The Research on Optimal Investment Strategy Based on Apriori BP Neural Network Model

Tingyue Wang[1], Jiangrong Qiao[2, *]

[1]Central University of Finance and Economics, Beijing, 102200

[2]School of Software, Taiyuan University of Technology, Taiyuan, Shanxi, 030024

*Corresponding author: qiaojiangrong0311@163.com

Keywords: Apriori algorithm; BP neural network; Quantitative investment; Trading strategy

Abstract: Quantitative investment is an emerging concept of systematic investment approach using computer programs combined with financial theories to perform investment analysis and trading. This paper focuses on designing a quantitative trading strategy to optimize the investment strategy and then designing the BP neural network model based on the Apriori algorithm with a risk function to facilitate the computation of different scenarios for three different types of trader investments, building a dynamic programming model later. Moreover, the robustness and sensitivity analysis of the model is tested, and the model performs stably. Besides, Using transaction cost as a perturbation term, it is concluded that transaction cost grows slightly inversely to the final value of the portfolio and the change in the number of transactions. Finally, The model is less sensitive, has good market adaptability, and has real significance.

1. Introduction

Quantitative investing is a new analytical approach that combines modern mathematical theory with financial data. This dynamic online decision-making problem requires computers to extract current market critical information and make optimal trading decisions. At the same time, forming a portfolio of several assets is a common way of investing to maximize returns.

This paper used the data of gold and Bitcoin to design a quantitative trading model to optimize the investment strategy and perform a sensitivity analysis to evaluate the pros and cons of the model.

2. Model Building and Results

2.1 The Price Forecasting Model

Due to the instability of financial data, the BP neural network has an extremely strong nonlinear mapping ability, which can effectively circumvent the shortcomings of other prediction models in treating financial data, so the prediction model is built based on the BP neural network. Moreover, the construction process of the model is as follows.

2.1.1 Model Building

First, construct the training set. A portion of the price samples of gold is randomly selected as the input layer values, and every five adjacent values are one basic unit, and the first three values are used to predict the last two values, and so on. Then, the input matrix $3 \times \alpha$ corresponding to the input layer is constructed, resulting in an output matrix $\alpha \times 1$. A randomly selected portion of the price samples of bitcoin is used as the values of the input layer. The same five adjacent values are used as a basic data unit, and the first three values are used to predict the last two, constructing an input matrix $3 \times \beta$, resulting in an output matrix $\beta \times 1$. Break up the numbers represented by the value of 1266 gold in 1266 days and randomly select 844 numbers from them. Disrupt the numbers represented by the value of 1827 bitcoins in 1827 days, and randomly select 1218 numbers from them.

Add a hidden layer between the input and output layers, the number of hidden layers needs to be slightly larger than the number of input layers, assume the number of hidden layers is 5, construct the relationship between the value of the hidden layer and the value of the input layer:

$$g_{\mathrm{ij}} = \frac{1}{1+e^{-k(w_{i1}I_{i1}+w_{i2}I_{i2}+w_{i3}I_{i3})}} \tag{1}$$

$$b_{\mathrm{ij}} = \frac{1}{1+e^{-\rho(\mathrm{v}_{i1}\mathrm{U}_{i1}+\mathrm{v}_{i2}U_{i2}+\mathrm{v}_{i3}U_{i3})}} \tag{2}$$

Second, the sigmoid function in BP neural network is used to predict the value of gold and the value of bitcoin.

$$G_{\mathrm{ij}} = \frac{1}{1+e^{-k(w_{i1}g_{j1}+w_{i2}g_{j2}+w_{i3}g_{j3}+w_{i4}g_{j4}+w_{i5}g_{j5})}} \tag{3}$$

$$B_{\mathrm{ij}} = \frac{1}{1+e^{-\rho(\mathrm{v}_{j1}\mathrm{b}_{j1}+\mathrm{v}_{j2}b_{j2}+\mathrm{v}_{j3}b_{j3}+\mathrm{v}_{j4}b_{j4}+\mathrm{v}_{j5}b_{j5})}} \tag{4}$$

By randomly adjusting the values of, and k, ρ, we ensure that the predicted values roughly match the original value-over-time graph.

Third, the remaining 422 values of gold and the remaining 609 values of bitcoin are used as the test set. After calculating the predicted values in the training set according to the sigmoid function in the prediction model, a value trend curve with only the training set is fitted, and a value trend curve with both the test set and the training set is fitted to test whether the constructed sigmoid function is appropriate.

Fourth, optimization is performed. Re-predict the estimates multiple times by randomly selecting 844 and 1218 values from 1266 values represented by gold and 1827 values represented by bitcoin, respectively, and using them as the validation set. Having the validation set effectively reduces the large errors caused by having only one training set and test set.

