of algorithm, we use the sliding window method to select different proportions of training data for experiments. We use the data smoothed by S-G algorithm to carry out the experiment and use the average goodness of fit (R_{ave}^2) of all sliding windows (the closer it is to 1, the better) to evaluate the performance of the sequence model. Because the sliding training is very time-consuming,

we use 100 epochs for the algorithm that needs iterative training, such as LSTM. The specific results are shown in Table 1 and Table 2.

Table 1. The comparison of algorithms with different proportions of initial training data (bitcoin)

Proportion Algorithm	20%	40%	60%	80%
AdaBoost [3]	0.490	0.703	0.836	0.920
Xgboost [4]	0.362	0.812	0.907	0.931
LightGBM [5]	0.561	0.899	0.926	0.967
LSTM [6]	0.612	0.951	0.991	0.998
ARIMA [7]	0.856	0.938	0.964	0.969

Table 2. The comparison of algorithms with different proportions of initial training data (gold)

Proportion Algorithm	20%	40%	60%	80%
AdaBoost [3]	0.508	0.862	0.956	0.991
Xgboost [4]	0.414	0.842	0.977	0.989
LightGBM [5]	0.674	0.872	0.965	0.993
LSTM [6]	0.712	0.971	0.997	0.999
ARIMA [7]	0.881	0.954	0.975	0.988

From Tables 1, 2, we found that LSTM has a high R_{ave}^2 , which undoubtedly shows the excellence of LSTM. However, it has an obvious disadvantage that it needs sufficient data to support training. We can see that when the proportion of initial training data is 20%, it is far better than ARIMA [8]. However, with the increase in proportion, the performance of LSTM is getting better and better. Other algorithms have the same characteristics, such as AdaBoost, Xgboost, and LightGBM. On the contrary, when the proportion of initial training data is low, ARIMA shows better performance. With the increase of the proportion, the performance is getting better and better, and finally close to LSTM, which shows that ARIMA has strong stability. Based on the above analysis, this paper uses ARIMA to build a price prediction model.

2.3. ARIMA Model

ARIMA is a variant of ARMA, so ARMA is worth introducing. ARMA model is actually the combination of AR and MA, which is quite self-explanatory. ARMA takes into consideration both the past values and past error terms and describes a (weakly) stationary stochastic process in terms of two polynomials. Formally a time series is ARMA (p, q) if it is stationary and

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q} \quad (5)$$

Where $\phi_p \neq 0, \phi_q \neq 0, w_t \sim N(0, \sigma^2)$. The parameters p and q are the autoregressive and the moving average orders respectively, as mentioned before. In terms of the back-shift operator, the ARIMA model can be written as:

$$\phi(B)x_t = \theta(B)w_t \quad (6)$$

ARIMA is ARMA modeled on a differenced series, e.g., First-order differencing: $x'_t = x_t - x_{t-1}$. It takes the difference between two observations. ARIMA model is usually denoted by ARIMA(p, d, q). p is the order of the autoregressive part, d is the degree of differencing and q is the order of the moving average part. The formal definition of ARIMA [1] model is

$$x'_t = \phi_1 x'_{t-1} + \dots + \phi_p x'_{t-p} + w_t + \theta_1 w_{t-1} + \dots + \theta_q w_{t-q} \quad (7)$$

Where x' denoted the difference series. In this paper, d is confirmed as 1, p and q through analysis. During calculation, iterative search within the range of $[0, 10]$ is used to find the optimal parameters (p_{best}, q_{best}) .

2.4. Trading Strategy and Feedback Model

To confirm the trading strategy, this article treats it as a "goal planning problem". On this basis, a concrete equation is established. First, the daily return on investment (ROI) [9] of bitcoin and gold for the next 10 days can be reliably obtained by predicting the results of the model, suggesting that we can formulate a goal planning problem with these predicted ROIs to obtain expected transaction times and expected transaction amount (percentage). When the expected trading time is reached, make a trade (gold trading hours need to meet the assumptions), then continue with the target planning for the next day to get the next trading time and expectations.

