Received: 29 August 2023 | Revised: 17 November 2023 | Accepted: 29 November 2023 | Published online: 6 December 2023

RESEARCH ARTICLE

Journal of Computational and Cognitive Engineering
2023, Vol. 00(00) 1–13
DOI: 10.47852/bonviewJCCE32021590

Online Adaptive Asset Tracking Algorithm with Ordinal Information

Feitong Lai[1], Chuxin Liang[1], Cuiyin Huang[1] and Hongliang Dai[1,*]

[1]*School of Economics and Statistics, Guangzhou University, China*

Abstract: In recent years, an increasing number of researchers have applied machine learning techniques to online portfolio selection (OLPS), aiming to improve the efficiency and effectiveness of portfolio management in the digital field. In this study, we design and implement a novel OLPS algorithm called "online adaptive asset tracking algorithm" (OAAT). Compared to the peak price tracking (PPT) algorithm, it complements more historical information of assets in the investment portfolio and provides a more effective solution for parameter selection of the PPT algorithm. Firstly, the OAAT algorithm updates investment proportions by tracking the historical information of assets, which includes recent peak prices, historical returns, and historical volatility. Secondly, the OAAT algorithm optimizes parameters through online learning. The initial parameters are selected based on the minimum sum principle of the ordinal information. After each phase of trading, the parameters are optimized through the gradient descent algorithm, and the average values of the optimal parameters in the last 5 days are used as the parameters of the next phase. Finally, with the optimized parameters and the tracked asset information, the fast error backpropagation algorithm outputs the investment ratio through gradient projection. Compared with the benchmarks, follow-the-winner, follow-the-loser, and pattern-matching-based algorithms under four Hong Kong stock index constituents data sets and three US stock index constituents data sets, the empirical comparative analysis and statistical test show that the OAAT algorithm can effectively determine the investment proportion to balance return and risk.

Keywords: online portfolio selection, asset information tracking, ordinal information, gradient descent, gradient projection

1. Introduction

Portfolio selection algorithms refer to decision-making methods rationally allocation and selection of the assets for investors under uncertain financial market environment (Zhou, 2018). After Markowitz (1952) proposes the development of mean-variance model, many researchers have carried out in-depth studies in the field of static modeling (Esfahani et al., 2016). However, due to the volatility of the financial market environment, investors need to constantly adjust their investment algorithms based on historical information for achievement of the goal of maximizing returns. On the basis of this fact, portfolio selection is also a dynamic issue. Therefore, researchers focused on the dynamic research work and achieved fruitful results in this field (Moghaddam et al., 2016). Meanwhile, with the gradual discovery of "abnormal" behavior in financial markets, behavioral finance has been blooming. In addition, with the rapid development of computer technology (Ma et al., 2021), sequence data prediction algorithm (also known as online learning) based on machine learning provides more robust and precise solutions for portfolio selection research (Tiwari et al., 2020). Therefore, portfolio research based on behavioral finance and online learning technology has become a new research direction and discipline frontier, namely as online portfolio selection (OLPS).

In financial markets, it is observed that along with the reversal effect, there is also a momentum effect present. Relevant studies show that investors' behavior does not obey the hypothesis of efficient market due to the abnormal environment. Therefore, they do not meet the hypothesis of mean regression of securities prices (Shiller, 2003). In such situation, investors will likely push securities that have performed better in the previous period higher. Therefore, Lai et al. (2018) proposed a peak price tracking (PPT) algorithm based on the inversion method. PPT algorithm, a follow-the-winner algorithm, measures the possible price of securities in the future according to the highest price of securities within a specific time window, aiming to capture the potential benefits. This algorithm uses the error backpropagation (BP) algorithm to optimize the objective, which can be applied to large-scale and rapid trading (Brahma et al., 2016; Lin et al., 2016; Raitoharju et al., 2016). Compared with some typical OLPS algorithms, it has better performance and can achieve greater wealth gains. However, the PPT algorithm only feedbacks the recent peak price information into the investment ratio, such information is not enough. In addition, the parameters of PPT algorithm are difficult to determine directly. To address these dual challenges, we put forward a novel online adaptive asset tracking (OAAT) algorithm with ordinal information for OLPS. The primary contributions of this study can be summarized as follows:

- The OAAT algorithm updates investment proportions by tracking the historical information of the assets, which includes not only recent peak price but also the historical return and volatility of each asset.

*Corresponding author: Hongliang Dai, School of Economics and Statistics, Guangzhou University, China. Email: hld618@gzhu.edu.com

- The OAAT algorithm optimizes its parameters through online learning. After each trading period, the algorithm utilizes the gradient descent algorithm to optimize two parameters, related to historical information, and update step size separately.
- The OAAT algorithm determines the initial parameters in the online learning algorithm based on the minimum sum principle of the ordinal information in the training set, this helps the gradient descent algorithm to find an approximate optimal solution.

This paper is structured as follows: Section 2 provides an extensive literature review. Section 3 describes the detailed model of the OAAT algorithm. Subsequently, Section 4 presents the comparative analysis of the empirical results of each algorithm. Section 5 conducts a statistical test on the empirical results in the previous section. Finally, Section 6 summarizes the OAAT algorithm and discusses its potential future developments.