Finally, to correct for errors, the error functions on gold and bitcoin are constructed separately.

$$E_{G_{ij}} = \frac{1}{2}\sum_{i=1}^{n}\left(I_{ij}-G_{ij}\right)^2 \tag{5}$$

$$E_{B_{ij}} = \frac{1}{2}\sum_{i=1}^{m}\left(U_{ij}-B_{ij}\right)^2 \tag{6}$$

Through the relationship between error and weights, the error is used to mobilize the magnitude of the weights and fit the optimal prediction image.

That is,

$$\frac{\partial E}{\partial W_{KL}} = \frac{\partial E}{\partial O_K}\cdot\frac{\partial O_K}{\partial W_{KL}} = -2\left(t_K-O_K\right)^2\cdot\frac{\partial sigmoid(\sum W_{KL}\cdot O_K)}{\partial W_{KL}} \tag{7}$$

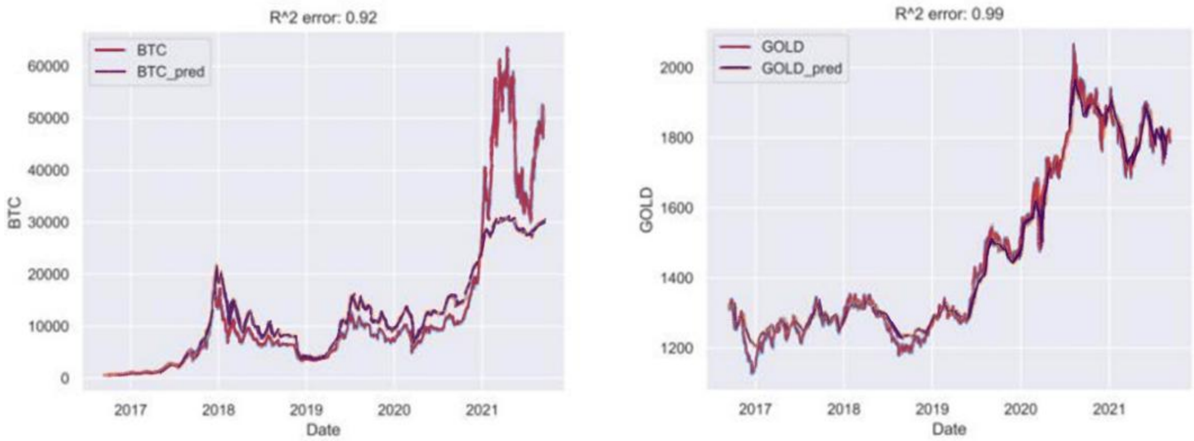
And then,

$$\frac{\partial sigmoid\left(x\right)}{\partial x} = \mathrm{s}igmoid\left(\mathrm{x}\right)\cdot\left(1-\mathrm{sigmoid}\left(\mathrm{x}\right)\right) \tag{8}$$

Finally,

$$\frac{\partial E_L}{\partial W_{KL}} = -(\mathrm{e}_K)\cdot sigmoid(\sum W_{KL}\cdot O_K)(1-sigmoid(\sum W_{KL}\cdot O_K)\cdot O_K \tag{9}$$

2.1.2 Results



(a) Bitcoin prediction error (b) Gold prediction error

Figure 1. Prediction Error

The prediction error for bitcoin is about 0.98 and for gold is 0.93. The fact is that the error is closer to 1 means that the prediction is much better. It means that our model has a good prediction.

2.2 The Dynamic Programming Model

Apriori algorithm finds frequent sets by the support and generates corresponding association rules. Therefore, the Apriori algorithm is coupled with the BP neural network model to reflect the investment types of the three types of traders to reflect the investment types of the three types of traders.

2.2.1 Apriori algorithm

The time span during which traders sell, buy, and hold amounts for gold and bitcoin makes them well suited to use price-moving average-based trading rules.

Define the moving average of gold prices of total length m as:

$$\overline{\mathrm{p}_{t,m}} = \frac{1}{\mathrm{m}} \sum_{i=0}^{m-1} p_{t-i} \tag{10}$$

Define the moving average of bitcoin prices of total length n as:

$$\overline{p_{t,n}} = \frac{1}{n} \sum_{i=0}^{n-1} p_{t-i} \tag{11}$$

Define the relative change in the moving average price of gold of total length m for the moving average price of total length n as:

$$x_{1,t}^{(m,n)} = \ln\left(\frac{\overline{p_{t,m}}}{p_{t,n}}\right) \tag{12}$$

$x_{1,t}^{(m,n)} > 0$ indicates an upward price trend, $x_{1,t}^{(m,n)} < 0$ indicates a downward price trend.