Next, we turn the goal programming problem into a mathematical model that can be optimized. Given the asset weight of bitcoin W_b , the asset weight of gold W_g , the asset weight of cash W_c and the prediction return of bitcoin P_b , the prediction return of gold P_g , and the transaction (purchase or sale) cost $\alpha_{bitcoin}, \alpha_{gold}$. The goal equation on a certain day can be established as

$$R_{expected} = (W_b + \Delta_b) \times Pb[D_d] + (W_g + \Delta_g) \times Pg[D_g] + (W_c - \Delta_b - \Delta_g) - C \quad (8)$$

Where Δ_b and Δ_g are the expected transaction proportion, D_b and D_g are the expected transaction time, C is the transaction cost.

$$C = |\Delta_b| \times \alpha_{bitcoin} + |\Delta_g| \times \alpha_{gold} \quad (9)$$

Therefore, we have established our prediction return equation (Eq.8), and then establish the objective function and add some constraints to form a mathematical expression of the goal programming problem. The objective function can be described as:

$$\begin{aligned} & \max_{\{\Delta_b, \Delta_g, D_d, D_g\}} R_{expected} \\ & s.t. \begin{cases} W_b + W_g + W_c = 1 \\ 0 \leq W_b \leq 1, \\ 0 \leq W_g \leq 1, \\ 0 \leq W_c \leq 1, \\ 0 \leq D_d \leq 9, D_d \in \mathbb{Z}, \\ 0 \leq D_g \leq 9, D_g \in \mathbb{Z}, \end{cases} \end{aligned} \quad (10)$$

As for how to optimize Eq.10, we found that the difficulty of the problem is that D_d and D_g are actually an index about prediction return, so we limit the range through the design parameters $[U_b, U_g, B_b, B_g] \in [0, 10]$, and then use cyclic calculation to find the optimal optimization parameters. These parameters affect the transaction frequency (cost) to a certain extent, which are also important parameters for subsequent analysis of transaction cost. Therefore, the optimization problem of each cycle is rewritten as

$$\begin{aligned} & \max_{\{\Delta_b, \Delta_g, D_d, D_g\}} R_{expected} \\ & s.t. \begin{cases} W_b + W_g + W_c = 1 \\ 0 \leq W_b \leq 1, \\ 0 \leq W_g \leq 1, \\ 0 \leq W_c \leq 1, \\ D_d = D'_d, D'_d \in \mathbb{Z}, \\ D_g = D'_g, D'_g \in \mathbb{Z}, \end{cases} \end{aligned} \quad (11)$$

Where $D'_d \in [B_b, U_b]$ and $D'_d \in \emptyset$, $D'_g \in [B_g, U_g]$ and $D'_g \in \emptyset$. Obviously, Eq.11 becomes a linear programming problem, i.e., it can be optimized. After a cycle optimization is completed, we cycle optimization in turn until we find the best condition of the parameters $\{\Delta_b, \Delta_g, D'_d, D'_g\}$. After returning, D'_d , D'_g needs to add the date of the current day to be the real date of the expected transaction.

Next, we need to define the trading formula of bitcoin and gold. If it comes to the day when bitcoin expects to trade, the specific trading equation is as follows

$$\begin{cases} W_b = W_b + W_c(1 - \alpha_{\text{bitcoin}}), W_c = 0, & \text{if } \Delta_b > 0, \Delta_b W_b - W_c > 0 \text{ ①} \\ W_b = W_b + \Delta_b W_b(1 - \alpha_{\text{bitcoin}}), W_c = W_c - \Delta_b W_b, & \text{if } \Delta_b > 0, \Delta_b W_b - W_c \leq 0 \text{ ②} \\ W_c = W_c + W_b(1 - \alpha_{\text{bitcoin}}), W_b = 0, & \text{if } \Delta_b < 0, \Delta_b W_b + W_b < 0 \text{ ③} \\ W_c = W_c - \Delta_b W_b(1 - \alpha_{\text{bitcoin}}), W_b = W_b + \Delta_b W_b, & \text{if } \Delta_b < 0, \Delta_b W_b + W_b \geq 0 \text{ ④} \end{cases} \quad (12)$$

Where cases ①, ②, ③, ④ show cash all in bitcoin, position increase, position clearing and position reduction respectively.

Similarly, if it comes to the day when bitcoin expects to trade and gold can be traded on the day, the specific trading equation is as follows

$$\begin{cases} W_g = W_g + W_c(1 - \alpha_{\text{gold}}), W_c = 0, & \text{if } \Delta_g > 0, \Delta_g W_g - W_c > 0 \text{ ⑤} \\ W_g = W_g + \Delta_g W_g(1 - \alpha_{\text{gold}}), W_c = W_c - \Delta_g W_g, & \text{if } \Delta_g > 0, \Delta_g W_g - W_c \leq 0 \text{ ⑥} \\ W_c = W_c + W_g(1 - \alpha_{\text{gold}}), W_g = 0, & \text{if } \Delta_g < 0, \Delta_g W_g + W_g < 0 \text{ ⑦} \\ W_c = W_c - \Delta_g W_g(1 - \alpha_{\text{gold}}), W_g = W_g + \Delta_g W_g, & \text{if } \Delta_g < 0, \Delta_g W_g + W_g \geq 0 \text{ ⑧} \end{cases} \quad (13)$$

Where cases ⑤, ⑥, ⑦, ⑧ show cash all in gold, position increase, position clearing and position reduction respectively.

