A Study of Trade Strategies Based on the Markov Regime Switching Model

Zhijie Shen[1, &], Zicheng Wei[1, &] and Yang Zhang[1, *, &]

[1]DGUT-CNAM Institute, Dongguan University of Technology, Dongguan, China

*Corresponding author: 15876990520@163.com

[&]These authors contributed equally to this work

Abstract. Market traders often buy and sell stocks to maximize their total return. For each purchase and sale, there is often a return commission. Having trading technology plays an important role in quantitative trading. In this paper, we first use the XGBoost model to learn the historical price fluctuation data of gold and bitcoin, and the prediction accuracy R2 is between 0.998 and 0.999, which is a good fit. Then LT algorithm and PS algorithm are used to identify the rules, the Markov regime switching model is used to determine the bear and bull market, and finally, the DQN model is used to plan the trading strategy and add a machine learning algorithm to it to optimize the strategy to get the initial $1000 in the future investment strategy.

Keywords: XGBoost; Markov Regime Switching Model; DQN; Trade Strategy.

1. Introduction

Trading strategies play a very important role in trading financial assets, and how to automate the selection of trading strategies in complex and dynamic financial markets is an important research direction in modern finance. To maximize their returns, market traders usually change their investment objects frequently. Considering the international status of gold, it used to be the favorite of market traders. However, with the advent of Bitcoin, investors seem to have found another reliable investment object. However, considering that both purchases and sales require a certain commission, the change of investment object should not be too frequent. Therefore, how maintaining a balance between buying and selling has become a big problem for market traders.

To alleviate this situation, this paper decides to build a model. Assuming that having $1000 available to buy gold or bitcoin on September 11, 2016, a daily money flow prediction model for gold and bitcoin is built based on historical data, and then a Markov zone shift model is used to determine the hypothetical bull and bear markets, and a machine learning algorithm is used to make decisions and optimize investments to obtain the optimal investment strategy.

2. Gold and bitcoin price float prediction based on the XGBoost model

As there are missing historical data and no weekend data, interpolation is used to fill in the data to make a complete time series.

XGBoost is the abbreviation of "Extreme Gradient Boosting". The XGBoost algorithm is a kind of synthesis algorithm that combines basis functions and weights to form a good data-fitting effect. Considering the XGBoost model has the advantages of strong generalization ability, high scalability, and fast computing speed, it has been welcomed by the fields of statistics, data mining, and machine learning since it was proposed in 2015.

For a dataset containing n m dimensions, the XGBoost model can be expressed as:

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i)\,, \quad f_k \in F \quad (i = 1,2,\ldots n) \tag{1}$$

Among them, $F = \{f(x) = w_{q(x)}\}$ $(q\colon R^m \to \{1,2,\ldots,T\}, w \in R^T)$ is the set of CART decision tree structure, q is the tree structure of samples mapped to leaf nodes, T is the number of leaf nodes, and w is the real fraction of leaf nodes. When building the XGBoost model, it is necessary to find the optimal parameters according to the principle of minimizing the objective function to establish the optimal model. The objective function of the XGBoost model can be divided into the error function term L and the model complexity function term Ω. The objective function can be written as:

$$Obj = L + \Omega \tag{2}$$

When using the training data to optimize the training of the model, it is necessary to keep the original model unchanged and add a new function f to the model to reduce the objective function as much as possible. The specific process is as follows [1]:

$$\begin{gathered} \widehat{y}_i^{(0)} = 0 \\ \widehat{y}_i^{(1)} = \widehat{y}_i^{(0)} + f_1(x_i) \\ \widehat{y}_i^{(2)} = \widehat{y}_i^{(1)} + f_2(x_i) \\ \cdots\cdots \\ \widehat{y}_i^{(t)} = \widehat{y}_i^{(t-1)} + f_t(x_i) \end{gathered} \tag{3}$$

Where: $f_t(x_i)$ is the predicted value of the t-th model, and $\widehat{y}_i^{(t)}$ is the new function added at the t-th time. At this time, the objective function is expressed as:

$$Obj^{(t)} = \sum_{i=1}^{n} (y_i - (\widehat{y}_i^{(t-1)} + f_t(x_i)))^2 + \Omega \tag{4}$$

In the XGBoost algorithm, to quickly find the parameters that minimize the objective function, a second-order Taylor expansion is performed on the objective function to obtain an approximate objective function.