2. Literature Review

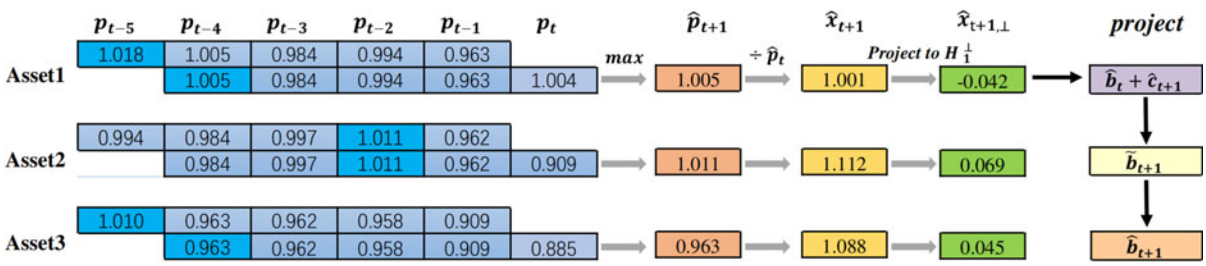
2.1. Typical algorithms for the OLPS

According to the research result of Li and Hoi (2018), they summarized typical algorithms for the online portfolio research, including benchmarks algorithm, pattern-match-based algorithm, follow-the-winner algorithm, follow-the-loser algorithm, and pattern-match-based algorithm. The benchmark algorithms mainly include the buy and hold (BAH) (Cover, 1991) and constant rebalanced portfolio (CRP) algorithms (Li & Hoi, 2018). BAH is an algorithm that does not change the allocation of assets after determination of an appropriate portfolio and holding period. In this way, the portfolio has the advantage of low transaction costs and management costs; however, it also exposes the assets completely to market risks and loses the opportunity of potential profit during the market fluctuations. CRP is an algorithm that readjusts assets allocation at the end of each trading cycle and puts all the assets into the fastest-growing securities. However, in the complex markets, where prices sometimes change rapidly, it becomes too risky to follow the best CRP as an investor's target. In pattern-match-based algorithm, historical data are used to fit and optimize the model for the prediction of future prices (Györfi et al., 2006). Typical pattern-match-based algorithms mainly include the correlation-driven nonparametric learning approach (CORN) (Li & Hoi, 2018), etc. In the follow-the-loser algorithm, it is assumed that stocks will reverse in the future and will reduce the proportion of investment in securities with good returns in the past and will also increase the securities with poor returns in the past. This algorithm is also known as reverse (mean regression) trading algorithm (Borodin et al., 2004). Typical algorithms of this algorithm mainly include the Anticor system (Borodin et al., 2004), the OLMAR (Li & Hoi, 2018), and the robust mean regression (RMR) algorithms (Huang et al., 2016). These algorithms mainly use the average value regression theory and classification perceptron algorithm, and having the advantage of easy calculation. Ortobelli Lozza et al. (2022) compared the performance of two follow-the-loser algorithms, Passive aggressive mean reversion algorithm (PAMR) and Online portfolio selection algorithm with moving average reversion (OLMAR), with the classical mean-variance method in the US stock market, and discussed their advantages and limitations. The follow-the-winner algorithm generally believes that securities that have shown good performance in the early stages are likely to continue performing well in later stages (Frost et al., 2020). This algorithm is also known as momentum algorithm due to the ability of increment of the holding of the securities with good returns in the past and reduction of the securities with poor returns in the past (Dong & Zhou, 2019). Typical follow-the-winner algorithms mainly includes the universal portfolios (UPs) (Stella & Ventura, 2011), online newton step (Wu et al., 2019), and so on. The UP algorithm does not make any assumptions about the random return distribution of the stock market; however, this algorithm requires multiple integrals calculation. When the number of securities is large, it will bring the disadvantage of higher calculation cost. The current momentum effect systems have shortcomings in effective trend pattern learning mechanisms, making them unable to compete with state-of-the-art reversal effect systems and have certain limitations in capturing excess returns. In addition, there are some other types of OLPS algorithms. Zimmert et al. (2022) introduced a new OLPS algorithm – BISONS, which achieves near optimal regret in the optimal portfolio problem without requiring any assumptions on the gradient. Yao and Zhang (2023) introduce an OLPS optimization algorithm utilizing an elastic-net for regularization and a linearized augmented Lagrangian method for solution, which demonstrates superior efficiency and performance compared to state-of-the-art algorithms on benchmark data sets. Zhang et al. (2023) propose a novel OLPS algorithm that uses mirror descent and Bregman divergence to obtain low dimension regret and computational complexity, outperforming other algorithms in the Chinese futures market.

2.2. PPT algorithm

Lai et al. (2018) introduced the PPT algorithm as a solution to the limitations of current follow-the-winner algorithms in predicting asset prices. The algorithm identifies the highest asset prices within a specified time window width and considers them as the potential peak prices for the next period. To maximize cumulative wealth (CW), the forecasted price is then incorporated into the investment proportion using the fast error BP algorithm. Figure 1 illustrates the PPT algorithm.

Figure 1
Diagram of the PPT algorithm



Building on their previous work, Dai et al. (2022) introduced the trend promote price tracing (TPPT) in 2022, which applies a unique slope-based method to effectively predict upcoming price trends. Compared to the previous PPT algorithm, TPPT boasts significantly improved price tracking capabilities. Dai et al. (2023) developed an adjusted PPT approach. By introducing specific parameters, the algorithm is able to fine-tune the influence of peak price and residual factor, resulting in a more accurate prediction of future prices.

2.2.1. Parameter selection of the PPT algorithm

The PPT algorithm sets the highest prices of assets within the time window as the potential highest prices for the next period and then uses the fast error BP algorithm to feed back the potential highest prices into the investment ratio to maximize CW (Lai et al., 2018). In the PPT algorithm, we need to set two parameters, namely the window width of recent peak price (w) and the update step size parameter (ε). First of all, in the usual setting, the window width of recent peak price is represented as $w = 5$, which is suitable for the stock's 5-day moving average (Cover, 1991; Huang et al., 2016; Li & Hoi, 2018). In previous studies, short-term asset price analysis based on the recent 5-day window can provide significant and dependable information. In addition, both active and defensive investors tend to adopt the 5-day moving average to specify the investment algorithm. Therefore, $w = 5$ is set. Another parameter ε describes the updating step size of the investment portfolio, which is set based on testing the performance at different scales such as 5, 10, 50, 100, 500, and 1000.

2.3. BP algorithm

The fast error BP algorithm is a kind of simple and explicit matrix calculation method, which has certain application value in large-scale and time-limited cases. Lin et al. (2016) proposed an innovative and efficient network that utilizes a new type of type-2 fuzzy cerebellar model articulation controller as the built-in structure of the adaptive filters to solve some complex signal processing problems. Raitoharju et al. (2016) conducted research on the training process of Radial basis function neural network (RBFNN) and compared the differences in classification performance and computational efficiency between class-specific, input, and input–output clustering methods. The research results indicate that using class-specific clustering methods to train radial basis function neural networks can significantly reduce the complexity of overall clustering. Brahma et al. (2016) proposed multiple metrics to measure manifold entanglement based on specific assumptions and conducted experiments using both synthetic and real-world data sets to validate their approach. However, to solve the portfolio vector, the BP algorithm in the PPT algorithm is different from the recent BP algorithms above (Lai et al., 2018). Lai et al. (2018) devises a fast BP algorithm through the gradient projection. Instead of the output bias, it feedbacks the increasing power (Bertsekas, 1997).

2.4. Online learning

Online learning refers to a learning process in which it is not necessary to fully obtain all data samples, but instead, new samples from data streams are processed incrementally, and the learning algorithm of the model is updated in real-time (Singh & Thurman, 2019). Online learning also has many applications in the financial field. Tantisripreecha and Soonthomphisaj (2018) proposed an online learning method, called LDA-Online algorithm, for predicting stock trends. In both batch processing and online learning scheme, LDA-Online algorithm demonstrated the most superior performance in the study. Salas (2020) developed a Bayesian treatment of the online passive-aggressive and gradient descent algorithms, implementing uncertainty modeling, probabilistic predictions, and automatic, data-dependent hyperparameter tuning. Soleymani and Paquet (2021) conducted online learning through a passive concept drift method to manage unforeseen variations in the data distribution, allowing for real-time processing and updating of the DeepPocket model. Li et al. (2023) converted the online factor selection task into an online learning challenge, effectively balancing the cost and accuracy in an online portfolio algorithm.