Through the above discussion, we have established a strategic trading model based on linear programming, which can change the trading frequency by controlling four parameters U_b, U_g, B_b, B_g . In order to further simulate the real trading situation, the style of investors we consider will affect the trading, so we design four relevant parameters B_I, B_D, G_I, G_D to describe them, and then skillfully feedback the above trading strategy model. These parameters enlarge or reduce the amount of the desired transaction to simulate different styles.

$$\begin{cases} \Delta_b = B_I \times \Delta_b, & \text{if } \Delta_b > 0 \text{ ⑨} \\ \Delta_b = B_D \times \Delta_b, & \text{if } \Delta_b \leq 0 \text{ ⑩} \\ \Delta_g = G_I \times \Delta_g, & \text{if } \Delta_g > 0 \text{ ⑪} \\ \Delta_g = G_D \times \Delta_g, & \text{if } \Delta_g \leq 0 \text{ ⑫} \end{cases} \quad (14)$$

If $B_I > 1$, it will enlarge the expected transaction amount for increasing bitcoin position. If $B_D > 1$, it will enlarge the expected transaction amount for decreasing bitcoin position. If $G_I > 1$, it will enlarge the expected transaction amount for increasing gold position. If $G_D > 1$, it will enlarge the expected transaction amount for decreasing gold position. In other words, if traders like to trade bitcoin, we can make the values of B_I and B_D much greater than G_I and G_D . If traders are sensitive to risks and are easy to carry out clearance transactions, we can make B_D and G_D much larger than B_I and G_I . On the contrary, traders are more aggressive and like full positions. We can make B_I and G_I greater than B_D and G_D . In general, the four parameters can flexibly affect transactions.

3. Model analysis

3.1. Analysis of the Trading Strategy Model

(1) Maximum return analysis

Through the analysis of the maximum return, we can get three effective facts as follows: The established trading strategy model can increase the value of assets and achieve excess earnings; By

analyzing the asset change curve, we can judge whether the established trading strategy model can make a reasonable decision-making transaction; By calculating some financial indicators of asset changes, the effectiveness of the model can be clearly explained.

Compare the curve in Fig.2 with the full line of gold in Fig.5, we found that our strategy has good risk control ability. The price of bitcoin began to plummet around December 11, 2017, and the trading strategy model chose to clear the position of bitcoin. A large number of purchases were made around February 11, 2018, and the price of bitcoin has begun to rise slowly. This shows that during this period, the trading strategy model is reasonable for the decision-making and trading of bitcoin, and avoids the risk to a great extent. When the rise begins, the model starts to buy a lot to realize the income. Coincidentally, around July 2019, the model decided to clear the warehouse again. At the same time, the price of bitcoin began to decline slowly. Until around March 2020, the strategy began to buy a large number of bitcoin. As we all know, March 2020 is the beginning of bitcoin investment frenzy.



Figure 2. Gold daily prices and gold daily prices

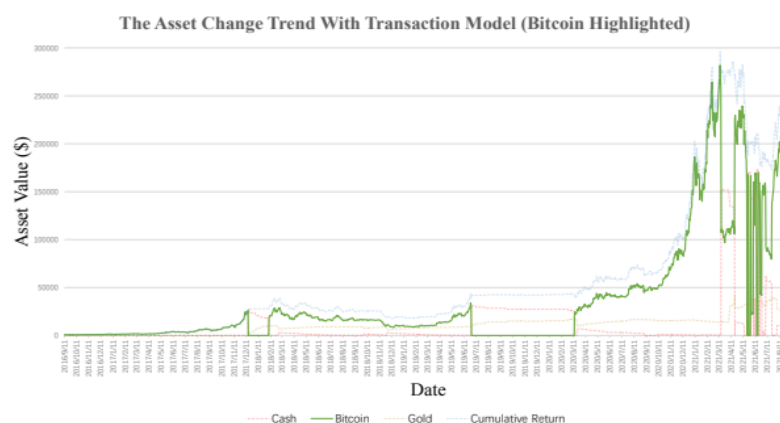


Figure 3. The asset change trend with transaction model (the bitcoin curve highlighted)

Around March 2021, the price of bitcoin began to fluctuate. At this time, the trading decision of model changed frequently and there were mistakes in decision-making, but the impact on the whole was not great. Around March 10, the model decided to reduce the position of bitcoin and increase the position of gold at the same time, which perfectly controlled the risk and retained the income.