$$Obj^{(t)} \approx \sum_{i=1}^{n} [(y_i - \hat{y}^{(t-1)})^2 + 2(y_i - \widehat{y}_i^{(t-1)}) f_t(x_i) - h_i f_t^2(x_i)] + \Omega \tag{5}$$

When the constant term is removed, it can be seen that the objective function is only related to the first and second derivatives of the error function. At this point, the objective function is expressed as:

$$Obj^{(t)} \approx \sum_{i=1}^{n} [g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \gamma T + \frac{1}{2} \sum_{j=1}^{T} w_j^2 = \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2] + \gamma T \tag{6}$$

If the structural part q of the tree is known, the objective function can be used to find the optimal w_j, and the optimal objective function value can be obtained. Its essence can be classified as the problem of solving the minimum value of the quadratic function. Solutions can be like this:

$$w_j^* = \frac{-\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \tag{7}$$

$$Obj = -\frac{1}{2} \sum_{j=1}^{T} \frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} \tag{8}$$

Obj is a scoring function that can be used as an evaluation model. The smaller the Obj value is, the better the model effect will be. By recursively calling the above tree-building method, a great number of regression tree structures can be obtained, and Obj is used to search for the optimal tree structure and put it into the existing model to establish the optimal XGBoost model [2].

After removing outliers, we added the genetic algorithm to get the following results:

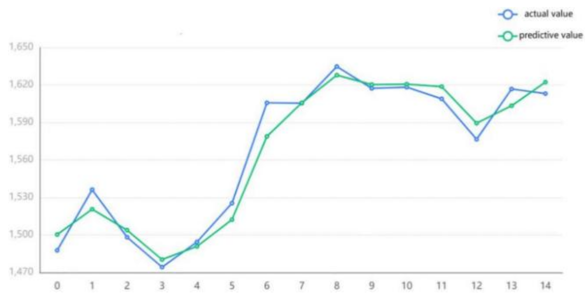

Fig. 1 Fitted curve

The prediction accuracy R^2 is between 0.998 and 0.999, which is a good fit.

3. Investment planning models

3.1 Identification rules

The identification and prediction of bull and bear markets are completed by the rule-based method in two steps, and the model can be used to identify and predict at the same time. To this end, this section is divided into two parts, and the Marko demarcation transformation introduces recognition rules and prediction methods, respectively. Among them, the identification of bull and bear markets is done through rule sets, including LT and PS algorithms [3].

(1) LT algorithm

In the algorithm of Lunde and Timmermann (2004), peaks and troughs must meet minimum size requirements to be a bullish and bearish transition. A bull market is between a trough and a subsequent peak, and a bear market is between a trough and a subsequent trough. If the index is up λ_1 since the last trough, then a bull market is here. If the index has fallen λ_2 since the last peak, then a bear market is in. Identifying peaks and troughs in an exponential sequence $\{P_t\}_{t=1}T$ requires the following iterative search process starting from the peaks or troughs:

1. If the last observation was a peak, its value is set to P^{max}. Consider the follow-up period:

① If the index exceeds P^{max}, then replace P^{max}.

② If the index drops λ_2, then the trough has been found.

③ If the above conditions are not satisfied, then it will not be updated.

2. If the last observation was a trough, its value is set to P^{min}. Consider the follow-up period:

① If the index is lower than P^{min}, then replace P^{min}.

② If the index goes up λ_1, then the peak is found.

③ If the above conditions are not satisfied, then it will not be updated.

Now we will set λ_1=0.20, and λ_2=0.15, indicating that if the stock index rises by more than 20%, considering it as a bull market, and if it falls by 15%, it means a bear market. To initiate the search process, it is necessary to determine whether the market is initially bullish or bearish. We count the number of times the maximum and minimum values of the index have to be adjusted since the first observation. If the highest price had to adjust three times first, the market started bullishly, otherwise, it started bearishly.

(2) PS algorithm

In the algorithm of Pagan and Sossounov (2003), the peaks and troughs are also used as the transition points between bullish and bearish markets. However, the PS algorithm does not require the magnitude of the index change but specifies the minimum duration of the bull and bear market cycles. The algorithm is divided into five steps:

① Find all local maxima and local minima in the exponential sequence. The so-called local maxima are the value that is larger than both the previous and future $\mathrm{r}_{\mathrm{window}}$ periods, and the local minimum refers to a value that is smaller than both the previous and future $\mathrm{r}_{\mathrm{window}}$ periods.