Seven fuzzy sets are defined for the percentage magnitude of $x_{1,t}^{(m,n)}$ change: "Positive Small (PS)," "Positive Medium (PM)," "Positive Large (PL)," "Negative Small (MS)," "Negative Medium (MM)," "Negative Large (ML)," and "Zero (Z)."

Let signal be the strength of a signal indicating the strength of buying bitcoin or gold and selling bitcoin or gold, which can be positive or negative. Define 7 fuzzy sets of signal: "buy small (BS)", "buy medium (BM)", "buy large (BB)", "sell small (SS)", "Sell Medium (SM)", "Sell Large (SB)", and "Hold (K)". For example, the affiliation function for Sell Big (BB):

$$U_{BB} = \begin{cases} (signal - 0.2)/0.2, & if\ signal \in [0.2,0.4] \\ (signal - 0.2)/0.2, & if\ signal \in [0.4,0.7] \\ 0, & otherwise \end{cases} \tag{13}$$

0.2,0.4,0.7 represents 20%, 40%, and 70% of a trader's buying or selling power to buy or sell gold and bitcoin. It represents the trader's three possible risk preferences: averse, neutral, and favorite. That is, the seven fuzzy sets represent the trader's willingness to use approximately 20%, 40%, and 70% of buying power, willingness to sell, and reservations, respectively.

The rule of the Apriori algorithm is to use fuzzy set $x_{1,t}^{(m,n)}$ and fuzzy set signal to find all frequent itemsets of $x_{1,t}^{(m,n)}$ and signal and to mine association rules. The procedure is as follows:

1. Input: two sets of all price datasets for gold and bitcoin:

$$p_g = \{p_0, p_1, ..., p_n\}$$
$$p_b = \{p_0, p_1, ..., p_n\} \tag{14}$$

Set the minimum support min-sup, and set the minimum confidence min-conf.

2. Output: associated rule base R

Algorithm implementation process: assume that the domain of $x_{1,t}^{(m,n)}$ has "Positive small (PS)," "Positive medium (PM)," "Positive large (PL)," "Negative small (MS)," "negative medium (MM)," "negative large (ML)" and "zero (Z)" 7 fuzzy sets, the thesis domain of signal is divided into "buy small (BS)," "buy the medium (BM)," "buy large (BB)," "sell small (SS), "Sell Medium (SM)," "Sell Large (SB)" and "Hold (K)" 7 fuzzy sets and these 14 fuzzy sets are labeled 1 -14, the daily trading operation can choose one from serial number 1-7 and one from serial number 8-14 respectively, which exactly represent the three preferences of trader's aversion, neutral and favorite.

2.2.2 Apriori Algorithm Coupling with BP Neural Network

Coupling the Apriori algorithm with a BP neural network, the implementation steps are as follows:

At time t, $X_{1,t}^{(m,n)}$ (m=1266, n=1827) (m=1266, n=1827) is substituted into each of the corresponding 7 fuzzy sets of the affiliation function and obtain the input signal matrix. The output signal matrix is obtained by putting the buying power or willingness to sell of traders at the moment t+1 corresponding to the affiliation functions of seven fuzzy sets ("BS," "BM," "BL," "SS," "SL," "SM," "K"), respectively.

The BP neural network is contacted for continuous training, validation, and testing. Make the input signal matrix correspond to the input values, and the output signal matrix correspond to the target output values. Create input, hidden, and output layers, and go through training, validation, and multiple testing. After repeated numerous training, the network with high enough accuracy can respond accurately to the input layer input training set, and the relationship rules between fuzzy sets are tighter within.

Data defuzzification. When a new fuzzy signal is an input to the trained network as input layer data, the output layer outputs an output signal data. Then, the defuzzifier converts the output signal into a non-fuzzy variable as the output result. It effectively reflects the three trader trading strategies of risk-averse, risk-neutral, and risk lappetite.

2.2.3 Risk Model

As traders trade in various situations, they are affected by many aspects. Therefore, a model about the magnitude of risk is needed. Risk is related to the total value of available wealth, the standard deviation of trading returns, and the mean of trading returns.

So a more intuitive expression can be used to represent the risk a trader faces in buying or selling gold and bitcoin.

$$VaR=\sum_{i=1}^{n} y_i c_i (Z\sigma_i+\mu_i) \tag{15}$$

Where y_i and h_i denote the value of investments in gold and bitcoin as a proportion of the total value available, denotes the value of investments in gold and bitcoin per day, σ denotes the standard deviation of returns, μ denotes the mean of returns, Z denotes the sampling quantile of a normal distribution, and the sampling quantile Z can be further expressed as:

$$1-\alpha=\int_{-\infty}^{Z}\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}dx \tag{16}$$

a is the 95% confidence level.