3. Online Adaptive Asset Tracking Algorithm with Ordinal Information

3.1. OLPS problem setting

The basic algorithmic structure of the OLPS algorithm is illustrated by the following trading scheme: it is assumed that the investor intends to invest in m types of assets and each type of assets has n transactions, and the asset price of phase t is denoted as $\mathbf{p_t} = (p_t^1, p_t^2, \cdots, p_t^m) \in R_m$. The relative price of phase t is denoted as:

$$\mathbf{x_t} = (x_t^1, x_t^2, \cdots, x_t^m) \in R_m,\ t = 1, 2, \ldots, n \tag{1}$$

where R_m is the set of vectors in m-dimensional space where all components are positive real numbers; $x_t^i = \frac{p_t^i}{p_{t-1}^i}$ is the relative price of phase t of the ith assets. Therefore, the net investment of the ith assets at phase t is x_t^i; therefore, the investment in the ith assets in phase t will be increased by a factor. The price changes in the financial market from t_1 to t_2 are denoted as $\mathbf{x_{t1}^{t2}}$ and $\mathbf{x_{t1}^{t2}} = \{x_{t1}, \cdots, x_{t2}\}$. So the price change in the market through the course of the transaction is

$$\mathbf{x_1^n} = \{x_1, \cdots, x_n\} \tag{2}$$

At the beginning of period t, the investment ratio of assets can be represented by the vector $\mathbf{b}_t$, whereas $\mathbf{b}_t^{(i)}$ represents the proportion invested in the ith assets. Obviously, the sum of the weights equals to 1; therefore, in the case of restricted short selling, it can be stated that the weight of the portfolio vector is always non-negative, namely:

$$\mathbf{b}_t \in \Delta_m, \Delta_m = \mathbf{b}\ : \mathbf{b} > \mathbf{0}, \mathbf{b}^T\mathbf{1} = 1 \tag{3}$$

An investment algorithm consisting of n trading period is represented by $\mathbf{b_1^n} = \{\mathbf{b}_1, \cdots, \mathbf{b}_n\}$. In the portfolio vector $\mathbf{b}_t$, $\mathbf{b}^T 1$ is the dot product of b and 1 with a value of 1. At the start of the trading period t, capital can be allocated to various assets in accordance with the corresponding ratio and then change as the relative price of each asset. The capital growth multiple can be expressed as $\mathbf{b}_t^T\mathbf{x}_t = \sum_{i=1}^m b_t^i x_t^i$.

Since there are T trading periods, the proceeds from the previous period and also can be invested in the next trading period. The relative price reinvestment principle is adopted in the model. Therefore, the cumulative return of the portfolio after the end of the nth trading period can be calculated by multiplying the net worth with the growth multiple of each period as follows, which represents the combined effect of the portfolio's growth over multiple periods.

$$S_n(\mathbf{b}_1^n) = S_0 \prod_{t=1}^{n} \mathbf{b}_t^T \mathbf{x}_t = S_0 \prod_{t=1}^{n} \sum_{i=1}^{m} b_t^i x_t^i \tag{4}$$

where S_0 is the initial capital, generally set $S_0 = 1$. The exponential growth rate of $\mathbf{b}_1^n$ is defined as $\mathbf{w}_n(\mathbf{b}_1^n)$:

$$\mathbf{w}_n(\mathbf{b}_1^n) = \frac{1}{n}\log(S_n(\mathbf{b}_1^n)) = \frac{1}{n}\sum_{i=1}^{m}\log(b_t^i x_t^i) \tag{5}$$

Portfolio selection enables trading online and dynamically. After the manager determines the best portfolio proportion $\mathbf{b}_t$ based on historical data of assets x_{t-1}^i, the portfolio growth multiple of the current period can be rewritten as $\mathbf{b}_t^T\mathbf{x}_t$. Repeat the above procedure until the final trading period, and the cumulative net worth of S_n can be calculated.

The trading algorithm implemented under this framework has three assumptions:

- This study assumes that there are transaction costs and tax burdens in trading algorithms (transaction cost ($tc = 0.001$) in parameter settings).
- This study assumes high market liquidity, allowing individuals to freely trade assets at the closing price. Therefore, in the selection of stock index constituents data sets, we choose the stocks with high trading volume in the index.
- This study assumes that any OLPS algorithm has no impact on the market behavior.

3.2. Asset information tracking algorithm

The asset information tracking (AIT) algorithm enhances the PPT algorithm by incorporating additional historical asset information, including historical returns and volatility.

3.2.1. Asset information model

Due to the different dimensions of different stocks, it is difficult to directly compare the changes of stock prices to reflect the actual changes of different stocks. Therefore, the relative price vector is used to eliminate the influence of different stock dimensions, which is expressed as;

$$\mathbf{x}_t = \frac{\mathbf{p}_t}{\mathbf{p}_{t-1}} \tag{6}$$

According to follow-the-winner principle and the PPT algorithm, the OAAT algorithm sets the highest stock price in the historical window as the highest possible price for the next period. It is expressed as

$$\hat{\mathrm{P}}_{t+1}^i = \max_{0\le k\le \omega_1 - 1} \mathrm{P}_{t-k}^i, i = 1, 2, \cdots, m \tag{7}$$

where P_{t-k}^i represents the price of the ith stock under the previous time window; $\hat{\mathrm{P}}_{t+1}^i$ represents the highest possible price of the ith stock in the next period; and ω_1 represents the time window of recent peak price (usually set to 5).

In addition to the recent peak price, we also incorporate the historical return and volatility of each asset to complement historical information. Since the proposition of the mean-variance model, researchers have often used the mean to describe return and the variance to characterize volatility (Markowitz, 1952). We have also adopted this approach in our analysis. Therefore, the model we have constructed to describe historical asset information is as follows:

$$\mathrm{H}_{t+1}^i = \frac{\sqrt{\sum_{t-\omega_2+1}^{t}\left(\frac{\mathrm{p}_t^i}{\mathrm{p}_{t-1}^i} - 1\right)^2}}{\frac{1}{\omega_2}\sum_{t-\omega_2+1}^{t}\left(\frac{\mathrm{p}_t^i}{\mathrm{p}_{t-1}^i}\right)}, i = 1, 2, \cdots, m \tag{8}$$

where $\sqrt{\sum_{T-\omega_2+1}^{T}\left(\frac{\mathrm{p}_t^i}{\mathrm{p}_{t-1}^i} - 1\right)^2}$ refers to the standard deviation of the relative price of the ith assets, that is, the historical volatility, $\frac{1}{\omega_2}\sum_{T-\omega_2+1}^{T}\left(\frac{\mathrm{p}_t^i}{\mathrm{p}_{t-1}^i}\right)$ refers to the mean value of the relative price of the ith assets, that is, the historical return, H_{T+1}^i represents the historical information of ith assets, and ω_2 represents the time window of historical information (usually set to 120 (half year) or 240 (1 year)).