② Construct a sequence of alternating maximum and minimum values by picking the largest maximum value and the smallest minimum value.

③ Review the peaks and troughs during the first and last two $\mathrm{r}_{\mathrm{censor}}$ periods.

④ Remove cycle less than $\mathrm{r}_{\mathrm{cycle}}$ bull and bear market.

⑤ Remove adjacent bull and bear market cycles and less than r_{phase}, unless the price changes by more than ζ.

Among them, set r_{window}=24, r_{censor}=52, r_{phase}=12, ζ=0.2. This paper reviews the first and last 13 weeks, not the 26 weeks taken by Pagan and Sossounov (2003). Reviewing 26 weeks means that a bear or bull market can only be determined after half a year, which seems like a long time, so this article uses a shorter period of 13 weeks to determine the initial and final state of the market.

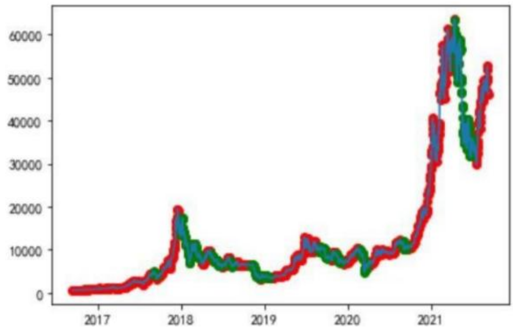

Fig. 2 Bear and Bull Market Forecast

3.2 The Markov Regime Switching Model

The determination and description of any cycle should first isolate the inflection points in the time series, and then divide the expansion and contraction periods based on the information on the inflection points, with the Markov switching model proposed by Hamilton being the most representative. the MS model describes the different characteristics and properties of economic behavior under different phases, states or mechanisms, so the MS model can also be called the Regime Switching Model (RS) [4].

The Markov regime switching model is essentially different from the rule-based and algorithm-based methods of the previous section. The identification and prediction of rule-based methods need to be completed in two steps, while the Markov regime switching models only need to be completed in one step. The model approach constructs a stock market index generation process that can be bullish and bearish over time. According to such a model, under a given stock index, probabilistic inferences about the state of the market bullish and bearish are generated.

The rule-based approach is simple, but not necessarily straightforward. Compared to this, the Marko designing transformation model has several advantages. First, the number of states of the stock market in the model can be expanded. The rule-based method can only classify the stock market state into a bull market and bear market, while there can be 3 or more stock market states in the model, such as bear market rebound and bull market pullback. Secondly, the model method research is more in-depth, and the theoretical properties of the model can be deduced to check whether it conforms to the actual characteristics of the stock market. Finally, by comparing the model coefficients and parameters research, summarize to make it easier to compare results across markets and over time. However, models are also flawed, and incorrect specifications, especially mistakes in the data generation process (omissions or misspecifications) can have a serious impact on the results. Because rule-based methods do not make strict assumptions of the distribution or the presence or absence of time-varying variables, they may be more reliable.

The Markov regime switching model in this paper considers the case where the number of states is 2 or 3, with a numerical representation after the abbreviation, such as RS2- and RS3-. Also consider the cases where the transition probabilities are constant and time-varying, denoted by the suffixes C and L, respectively. For example, the RS3L flag indicates that there are 3 states and time-varying transitions

A Probabilistic Markov regime switching Model. If the number of states is 2, it is divided into a bull market and a bear market. If the number of states is 3, it is divided into a bull market, a weak bear market, and a strong bear market. Generally, bull markets have positive returns and low volatility. Bear markets have negative yields and high volatility. In the case of 3 states, the bear market is divided into weak bear market and strong bear market according to the relative level of volatility. Use S^m to represent a collection of states.

It is assumed that the excess index returns r_t in each state S follow the excess stock index distribution, and each state has a specific mean and variance.

$$r_t \sim N(\mu_s^m, \omega_s^m), s \in S^m \tag{9}$$

Since the distribution of stock market returns is asymmetric and has thick tails, it contains positive. These characteristics are well represented by the distributional zoning transformation model.