From May to July 2021, the bitcoin market is unpredictable, and the decision-making of the model basically fails, which is also an important stage of final revenue reduction.

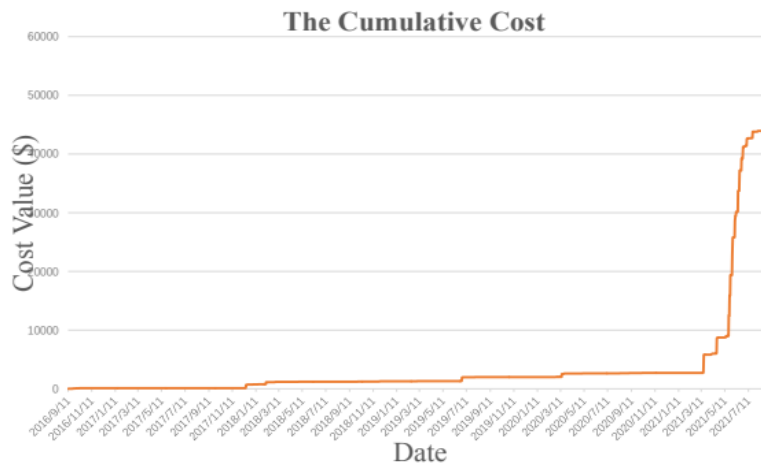


Figure 4. The change of cumulative cost in trading strategy with maximum return

Compare the curve in Fig.2 with the dotted line of gold in Fig.2, since the parameters of the model style are biased towards bitcoin, the fluctuation of the value of gold is relatively gentle. But in most cases, the trading model decides that buying gold is the time to reduce position of bitcoin. And its curve is related to $G_t(1)$, because the desire $B_t(11)$ to buy bitcoin is 11 times that of gold.

Overall, the optimal strategy model obtained by parameters search completes the excess return. The cumulative maximum return reached 296.1805 times, including 381 gold transactions and 58 bitcoin transactions. The total transaction cost was US \$52,835.92 (See Fig.4 for details), and the final total assets reached US \$238,869.01 (initial \$1,000). If we trade according to the same initial portfolio weight [0.9, 0.1, 0], we keep the positions and the final assets is US \$65,922.25. In other words, our excess return on September 10, 2021, was \$172,946.76.

3.2. Free parameters analysis

(1) Style parameters analysis

The above experimental result is the optimal trading strategy obtained by artificially controlling the style and parameters search, which can not reflect the impact of various trading styles on the return on investment to a certain extent. We set different style parameters $\{B_t, B_D, G_t, G_D\}$ for abundant experiments to verify the effectiveness of the trading strategy. The final results are shown in Table 3.

Table 3. The comparison of algorithms with different proportions of initial training data (bitcoin)

Style parameters $\{B_t, B_D, G_t, G_D\}$	Style description	Final Asset (\$)	Transactions (bitcoin + gold)	Transaction cost (\$)
[5, 10, 10, 15]	Prefer gold	48,048.03	(20+539)	7,873,90
[10, 15, 5, 10]	Prefer bitcoin	33,677.56	(57+509)	4,611,43
[5, 10, 10, 15]	Risk sensitive	34,786.28	(58+549)	3,509,16
[5, 10, 10, 15]	Risk indifferent	20,437.21	(95+573)	1,790.94
[11, 30, 25, 1]	Parameters search	238,869.01	(58+381)	52,835.92

Through the experiment of artificially setting style parameters (See Table 3 for details), we found that investors who like to invest in gold have obtained relatively stable return. The style is more radical, and investors who do not pay much attention to risk have a lower final return. By comparing the case of optimal parameters, the number of transactions obviously increases, but the transaction cost is lower than the optimal case. There is no obvious relationship between the number of transactions and the transaction cost with different sets of parameters, which shows that the transaction strategy model has certain control over each transaction.