To sum up, the asset information model we constructed is expressed as:

$$\hat{\mathbf{x}}_{t+1} = \frac{\hat{\mathbf{P}}_{t+1}}{\mathbf{p}_t} + \alpha\mathbf{H}_{t+1} \tag{9}$$

where $\hat{\mathbf{P}}_{t+1}$ represents the highest possible price in the next period, $\mathbf{p}_t$ represents the stock price in the current period, $\mathbf{H}_{t+1}$ represents the historical information, and α represents the historical information parameter.

Due to the presence of both momentum and reversal effects in financial markets, it is challenging to determine whether historical information has a positive or negative impact on assets, as well as to set the strength of its effect. Therefore, the setting of α is optimized by online learning algorithm, allowing for updates in each period. Detailed online learning methods of parameter optimization are described in Section 3.4.

3.2.2. Solving the portfolio vector with the BP algorithm

In the framework of the portfolio management, the primary objective is to accumulate more wealth over time. By ensuring that the growth factors at each stage are as optimal as possible, the algorithm is able to gain as much CW as possible. Therefore, the optimization goal of maximizing the wealth accumulation can be expressed as:

$$\hat{\mathbf{b}}_{t+1} = arg \max_{\mathbf{b}\varepsilon\Delta_d} \mathbf{b}_{t+1}^T\hat{\mathbf{x}}_{t+1}, s.t. \parallel \mathbf{b} - \hat{\mathbf{b}}_t \parallel \le \varepsilon,\ \varepsilon > 0 \tag{10}$$

In the process of using the BP algorithm to solve the portfolio ratio problem, we decomposed the investment proportion of the next phase $\mathbf{b}_{t+1}$ into two parts: the current investment proportion $\mathbf{b}_t$ and an additional increment $\mathbf{c}_{t+1}$. This decomposition enables us to further optimize the objective function:

$$\mathbf{b}_{t+1} = \mathbf{b}_t + \mathbf{c}_{t+1} \tag{11}$$

$$\mathbf{1}^T\mathbf{c}_{t+1} = 1\mathbf{b}_{t+1} - 1\hat{\mathbf{b}}_t = 0 \tag{12}$$

$$\parallel \mathbf{c}_{t+1} \parallel = \parallel \mathbf{b}_{t+1} - \hat{\mathbf{b}}_t \parallel \le \varepsilon \tag{13}$$

$$\max_{\mathbf{b}_{t+1}} \mathbf{b}_{t+1}^T\hat{\mathbf{x}}_{t+1} \Leftrightarrow \max_{\mathbf{c}_{t+1}} \mathbf{c}_{t+1}^T\hat{\mathbf{x}}_{t+1} \tag{14}$$

Then, based on the fast BP algorithm, we further decompose the asset information vector into perpendicular and parallel sections and solve the increment. The specific process is as follows:

$$If\ \hat{\mathbf{x}}_{t+1,\perp} = 0, then\ \hat{\mathbf{c}}_{t+1} = 0 \tag{15}$$

$$Otherwie\ \hat{\mathbf{x}}_{t+1,\perp} \neq 0, \hat{\mathbf{c}}_{t+1} = \frac{\varepsilon \hat{\mathbf{x}}_{t+1,\perp}}{\| \hat{\mathbf{x}}_{t+1,\perp} \|} \tag{16}$$

Finally, we feed back the increment to the investment vector proportion and obtain the next investment proportion $\mathbf{b}_{t+1}$ by projecting it onto a space with a value range from 0 to 1. This step can be described as follows:

$$\tilde{\mathbf{b}}_{t+1} = \hat{\mathbf{b}}_t + \hat{\mathbf{c}}_{t+1} \tag{17}$$

$$\hat{\mathbf{b}}_{t+1} = arg \min_{\mathbf{b}_t \varepsilon \Delta_d} \left\| \mathbf{b} - \tilde{\mathbf{b}}_{t+1} \right\|^2 \tag{18}$$

3.3. Online adaptive parameter optimization

3.3.1. The setting of the initial parameter with ordinal information

In the OAAT algorithm, we have two parameters that need to be updated through online learning: historical information parameter α and update step size parameter ε. Due to our inability to determine whether historical information has a favorable or unfavorable influence on assets, we set the initial point of the parameter α as 0. However, we still need to examine the initial update step size parameter ε for the portfolio. In the training set, we use an ordinal information method to determine the initial parameter.

To solve the average order value by "rank" in the nonparametric test, firstly, the values are sorted, and then the corresponding average ranking score are calculated (Delgado & Song, 2018; Huang et al, 2016). Based on the test results of each algorithm in each data set, the sorting value of the ith data set in the jth algorithm is expressed as r_i^j. After sorting, the corresponding value is assigned, for example, the sorting value of the largest CW is 1 and so on. The average order values of the algorithm under each data set are calculated according to the order values, and the average order values of the jth algorithm are recorded as

$$D_j = \frac{1}{m} \sum_{i=1}^{m} r_i^j \tag{19}$$

For evaluation of the effect of indicators, the difference between each order value and the first-order value is used to measure the gap between the best performance and the current performance as follows:

$$S_j^* = \sum_{i=1}^{m} \left(\max_{\Delta_\varepsilon} S_{t+1}^{(i)} - S_{t+1,j}^{(i)} \right) \tag{20}$$

According to the parameter performance of average order value and order value gap feedback, the initial point of the update step size parameter ε is optimized in the training set as follows:

$$\varepsilon = arg \min_{j \in \Delta_\varepsilon} \left(D_j + S_j^* \right) \tag{21}$$

3.3.2. Online learning and gradient descent

We employ online learning and gradient descent algorithm for parameter optimization in this section. Online learning is an iterative approach that continuously updates model parameters based on real-time data. Compared to traditional batch learning, online learning offers higher efficiency and lower storage requirements, making it particularly suitable for handling large-scale data sets. Through online learning and gradient descent, we achieve continuous optimization and learning of model parameters. During the iterative process, we continually input new assets prices data to the model, compute gradients through portfolio vector, and update parameters based on the direction and magnitude of the gradients. Through this iterative optimization, our aim is to obtain optimal model parameters that maximize the performance and generalization capabilities of the model.