Since the actual state of gold and bitcoin cannot be directly observed, it can be viewed as a latent variable that follows a first-order Markov chain with a time-varying transition matrix P_t^m. This matrix contains parameters:

$$\pi_{qst}^m \equiv Pr[S_t^m = s | S_{t-1}^m = q, z_{t-1}], s, q \in S^m \tag{10}$$

Of course, there are $\forall q, \forall t: \sum_{S \in S^m} \pi_{qst}^m = 1$ constraints. When the transition probability is constant, the constraints are such that the $k(k-1)$ orders free parameters are to be estimated. If transition probabilities are time-variant, use a multinomial Logit function:

$$\pi_{qst}^m = \frac{e^{\beta_{qs}^{m'} z_{t-i}}}{\sum_{S \in S^m} e^{\beta_{qs}^{m'} z_{t-i}}}, s, q \in S^m \tag{11}$$

Among that, $\forall q \exists s: \beta_{as} = 0$, make sure to be identified. Finally, set parameters for the probabilities of various states at the start of the model, $\zeta_s^m \equiv Pr[S_1^m = s]$. This paper refers to the parameter setting proposed by Hamilton (1994). At the same time, the constraints on $\Sigma sgs^m \zeta_s^m = 1$ should be satisfied. Treat the remaining parameters as free parameters and estimate them. The parameters of the resulting zoning conversion model are then estimated using the EM algorithm proposed by Dempster et al. (1977).

The EM algorithm (Expectation-maximization algorithm), also known as the maximum expectation algorithm, is an algorithm for finding the maximum likelihood estimation or maximum posterior estimation of parameters in a probability model, where the probability model depends on unobservable hidden variables. The expectation-maximization algorithm is computed alternately in two steps:

The first step is to calculate the expectation (E), using the existing estimates of the hidden variables to calculate their maximum likelihood estimates.

The second step is to maximize (M), maximizing the maximum likelihood value obtained at step E to calculate the value of the parameter. The parameter estimates found at step M are used in the calculation of the next step E, which alternates continuously. Here, π_{qt}^m is optimized by first calculating the maximum likelihood estimate by ζ_s^m. This process is carried out alternately and continuously optimized, and finally the probability of bull and bear markets at time t is obtained.

Then with the help of the DQN, we start to plan our strategy. First, the target network solves a "regression problem" where the input is the state of the environment and the output is the different values (action values) generated by multiple actions. After determining both the input and output of the network, the next step is to solve the problem of how to update these parameters in the network.

The idea is to use the temporal difference method based on the Bellman equation to make the difference between the target network and the network used for training (referred to as the agent network) as close as possible to the income value, which refers to the value from the current state through the benefit of reaching the next state after a decision. It is worth noting that the parameters of the target network in DQN are directly copied to the parameters of the agent network, using the same network structure. However, in the actual training process, the agent can only be trained by fixing the output of the target network, and the method of fixing the output of the network is to delay the update of the parameters of the target network so that the output remains unchanged in the fixed steps, thereby effectively activating the agent. The parameter update process of the network.

The choice of strategy in Q-learning (assuming that it is a deterministic strategy here) is to select the action that can maximize the action value, which can be expressed in mathematical form as:

$$\pi'(s) = argmax_a Q^{\pi}(s, a) \tag{12}$$

The greedy method is a variant of the greedy algorithm. The specific implementation method is to first let the program generate a random number in the [0, 1] interval from a uniform distribution. If the value is less than the preset 1-, select the action that can maximize the action value, otherwise the action will be selected randomly.

So that the highest return can reach 252424.6982 US dollars.

4. Summary

For the processing of the temporal prediction model, we used the relatively novel XGBoost model for data prediction, which has a better fit and better results. A game theory model is also added in the integration of trade strategies to make the model more realistic. Finally, we also used the DQN algorithm, which is equivalent to an improvement of the traditional Q-learning algorithm.

However, in the game theory model, we did not consider the macro environment, and other variables were ignored. For example, due to some special circumstances, some trading days cannot be traded normally.

References

[1] Scientific Platform Serving for Statistics Professional 2021. SPSSPRO. (Version 1.0.11)[Online Application Software]. Retrieved from https://www.spsspro.com.

[2] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System[J]. ACM, 2016.

[3] Xu Bohuan. Research on the identification and prediction of bull and bear markets in Chinese stock market[D]. Jiangsu:Nanjing University of Finance and Economics, 2020.

[4] Li, Nan. Analysis of economic cycle conditions of China-ASEAN monetary cooperation - based on Markovian zone transfer model[J]. Journal of Xiamen Institute of Technology,2015,23(6):18-24.