

Portfolio return prediction model based on gold and Bitcoin

Chengge Wen^{1,*}, Siyan Lu², Jiaxuan Jiang³

¹Business school, Central South University, Changsha, Hunan, 410000

²School of Geosciences and Info-Physics, Central South University, Changsha, Hunan, 410000

³College of Foreign Languages, Central South University, Changsha, Hunan, 410000

*Corresponding author: 1203521473@qq.com

Abstract. Maximizing returns has always been people's investment goal. Gold and bitcoin are popular with investors because of their hedges and volatility. However, markets are risky and can be influenced by different economic, political and environmental factors. As a result, bitcoin and gold prices fluctuate wildly, leading to uncertain investment and uncertain returns. In order to maximize the profit, this paper completes the data processing and model construction to make decisions. Based on the Markov decision process of avoiding risk avoidance, reducing transaction cost and maintaining liquidity, and assuming that the stock market is not affected by enhanced trading agent, deep reinforcement learning (DRL) is used to simulate stock trading. The application of the model is helpful to forecast the return of investment portfolio and brings strong application value to the relevant practitioners.

KeyWords: Transactional side; deep reinforcement learning; bitcoin.

1. Introduction

Market traders often buy and sell volatile assets with the goal of maximising their total returns. There is usually a commission on every sale. Two of those assets are gold and bitcoin. There is volatile property in the market, and its value constantly fluctuates over time. Traders can maximize total returns by constantly buying and selling volatile property to continuously increase the total value of the holding property. But different investment projects require a different commission for each sale. Investing is not blind, the key issue of trading decision is to execute the right decision [1] at the right time. When holding a variety of property such as gold, Bitcoin needs more strategy. Investors decide whether to buy the property or sell or continue to hold the property in the portfolio. We can provide us with the current transaction optimal strategy in the daily price changes of investment property known to date, finally achieving the purpose of maximizing the corresponding returns[2].

To explore the best investment trading strategy, our team has studied it with gold and Bitcoin. Bitcoin can be traded every day, gold can only be traded on trading days, and they have different trading commissions. Our team studied investment trading strategies by performing different portfolios for gold and bitcoin during the five-year trading period from November 9, 2016 to October 9, 2021. Develop a trading model based only on the daily price data as of the day, giving the best trading strategy of the day, including how to buy, sell and continue to hold property in the portfolio[3].

2. Model Preparation

2.1 Data selection

The data provided consist of the closing price of a troy ounce gold and the price of a bitcoin of each day during the five-year period commencing 9/11/2016. The data is pre-processed in the following fashion[4]:

The gold is not traded when the market is closed and as a result many dates are absent from the data. Additionally, some dates included in the data set do not have the corresponding closing price. All these dates are labelled as "not trading gold"[5].

2.2 Technical index addition

It is assumed that the trader is able to trade both gold and bitcoin in fractional shares. To simplify our model, we divide all prices of each equity by an arbitrary constant C_{equity} , where $C_{equity} p_{date, equity}$ for all dates, and the resultant values $p_{date, equity} / C_{equity}$ are used as the minimum units of trade for the equity[6].

For the model to mimic the decisions of human traders and fully exploit the patterns of the data, several common technical indicators are added to each equity of each date, which are:

(1) The n -day simple moving average, i.e. the average closing price for previous n trading days, where we take n as 5, 10, 30 to capture the trend of the prices of different length of periods;

(2) The difference of the price of the current day and the previous day;

(3) The prices of previous n days, where we take n as 1, 2, 3, 4, as the most recent prices have the largest influence on future prices based on the common random walk hypothesis[];

(4) The moving average convergence divergence (MACD), calculated by subtracting the 26-period exponential moving average (EMA) from the 12-period EMA; the signal line, which is a nine-day EMA of the MACD;

(5) The volatility, calculated by the standard deviation of previous 20 trading days, which measures the intensity of recent fluctuation;

(6) The bias, which is the percentage of the deviation of current day price from the 5-day simple moving average.

Because the calculation of some indicators requires data of previous dates, the data set is dropped until 21/10/2016 so that all dates have all indicators stated[7].