Due to the difficulty in solving the objective function of the AIT algorithm directly to obtain gradients, we need to employ a numerical approximation method. After setting appropriate initial points and step sizes, we use the central difference method to calculate gradients for iterative purposes. Through this process, we aim to determine the parameters that maximize the returns of the AIT algorithm given the price fluctuations of current period. The detailed formulas are as follows:

$$\mathrm{grad}_\alpha = \frac{1}{2h_1} \left(AIT(\sim, \alpha + h_1) - AIT(\sim, \alpha - h_1) \right) \tag{22}$$

$$\alpha = \alpha + learningrate_\alpha grad_\alpha \tag{23}$$

$$\mathrm{grad}_\varepsilon = \frac{1}{2h_2} \left(AIT(\sim, \varepsilon + h_2) - AIT(\sim, \varepsilon - h_2) \right) \tag{24}$$

$$\varepsilon = \varepsilon + learningrate_\varepsilon grad_\varepsilon \tag{25}$$

where $grad_\alpha$, $grad_\varepsilon$, respectively, represents the gradient of the parameters α and ε, and $learningrate_\alpha$, $learningrate_\varepsilon$, respectively, represents the learning rate of the parameters α and ε.

Finally, we use the average value of the best parameters in the last five periods as the optimal parameters in the next phase.

$$\alpha_{opt} = \frac{1}{\omega_1} \sum\nolimits_{t-\omega_1}^{t} \alpha_{best,t} \tag{26}$$

$$\varepsilon_{opt} = \frac{1}{\omega_1} \sum\nolimits_{t-\omega_1}^{t} \varepsilon_{best,t} \tag{27}$$

3.4. OAAT algorithm flow

Algorithm 1 outlines the basic flow of the AIT algorithm, while Algorithm 2 describes the fundamental flow of the OAAT algorithm.

Algorithm 1. AIT algorithm

Require: Recent window closing price $\{\mathbf{p}_{t-k}\}_{k=0}^{\omega_1 - 1}$; historical window closing price $\{\mathbf{p}_{t-k}\}_{k=0}^{\omega_2 - 1}$; current portfolio $\hat{\mathbf{b}}_t$; historical information parameter α; update step size parameter ϵ.
Ensure: The next portfolio $\hat{\mathbf{b}}_{t+1}$; the cumulative wealth change for the current period U_{t+1}.
1: Calculate $\hat{\mathbf{x}}_{t+1} = \frac{\hat{\mathbf{P}}_{t+1}}{\mathbf{p}_t} + \alpha \mathbf{H}_{t+1}$ by Equations (6)–(9).
2: Calculate $\hat{\mathbf{x}}_{t+1,\perp} = \left(\mathbf{I} - \frac{1}{m} 11^T \right) \hat{\mathbf{x}}_{t+1}$.
3: **if** $\hat{\mathbf{x}}_{t+1,\perp} \neq 0$ **then**
4: $\hat{\mathbf{c}}_{t+1} = \frac{\varepsilon \hat{\mathbf{x}}_{t+1,\perp}}{\| \hat{\mathbf{x}}_{t+1,\perp} \|}$
5: **else**
6: $\hat{\mathbf{c}}_{t+1} = 0$
7: **end if**
8: Calculate $\tilde{\mathbf{b}}_{t+1} = \hat{\mathbf{b}}_t + \hat{\mathbf{c}}_{t+1}$.
9: Projection: $\hat{\mathbf{b}}_{t+1} = \arg \min_{\mathbf{b}_t \varepsilon \Delta_d} \left\| \mathbf{b} - \tilde{\mathbf{b}}_{t+1} \right\|^2$.
10: Calculate $U_{t+1} = \hat{\mathbf{b}}_{t+1}^T \hat{\mathbf{x}}_{t+1}$.

Algorithm 2. OAAT algorithm

Require: Train set $\{\mathbf{p}_t^{train}\}$; test set $\{\mathbf{p}_t^{test}\}$; current portfolio $\hat{\mathbf{b}}_t$; initial investment capital S_0; update step size parameter set Δ_ε; historical information parameter α; max optimized iterations I_{max}
Ensure: The optimal parameter α_{opt} and ε_{opt}; the next portfolio $\hat{\mathbf{b}}_{t+1}$; cumulative wealth S_{t+1}
1: **for** $\varepsilon_j \in \Delta_\varepsilon$ **do**
2: **for** $t = 1$ to T_{train} **do**
3: $\hat{\mathbf{b}}_t$ = AIT($\{\mathbf{p}_t^{train}\}$, ε_j, α=0, $\hat{\mathbf{b}}_{t-1}$)
4: Calculate $S_{t+1,\mathrm{opt}} = S_0 \prod_{t=0}^{n-1} \hat{\mathbf{b}}_{t+1}^T \mathbf{x}_{t+1}$
5: **end for**
6: **end for**
7: Calculate the average order values and by Equations (17) and (18), Optimize the initial update step size parameter ε by Equation (19).
8: **for** $t = 1$ to T_{test} **do**
9: $U_{t+1}, \hat{\mathbf{b}}_t$ = AIT($\{\mathbf{p}_t^{test}\}$, ε_j, α, $\hat{\mathbf{b}}_{t-1}$)
10: Calculate $S_{t+1} = S_t U_{t+1}$
11: **for** i =1:I_{max} **do**
12: Optimize α and ϵ through gradient descent by Equations (23)–(27), and get optimal parameters α_{opt} and ε_{opt}
13: **end for**
14: **end for**

3.5. OAAT flow chart

The flow chart depicting the OAAT algorithm is presented in Figure 2.

4. Experimental Analysis

In this paper, 10 classical portfolio algorithms are selected for experimental analysis on four Hong Kong stock index data sets and three US stock index data sets to compare the effectiveness and stability of the OAAT algorithm. The experimental result of six of these algorithms (UBAH, UP, Anticor, EMA, RMR, CORN-U) is solved by the OLPS toolbox (Li et al., 2016). The experimental environment of this paper is Core (TM) i5-7300HQ CPU and 32GB memory card; programming language: MATLAB.

4.1. Introduction of the data sets

We collect four constituent stock data sets of Hong Kong indexes (from January 1, 2017 to December 31, 2022 (5 years totally)) and three constituent stock data sets of US indexes (from January 1, 2017 to December 31, 2021 (4 years totally)) for experimental analysis, as shown in Table 1. The selected stocks are all characterized by high trading volumes in the indexes.