2.2.4 Creation and Solution of Dynamic Programming Models

Finally, an optimal trading strategy is modeled based on the two most important factors of return and risk, where denotes the initial investment value of $1000, P and M denotes the daily price of gold and bitcoin, with S denoting the return and VaR denoting the risk.

Then the trader invests in gold and bitcoin, assuming that the trader invests in gold for f days and in bitcoin for h days, so the total number of days D = f + h, return:

$$s=\sum_{i=1}^{n}\frac{c_i y_i}{p_i}(p_{i+1}-p_i)+\sum_{j=1}^{n}\frac{c_j h_j}{m_j}(m_{j+1}-m_j)-\alpha\%\cdot c_i y_i-\beta\%\cdot c_j h_j \tag{17}$$

The final objective function has three cases corresponding to traders with three preferences, with correlation coefficients of 0.2, 0.4, and 0.7, respectively. i.e., the final objective function is:

$$\begin{cases}\max Z=\sum_{i=1}^{n}(0.2s-0.8VaR)\\ \max Z=\sum_{i=1}^{n}(0.4s-0.6VaR)\\ \max Z=\sum_{i=1}^{n}(0.7s-0.3VaR)\end{cases} \tag{18}$$

According to the above modeling process, the final value is: (Unit: USD)

Table 1. The Risk Values

Risk-averse	Risk neutral	Risk appetite
1642.54	6357.66	12454629.57

2.3 Model Evaluation

2.3.1 Sensitivity Analysis and Market Adaptability

In sensitivity analysis, commissions are specified as a perturbation term to analyze how the trading strategy reacts to different commissions at different times and to observe the impact of commission changes on the strategy and results.

After adjusting the transaction costs for gold and bitcoin, Table 2 shows the changes in investment strategies and returns corresponding to the various transaction cost adjustment scenarios. Considering the need to be as realistic as possible, we continue to use the risk-neutral scenario for the run validation.

Table 2. Analysis Results

a_{gold}	$a_{bitcoin}$	Returns	Number (gold)	Number (bitcoin)
1%	2%	6357.66	141	498
1%	1.5%	6752.12	138	512
1%	2.5%	5829.78	124	455
1.5%	2%	6268.43	119	479
0.5%	2%	6286.94	164	501
2%	4%	6211.39	99	405

Note: a_{gold}, $a_{bitcoin}$ is the transaction cost of gold and bitcoin, Returns is the final total value of the portfolio, number (gold), number (bitcoin) is the change in the number of transactions of gold and bitcoin, respectively.

The results show that an increase in transaction costs leads to a small decrease in final value and a slight decrease in the number of transactions, and a decrease in transaction costs leads to a small increase in final value and a slight increase in the number of transactions. The magnitude of the variation in the results shows that this strategy is not very sensitive to transaction costs.

2.3.2 Comparison with ARIMA model

The ARIMA(p,d,q) model can be used to predict time-series data.

$$\left(1-\sum_{i=1}^{p}\phi_i L^i\right)\cdot(1-L)^d X_t=\left(1+\sum_{i=1}^{q}\theta_i L_i\right)\varepsilon_t \tag{19}$$

The ARIMA model cannot capture patterns if the data is unstable, such as those associated with financial assets. After running, the final value of $3340.16 was obtained, but the maximum retracement reached as much as 35%, which is a strategy with great volatility and risk, and the results are not as good as the model using neural networks.

3. Summary

Thorough consideration and analysis of the realities of different types of traders, this paper design the neural network model based on the Apriori algorithm with a risk function, recognizing that the optimal strategy is one with a steadily increasing rate of return. Moreover, the model is designed to facilitate the computation of different scenarios for three different trader investments while building a dynamic programming model later. The final dynamic programming model is prepared for the solution of the final dynamic programming model.

Besides, after replacing the model with ARIMA, it is incomparable to the model used in our strategy. Both in terms of the data of the financial indicators and the characteristics of the model itself. Moreover, by replacing the respective transaction costs of gold and bitcoin in the model, we test that both the final value and the number of transactions in the strategy move inversely to the transaction costs, and the model is not very sensitive.

Finally, Due to the limitation of the topic, we cannot use more data for training. Neural networks do not perform as well as large data volumes for small data volumes, and more data sets can be introduced in the future to train the model.

References

[1] Yao, H. C., Lai, J. W., Xia, SH. M., Chen, Sh. M. (2021,10). Fuzzy trading decision based on Apriori algorithm and neural network,41(10),2868-2891

[2] Yang Zhou. (2019) Applied research on fund forecasting based on BP neural network,1674-0688

[3] Zhang, Ling. (2014) Optimal Asset Portfolio Selection Based on Hidden Markov Models, F830.59, F224.3

[4] Antonios Georgantas. (2018) Robust Optimization Approaches for Portfolio Selection: AComputationaland Comparative Analysis