(2) Initial portfolio weights analysis

In order to explore the sensitivity of initial portfolio weights, we conducted an experiment with different initial portfolio weights. The experiment adopted the optimal style parameters $\{B_I, B_D, G_I, G_D\}$ with [11, 30, 25, 1] and the boundary parameters are [5, 5, 8, 8]. The specific results are shown in Table 4.

Table 4. The relationships between (transactions, cost, final asset) and different initial portfolio weights

Initial portfolio weights $[W_b, W_g, W_c]$	Final Asset (\$)	Transactions (bitcoin + gold)	Cumulative transaction cost (\$) 2016/9/11-2021/9/10
[0.9, 0.1, 0]	238,869.01	(58+381)	52,835.92
[0.6, 0.4, 0]	117,380.69	(58+549)	26,057.96
[0.5, 0.5, 0]	234,296.08	(58+321)	51,791.85
[0.3, 0.7, 0]	162,292.67	(58+549)	35,984.81
[0.1, 0.9, 0]	62,816.59	(58+549)	7,548.58
[0.1, 0.1, 0.9]	252,164.98	(58+379)	55,740.51
[0, 0, 1]	122,031.02	(58+547)	26,585.12

It is found that the initial portfolio weights had little effect on the trading frequency of bitcoin, but will affect the trading frequency of gold. An interesting phenomenon is that the lower the gold trading frequency, the higher the total transaction cost. Our guess is that bitcoin has a large position, and there will be more clearance transactions, resulting in an increase in the overall transaction cost. It is not difficult to find a rule that transaction costs are positively correlated with final assets.

(3) Boundary parameters analysis

In order to explore the sensitivity of boundary parameters, we conducted an experiment with different boundary parameters $([B_b, B_g, U_b, U_g])$. $[a, a, b, b]$ means that a gold or bitcoin transaction may occur within $a \sim b$ days after the goal programming of trading strategy model. Similarly, the experiment adopted the optimal style parameter $[B_I, B_D, G_I, G_D]$ with [11, 30, 25, 1] and the initial portfolio weights are [0.9, 0.1, 0].

The specific results are shown in Table 5. We find that the transaction frequency of bitcoin will slowly decrease with the increase of the boundary parameters. At the same time, the transaction of bitcoin has a great impact on the cost [10]. Similarly, with the same observation as the previous experiment, the greater return, the higher cumulative transaction cost.

Table 5. The relationships between (transactions, cost, final asset) and different boundary parameters

Boundary parameters $[B_b, B_g, U_b, U_g]$	Final Asset (\$)	Transactions (bitcoin + gold)	Cumulative transaction cost (\$) 2016/9/11-2021/9/10
[1, 1, 1, 1]	9.64	(1815+1227)	1,566.77
[1, 1, 2, 2]	1,408.03	(297+616)	4,450.33
[1, 1, 3, 3]	3,438.49	(162+609)	3,535.96
[2, 2, 4, 4]	1,949.9	(102+583)	1,543.56
[3, 3, 5, 5]	12,894.23	(76+568)	4,379.22
[4, 4, 6, 6]	234,296.08	(58+321)	51,791.85
[5, 5, 7, 7]	97,639.23	(48+285)	15,053.79
[6, 6, 8, 8]	8,347.63	(41+451)	603.94
[7, 7, 9, 9]	8,639.58	(38+439)	601.1
[8, 8, 10, 10]	7,470.02	(34+437)	592.18

In general, through the results of three experiments about free parameters, we summarize the following conclusions. The trading strategy model shows that trader who don't care much about risk often get lower return, which is in line with the real trading situation to a certain extent; Bitcoin accounts for a large transaction cost and has a great impact on the final return; The initial portfolio

weights have little effect on the transaction frequency of bitcoin; High frequency trading does not help return. On the contrary, the situation of obtaining high return is that the trading frequency is low.

3.3. Transaction cost sensitivity analysis

Determine how sensitive the strategy is to transaction costs. We experimented with different pairs of transaction costs. By fixing the transaction cost of gold or bitcoin, the other gradually increased from 0%. See Table 6 for detail.

Table 6. The different settings for transaction costs

α_{bitcoin}	α_{gold}	Step length
[0, 4%]	1%	0.2%
2%	[0, 2%]	0.2%

The specific experimental results are shown in Fig.5. With the increase of bitcoin transaction cost setting, the cumulative return obviously decreases slowly. However, with the increase of the transaction cost setting of gold, the cumulative return has not changed too dramatically, indicating that our trading strategy model is more sensitive to the transaction cost of bitcoin. The value of gold is relatively stable and plays a good role in preserving the value of assets. Even if the transaction cost increases, many people will be willing to invest in gold.