4.2. Introduction to the OLPS algorithms

The selected OLPS algorithms in this experiment are presented in Table 2. Time complexity of these algorithms in the table refers to the summary and analysis of the research work from Li and Hoi, (2018). Time complexity of the algorithm provides an important basis for the efficiency of the algorithm, which has great importance on the reference value for the investment transaction process, where m represents the number of assets; n represents the number of trading periods; and M represents the maximum iteration count. Due to the OAAT algorithm uses online learning

Figure 2
The flow chart of OAAT algorithm

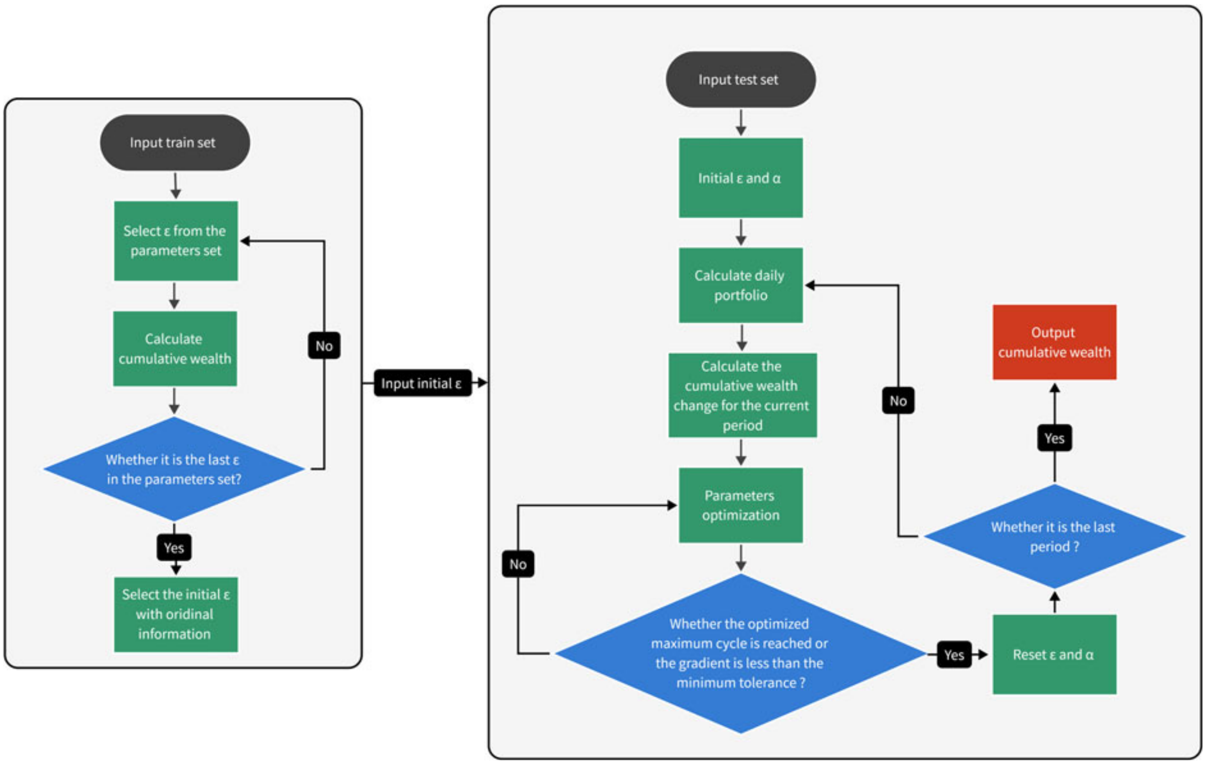


Table 1
The summary of four Hong Kong data sets and three US data sets used for this experimental analysis

Data set	Abbreviation	Main constituent stocks	Number of stocks
Hong Kong Hang Seng	HK50	0175, 0288, 0386, 0535, 0728, 0857, 0883, etc.	18
Hang Seng Composite Industry	HSCIIG	0144, 0152, 0489, 0586, 0631, 0658, 0732, etc.	17
Hang Seng Composite Industry Telecom	HSCIT	0008, 0215, 0728, 0762, 0941, 1883, 6823	7
Hang Seng Composite Industry Utilities	HSCIU	0002, 0003, 0006, 0135, 0257, 0270, 0371, etc.	24
S&P500 (Group A)	SP500-A	TSLA, BAC, FRC, AMD, F, AAPL, CCL, etc.	14
S&P500 (Group B)	SP500-B	T, META, USB, RF, C, MU, TFC, CMCSA, etc.	14
NASDAQ 100	NDX	INTC, GOOGL, MSFT, META, MU, CMCSA, etc.	13

Table 2
Summary of algorithms used for experimental analysis

Category	Algorithm	Time complexity	Explanation
Benchmark	UBAH	$o(m+n)$	Buy and hold algorithm
Follow the winner	UP	$o(n^m)$	Pan securities portfolio algorithm
Follow the loser	Anticor	$O(m^2n)$	Anti-Correlation algorithm
Pattern matching	CORN	$o(mn^2)$	Relevance driven nonparametric learning
Follow the loser	EMA	$o(mn)$	Online moving mean regression algorithm
Follow the loser	RMR	$o((M+1)\ mn)$	Robust mean regression algorithm
Follow the winner	PPT	$o(m^2n)$	Peak price tracking algorithm
Follow the winner	TPPT	$o(m^2n)$	Trend promote price tracing ensemble algorithm
Follow the winner	APPT	$o(m^2n)$	Adjusted peak price tracking algorithm
Online learning	OAAT (ours)	$o(Mm^2n)$	Online adaptive asset tracking algorithm

and gradient descent algorithm to optimize parameters, its time complexity is slightly higher than other algorithms.

4.3. Empirical results of parameter optimization

The OAAT algorithm mainly includes four important parameters: transaction cost (tc), the window width of recent peak price (ω_1), historical information parameter (α), and update step size parameter (ϵ). The first two parameters are manually set. It is assumed that the transaction cost rate is 0.001. The window width of recent peak price ω_1 is 5, which is suitable for the stock's 5-day moving average. The other two parameters are dynamically adjusted by our online learning algorithm.

Initial step size ϵ is optimized and adjusted through the ordinal information principle of CW. The specific method presents different results value of ϵ in seven data sets. The more the CW, the better the parameter ϵ is set. Table 3 displays the CW and its corresponding ordinal value under different ϵ for each data set. From Table 3, it is obvious that when $\epsilon = 1000$, the average ranking score and the gap with the first order value are the smallest. Therefore, in the OAAT algorithm, initial step size ϵ will be set as 1000 in subsequent experiments.

We can also observe that when the parameter ϵ is set to 5, the average rank of each data set is also good, but the order value gap is larger. This is because when we set ϵ=5, it means that the update step is small, and the changes in the investment portfolio of are limited. In data sets with lower returns such as HSCIU and SP500-B, this approach helps to reduce transaction costs and control risks. However, this approach is not conducive to pursuing higher returns. Therefore, when ϵ=5, compared to the other settings, the order value gap is the largest.

Then, we optimize the historical information parameter α and update step size parameter ϵ through online learning and gradient descent, and some optimization results are shown in Figure 3. We can see that α and ϵ are adaptively adjusted over time. When α becomes greater than 0, it means that the OAAT algorithm determines that the current market has larger reversal effect than momentum effect. On the other hand, when α becomes less than 0, it means that the OAAT algorithm determines that the current market has larger momentum effect than reversal effect. When α equals 0, it means that the OAAT algorithm does not consider historical asset information into the construction of the current investment portfolio.