2.3 Reinforcement learning constitutes the environment

Similar to the tutorial FinRL: Multiple Stock Trading, we model the portfolio management process as a Markov Decision Process (MDP). We then formulate our trading goal as a maximization problem[8]. The algorithm is trained using Deep Reinforcement Learning (DRL) algorithms and the components of the reinforcement learning environment are:

Action: The action space describes the allowed actions an agent can take at a state. In our task, the action

$w(t) \in RN$ corresponds to the portfolio weight vector decided at the beginning of time slot, and should satisfy the constraints: firstly, the each element is between 0 and 1, secondly the summation of all elements is 1.

Reward function: The reward function $(s(t), w(t), s(t+1))$ is the incentive for an agent to learn a profitable policy. We use the logarithmic rate of portfolio return: $\ln(w(t)^T y(t))$ as the reward, where $y(t) \in RN$ is the price relative vector.

SRL Algorithms: We use two popular deep reinforcement learning algorithms: Advantage Actor Critic (A2C) and Proximal Policy Optimization (PPO).

Environment: Dow Jones 30 constituent stocks during 01/01/2009 to 09/01/2021

Action: portfolio weight of each stock is within [0,1]. We use softmax function to normalize the actions to sum to 1.

State: Covariance Matrix, MACD, RSI, CCI, ADX, state space shape is (34, 30). 34 is the number of rows, 30 is the number of columns.

Reward function: $r(s, a, s') = p_t, p_t$ is the cumulative portfolio value. Environment: portfolio allocation for Dow 30 constituents.

3. The Model of Deep Reinforcement Learning

The model is based on Deep Reinforcement Learning (DRL), which combines the framework of reinforcement learning and deep learning techniques.

The Markov Decision Process (MDP) is used to describe the trading process. We have the following

definitions:

A portfolio $P = (C, n_G, n_B)$ is the allocation of equities on a given date, where $C \in [0, +\infty)$ is the amount of cash in the portfolio, and $n_G, n_B \in \mathbb{N}$ respectively represent the number of shares of gold and bitcoin in the portfolio;

An observation \mathcal{O} is the share price p_G, p_B , whether gold is traded, and other technical indicators of the equities on a given date;

A given date is thus represented as a state $\mathcal{S} = (P, \mathcal{O})$;

Given a state \mathcal{S} , the value of an equity E is $v_E = n_E p_E$, and the value V of a state \mathcal{S} is the sum of cash and the market values of all equities in the portfolio under current market prices, $V = C + v_B + v_G$;

An action $A = (a_C, a_G, a_B)$ where $0 \leq a_C, a_G, a_B \leq 1$, $a_C + a_G + a_B = 1$ its elements respectively representing the intended ratio of value allocated to cash, gold and bitcoin, is taken by an agent given a state \mathcal{S} under a certain policy π ,

An equity allocation function f takes an action A and produces a new portfolio P' under certain restrictions;

A policy π is the agent's behavior function describing the trading strategy at state \mathcal{S} , standing for the probability distribution of actions at state \mathcal{S} ;

The reward R of an action A under a state \mathcal{S} , which results in the new state \mathcal{S}' , is the difference of the values of the two states $R = V(\mathcal{S}') - V(\mathcal{S})$;

Q-value $Q_\pi(\mathcal{S}, A)$ is the expected value of reward of taking action A at state \mathcal{S} under policy π .

(More restraints of the definitions)

The environment is modelled to simulate that of a trader in the gold and bitcoin market. To train the model each environment randomly picks a N -day period and slice the data only from these dates. The portfolio of the trader is initialized as $(1000, 0, 0)$. The environment then informs the agent of the first date's state. Then on each day, it sequentially takes the following steps:

The environment receives the action A given by the agent and updates the traders portfolio as $P' = f(A)$ according to the equity allocation function.

The environment shifts to the next date. It calculates the reward of given the new date's market and informs the agent of the reward as well as the new date's state.