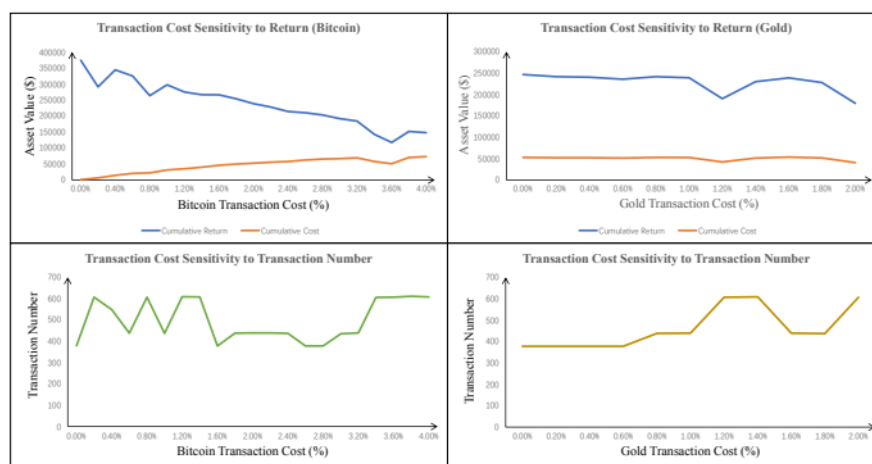


Figure 5. The results of transaction cost sensitivity analysis

At the same time, we find that the cost setting of bitcoin is positively correlated with the cumulative cost of the transaction strategy model, which means that the transaction costs generated by our model are mainly concentrated in bitcoin transactions. There is no obvious relationship between transaction frequency and cost setting, which shows that our transaction strategy model is stable to a certain extent and can automatically adjust and maximize benefits.

We found an interesting phenomenon. When bitcoin transaction has no cost, our strategy model achieves nearly 400 times of investment return. If this happens in the real trading market, it is reasonable. Because bitcoin has no transaction cost, traders may be more crazier about it.

4. Conclusion

This paper uses a very effective smoothing denoising algorithm (S-G smoothing), which can reduce the invisible interference in the data, so as to establish a good price prediction model and trading strategy model. The established price prediction model can automatically search for the best parameters, without much manual intervention. The trading strategy model established in this paper simulates real trading well, and obtains the optimal operation expectation through objective planning. This paper considers the real situation, that is, the trading style of investors will affect the trading. Therefore, this paper establishes a feedback model based on style parameters to help us make better trading strategy decisions, and proves that the feedback model is effective.

References

- [1] SAVITZKY, A., AND GOLAY, M. J. E. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry* 36 (1964), 1627–1639.
- [2] Zhu Hongyun, Wang Changlong, Wang Jianbin, Ma Xiaolin. Signal noise reduction method based on singular value decomposition and Savitzky-Golay filter [J]. *Computer Applications*, 2015, 35(10): 3004-3007+3012.
- [3] FREUND, Y., SCHAPIRE, R. E., ET AL. Experiments with a new boosting algorithm. In *icml* (1996), vol. 96, Citeseer, pp. 148–156.
- [4] GRAVES, A. Supervised sequence labelling with recurrent neural networks. In *Studies in Computational Intelligence* (2008).
- [5] KE, G., MENG, Q., FINLEY, T., WANG, T., CHEN, W., MA, W., YE, Q., AND LIU, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).
- [6] CHEN, T., AND GUESTRIN, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (2016), pp. 785–794.
- [7] ASTERIOU, D., AND HALL, S. G. Arima models and the box–jenkins methodology. *Applied Econometrics* 2, 2 (2011), 265–286.
- [8] Jin Yongchao, ZHUANG Xiaodie, Wang Renfang, Cao Qian. Thermal Coal Futures Price Prediction Based on ARIMA BP Neural Network [J]. *Journal of North China University of Science and Technology (Natural Science Edition)*, 202, 44(02): 92-98.
- [9] Zhang Xinguo, BAI Peiming. Thinking on constructing ROI model of marketing performance evaluation [J]. *Journal of hubei university of economics (humanities and social sciences edition)*, 2014, 11(07): 56-58+66.
- [10] Deng Hequan, Dai Liangtie, DONG Limin, Li Ying. Application of ROI Model in training effect Evaluation [J]. *China Business and Trade*, 2013(11): 49-50.