4.4. Test set results and analysis

This paper evaluates and compares the performance of the algorithm according to several common indicators. The specific introduction and analysis are as follows.

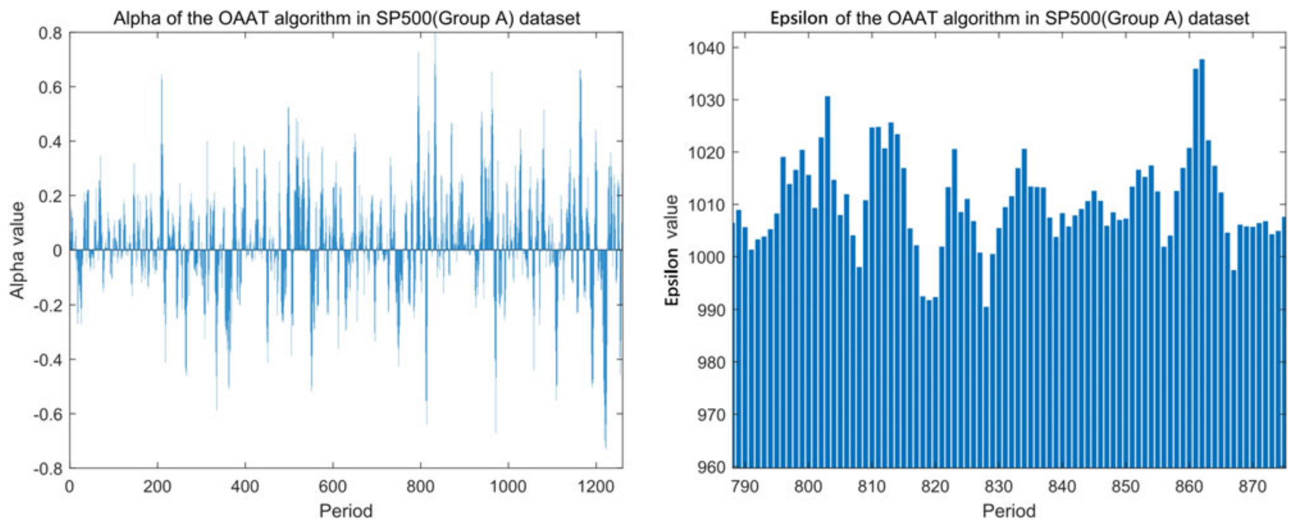
CW quantifies the overall capital growth achieved throughout the entire trading cycle. It is determined by dividing the total capital at the end of the period by the total capital at the start of the period. Table 4 shows that the OAAT algorithm is the most advantageous of 10 algorithms during three data sets including HSCIT, HSCIU, and SP500-B data set. In the remaining four data sets, OAAT algorithm performs little worse. In the HK50, it ranks third with a score of 1.128, falling behind the Anticor-1 and UP algorithms. In the HSCIIG, it ranks second with a score of 1.680, trailing the Anticor-1 and RMR algorithms. Similarly, in the SP500-A data set, the OAAT algorithm comes in second place, just behind UBAH. Lastly, in the NDX data set, the OAAT algorithm ranks second, with RMR algorithm taking the lead. It can be seen that OAAT algorithm is an algorithm with strong profitability.

Table 3
The cumulative wealth and ordinal information of seven data training sets

	5		50		100		500		1000	
Project ε	CW	Rank	CW	Rank	CW	Rank	CW	Rank	CW	Rank
HK50	4.903	5	5.791	3	5.706	4	5.897	2	5.977	1
HSCIIG	4.308	5	4.948	4	5.051	3	5.312	2	5.358	1
HSCIT	1.082	5	0.955	3	0.929	5	0.947	4	0.961	2
HSCIU	1.573	1	1.519	4	1.421	5	1.527	3	1.535	2
SP500-A	2.911	1	2.270	2	2.217	5	2.222	4	2.234	3
SP500-B	0.998	1	0.888	5	0.915	2	0.904	3	0.900	4
NDX	2.209	1	2.764	4	2.766	3	2.943	2	3.000	1
Average order value	2.714		3.571		3.857		2.857		**2**	
Order value gap	2.915		1.764		1.892		1.146		**0.934**	

Note: The bold value represents the parameters with the highest average ranking in the training set.

Figure 3
Online learning adaptive values of α and epsilon under the SP500 (Group A) constituent stock data set



$$\hat{S}_n = \max_{\{\mathbf{b}\varepsilon\Delta_d\}_{t=1}^{n}} \prod_{i=1}^{n} (\mathbf{b}_t^T \mathbf{x}_t) \tag{28}$$

Annualized return (AR) reflects the compound effect of the actual returns of the algorithm. Actually, when converting the current yield into the annual yield under the consideration of compound interest, the larger index value will be better. Table 5 shows that the AR of the OAAT algorithm ranks first in the HSCIT, HSCIU, and SP500-B data set, second in the SP500-A, NDX data set, and third in the HSCIIG and HK50. It is found that the OAAT algorithm is capable of constructing investment portfolios with strong potential for returns. AR can be calculated as follows:

$$AR = S_n^{\frac{252}{n}-1} \tag{29}$$

Sharpe ratio (SR) quantifies the amount of additional return that an investment portfolio generates per unit of total risk taken. The higher the SR value, the better the performance of the financial asset in balancing returns and risks. Table 6 indicates that the OAAT algorithm is a feasible algorithm balancing both benefits and risks. The OAAT algorithm ranks first among the HSCIT, HSCIU, and SP500-B data set. In the remaining HK50, HSCIIG, SP500-A, and NDX, although OAAT algorithm does not occupy the first place, however, it is generally above average. SR can be calculated as follows:

$$SR = \frac{E(R_p) - R_f}{\sigma_p} \tag{30}$$

where $E(R_p)$ represents the expected yield for investors; R_f represents the risk-free yield; and σ_p represents the standard deviation of the portfolio.

Calmar ratio (CR) refers to the ratio of determinate return to diminishing risk in a hedge fund. CR represents the ratio of the AR to max drawdown. Table 7 indicates that the risk management of the OAAT algorithm is comparatively strong. Despite ranking third in the SP500-A data set and fourth in the HK50 data set with scores of 0.865 and 0.057, respectively, this algorithm outperforms others in various data sets. This implies that the OAAT algorithm possesses a substantial advantage across a majority of the data sets under consideration.

$$CR = \frac{AR}{MDD} \tag{31}$$

Figure 4 displays the CW trend chart, which is based on the experimental outcomes derived from seven portfolio data sets executed under distinct algorithms. Two significant observations can be made from Figure 4. Firstly, the OAAT algorithm witnesses a downturn during period 800 (early March 2020, the commencement phase of the pandemic), but its recovery is fast, and it attains a prominent position in the succeeding stages, signifying its capability to handle unforeseen events relative to other algorithms. Secondly, throughout the simulation process, the OAAT algorithm essentially maintained its position around the top three in terms of performance, with particular emphasis on its superiority in the later stages (e.g., period 1000–1500).