A naive way of defining f is to find the closest number of shares of such a value ratio. We define such an f as f_0

The agent needs to develop a set of rules that decide the best award based on the given state and reward. This is taken care of with the use of a DdpAgent by TensorFlow. The agent is trained with $\gamma = 0.05$, batch size 60 and iteration number 400. (Details about random policy, replay buffer, etc.) (some graphs here)

After training the model is run on the entire data set to obtain the overall profit rate of the strategy. (We need some indicators of the model's performance.)

When the transaction fee is zero the trained model with f_0 performs quite well with profit rate ... (This is actually question 3) However when the transaction fee gets higher the model tends to degenerate... (some terrible graphs here)

This is due to the high frequency of transaction taken by the model. We mitigate this by redefining f In order to reduce unnecessary transaction fee, no transaction will be conducted if the change rates of n_G, n_B are both less than $\beta\%$.

In this section we will use the Markov Decision Process (MDP) to model the stock trading process and the objective function will be returned when it achieves its maximum.

We aim to describe the stochastic process of dynamic stock market, a MDP was deployed as following:

State: $s = [s, a, s']$: a vector that includes stock prices $p \in \mathbb{R}^D_+$, the stock shares $h \in \mathbb{Z}^D_+$, and the remaining balance $b \in \mathbb{R}$, where D denotes the number of stocks and Z denotes non-negative integers.

Action: a vector of actions over D stocks. The actions could be taken upon each of share includes selling, buying, or holding, representing for decreasing, increasing, and no change of the stock shares h , respectively.

Reward: the incentive mechanism for an agent to learn a better action. At state a , the firsthand reward of taking action a , and then arriving at the new state s'

Q-value: $Q_{\pi(s,a)}$: under policy π , the expected value of reward of taking action a at state s . The state transition of stock trading is shown. In the portfolio, there are three possible actions could be taken on stock $d(d = 1, 2, 3, \dots, D)$.

Selling $k[d] \in [1, h[d]]$ shares results in $h_{t+1}[d] = h_t[d] - k[d]$, where $k[d] \in Z^+$ and $d=1, \dots, D$.

Similarly, holding results in $h_{t+1}[d] = h_t[d]$.

Finally, buying shares results in $h_{t+1}[d] = h_t[d] + k[d]$. Action is taken at time t and then the stock price changes at time $t+1$. Consequently, the portfolio may be updated, for example, from portfolio value 0 to portfolio value 1. Where the portfolio value is $pTh + b$.

Now, consider the following constraints and assumptions, which are mainly used for practical reasons: risk aversion, transaction costs, liquidity.

Liquidity: we assume that stock market will not be affected by our reinforcement trading agent. These orders can be rapidly executed at the closing price.

Balance b being non-negative: the balance should not be less than 0, i.e. negative. At time t , the stocks are divided into sets for selling S , buying B , and holding H , where $S \cup B \cup H = 1 \dots D$. Meanwhile, there is no overlappings. Let $pB_t = [p_{it}: i \in B]$ and $kB_t = [k_{it}: i \in B]$ be the vectors of price and number of buying shares for the stocks in the buying set.

We can similarly define pS_t and kS_t for the selling stocks, and pH_t and kH_t for the holding stocks.

There is a relationship stating the constraints: $b_{t+1} = b_t + (pS_t T kS_t - (pB_t) T kB_t) \geq 0$

Transaction cost: transaction costs are included within each of the trade. We assume our commission to be 0.1% for gold and 0.2% for bitcoin of each trade as: $c_t = pTkt * 0.1\%$, $c_1 = pTkt * 0.2\%$

Risk aversion for market crash: there is always a possibility that some events might cause the market crash. To control this, financial turbulence index turbulence is used to measure the extreme value movements of the asset: $Turbulecet = (y_t - \mu) \sum -(y_t - \mu)' \in R$

4. Comprehensive analysis

We conducted a comprehensive analysis to obtain the optimization mode under different circumstances. We calculated the maximum return value when the transaction cost is the gold transaction cost and the bitcoin transaction cost combination as 0.005, 0.01, 0.02, 0.02, 0.04, 0.05 and 0.1, respectively [9].

Gold transaction cost 0.005, Bitcoin transaction cost 0.01 is shown as figure 1.

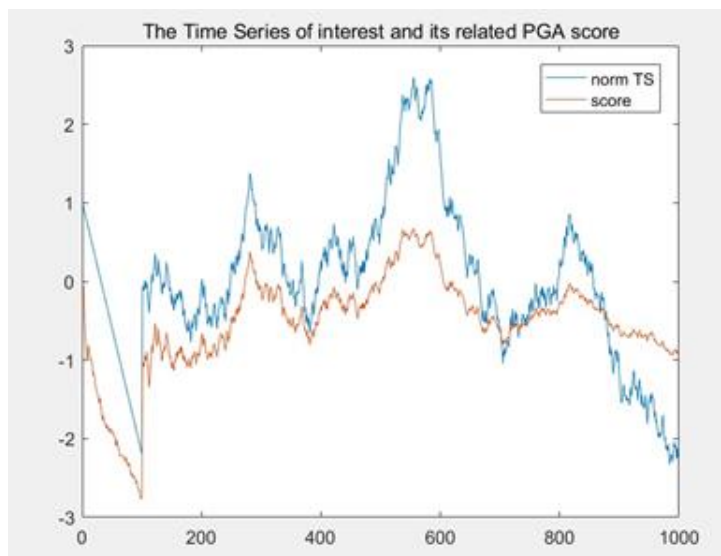


Figure 1. Gold transaction cost 0.005, Bitcoin transaction cost 0.01

Gold transaction cost 0.02, Bitcoin transaction cost 0.04 is shown as figure2.

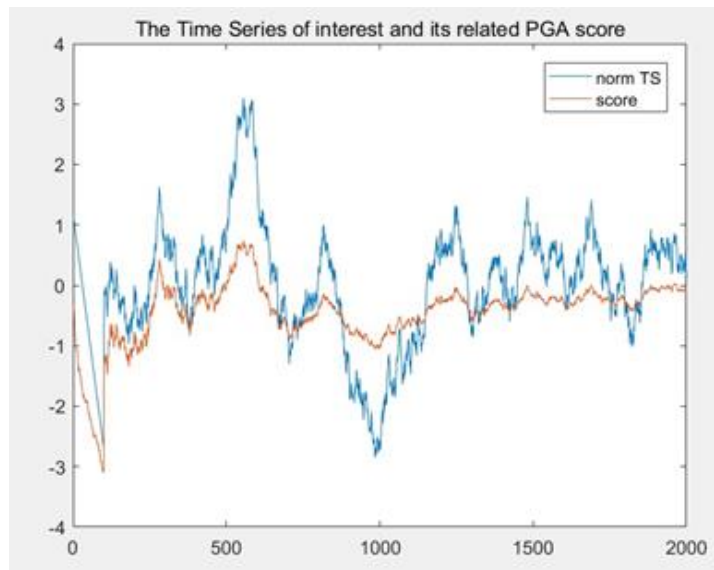


Figure 2. Gold transaction cost 0.02, Bitcoin transaction cost 0.04

Gold transaction cost 0.05, Bitcoin transaction cost 0.01 is shown as figure2.

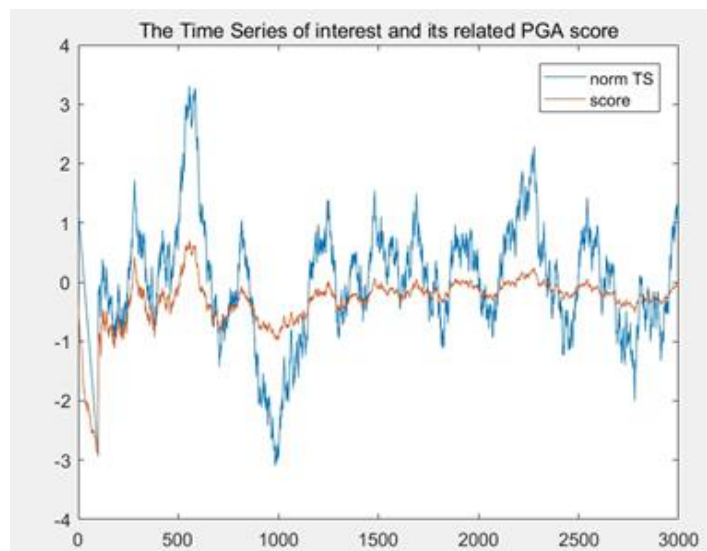


Figure 3. Gold transaction cost 0.05, Bitcoin transaction cost 0.01

Using such a model, we will get the maximum investment utility and calculate the change situation and the return maximum. As can be seen from the figure, the different transaction costs will change the location of their maximum value, based on the fact that the different transaction costs can be changed in the time series by affecting the investment expectations[10]. Different transaction costs do not have much impact on the maximum, presumably because the maximum has been greatly affected by the daily price in the past, and long transactions have stabilized it.

5. Conclusion

The model can give an optimization and improvement scheme for the relevant trading mode in a relatively short time, so that its efficiency, cost, and quality are all at a good level, and its effect is good, which can be dealt with for such problems. The model has a good application effect for such optimization problems, and can easily find its optimal trading mode, which is of great significance to its future research and development, and needs to be further improved through multiple modes. At the

same time, it can effectively regulate the possible risk value. After optimizing according to the goal of risk reduction and utility improvement, we can get its optimal value. When optimization, the model needs comprehensive analysis of multiple data to meet the specific requirements. Various conditions and values should be kept within a reasonable range. This is more difficult, and it needs to further optimize the model and control it. At the same time, for reinforcement learning and other methods, the relevant parameters need to consider, the adjustment of hyperparameters has high requirements, and the optimization algorithm should be used to obtain a more reasonable learning rate, iteration number and other parameters. The gold and bitcoin price prediction models based on DRL are scientific and reasonable, and can not only grasp the general characteristics of price data, but also grasp the time series characteristics of price data.

References

- [1] iMaps Capital Markets expands product portfolio with actively managed certificate for AI trading strategy[J]. M2 Presswire,2022.
- [2] Zhiduo Wang, Zixing Liu, Shuai Yin. Analysis of Trading Strategy based on ARIMA and Dynamic Programming Model[J]. International Core Journal of Engineering,2022,8(5).
- [3] Sohail Muhammad Khalid,Raheman Abdul,Iqbal Javid,Sindhu Muzammal Ilyas,Staar Abdul,Mushafiq Muhammad,Afzal Humaira. Are Pair Trading Strategies Profitable During COVID-19 Period?[J]. Journal of Information & Knowledge Management,2022,21(Supp01).
- [4] James Nick. Evolutionary correlation, regime switching, spectral dynamics and optimal trading strategies for cryptocurrencies and equities[J]. Physica D: Nonlinear Phenomena,2022,434.
- [5] Miseta Tamás,Fodor Attila,Vathy-Fogarassy Ágnes. Energy trading strategy for storage-based renewable power plants[J]. Energy,2022,250.
- [6] Xinchun Ye, Renjie Li, Hailong Yang. Best Daily Trading Strategy for Gold and Bitcoin[J]. Scientific Journal of Economics and Management Research,2022,4(4).
- [7] Song Dalei,Gan Wenhao,Yao Peng,Zang Wenchuan,Zhang Zhixuan,Qu Xiuqing. Guidance and control of autonomous surface underwater vehicles for target tracking in ocean environment by deep reinforcement learning[J]. Ocean Engineering,2022,250.
- [8] Wu Ling,Zhang Ji. Economic Operation and Management of Microgrid System Using Deep Reinforcement Learning[J]. Computers and Electrical Engineering,2022,100.
- [9] Kim Minwoo,Kim Jongyun,Jung Minjae,Oh Hyondong. Towards monocular vision-based autonomous flight through deep reinforcement learning[J]. Expert Systems With Applications,2022,198.
- [10] Xu Guoqiang,Jiang Weilai,Wang Zhaolei,Wang Yaonan. Autonomous Obstacle Avoidance and Target Tracking of UAV Based on Deep Reinforcement Learning[J]. Journal of Intelligent & Robotic Systems,2022,104(4).