According to the trend of wealth accumulation shown in Figure 4 and index evaluation in Tables 4, 5, 6, and 7, OAAT is a robust and effective algorithm and outstanding algorithm of follow the winner. Figure 4 shows the trend of accumulated

Figure 4
The trend of accumulated wealth of portfolio data sets under different algorithms

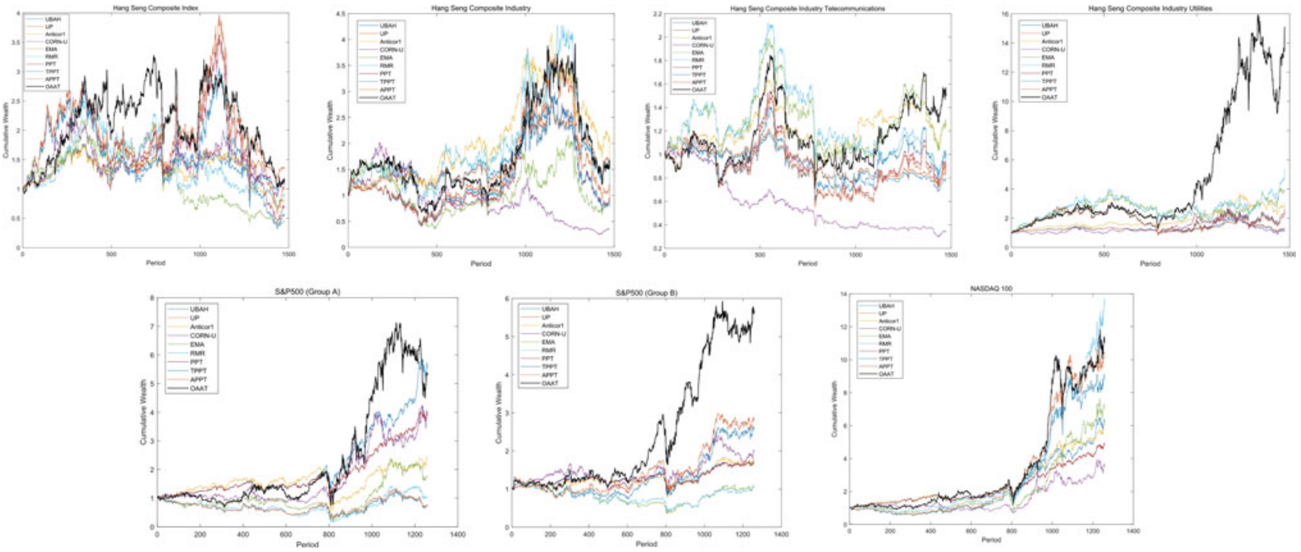


Table 4
Cumulative wealth of 10 OLPS algorithms in 7 portfolio data sets

Project	HK50	HSCIIG	HSCIT	HSCIU	SP500-A	SP500-B	NDX
UBAH	1.049	1.584	0.784	1.256	**5.530**	1.749	6.184
UP	1.285	1.662	0.839	1.328	3.887	1.682	4.780
Anticor-1	**1.633**	**2.110**	1.266	3.054	2.374	1.764	6.044
CORN-U	0.780	0.359	0.351	1.190	3.861	1.979	3.482
EMA	0.362	0.920	1.257	3.956	1.704	1.108	6.995
RMR	0.470	1.712	0.980	5.287	1.002	1.048	**13.290**
PPT	0.697	0.960	0.916	2.683	3.909	1.694	4.822
TPPT	0.555	0.918	0.999	2.401	0.707	2.586	8.731
APPT	1.017	1.218	0.875	2.474	0.727	2.824	10.012
OAAT	1.128	1.680	**1.520**	**15.110**	5.297	**5.626**	11.025

Note: The bold value represents the algorithm that has the most cumulative wealth growth in different data sets.

Table 5
Annualized returns of 10 OLPS algorithms in 7 portfolio data sets

Project	HK50	HSCIIG	HSCIT	HSCIU	SP500-A	SP500-B	NDX
UBAH	0.008	0.082	−0.041	0.040	**0.408**	0.118	0.440
UP	0.044	0.091	−0.030	0.050	0.312	0.110	0.368
Anticor-1	**0.087**	**0.136**	0.041	0.210	0.189	0.120	0.434
CORN-U	−0.042	−0.161	−0.164	0.030	0.311	0.146	0.284
EMA	−0.159	−0.014	0.040	0.265	0.113	0.021	0.476
RMR	−0.121	0.096	−0.004	0.329	0.000	0.010	**0.678**
PPT	−0.060	−0.007	−0.015	0.184	0.314	0.111	0.370
TPPT	−0.096	−0.014	0.000	0.162	−0.067	0.209	0.543
APPT	0.003	0.034	−0.023	0.167	−0.062	0.231	0.586
OAAT	0.021	0.093	**0.074**	**0.591**	0.396	**0.413**	0.617

Note: The bold value represents the algorithm that has the most cumulative wealth growth in different data sets.

Table 6
Sharpe ratios of 10 OLPS algorithms in 7 portfolio data sets

Project	HK50	HSCIIG	HSCIT	HSCIU	SP500-A	SP500-B	NDX
UBAH	−0.130	0.152	−0.548	−0.001	**1.284**	0.346	**1.377**
UP	0.017	0.196	−0.440	0.049	1.018	0.306	1.294
Anticor-1	**0.137**	**0.276**	0.006	0.593	0.334	0.274	1.213
CORN-U	−0.207	−0.498	−1.058	−0.026	0.618	0.321	0.557
EMA	−0.363	−0.104	0.000	0.483	0.124	−0.047	0.903
RMR	−0.304	0.109	−0.153	0.644	−0.069	−0.075	1.327
PPT	−0.174	−0.086	−0.185	0.291	1.024	0.312	1.304
TPPT	−0.238	−0.100	−0.136	0.246	−0.175	0.405	1.018
APPT	−0.065	−0.010	−0.211	0.258	−0.166	0.455	1.099
OAAT	−0.041	0.098	**0.112**	**1.187**	0.583	**0.882**	1.187

Note: The bold value represents the algorithm that has the most cumulative wealth growth in different data sets.

Table 7
Calmar ratios of 10 OLPS algorithms in 7 portfolio data sets

Project	HK50	HSCIIG	HSCIT	HSCIU	SP500-A	SP500-B	NDX
UBAH	0.016	0.142	−0.097	0.074	1.102	0.307	1.303
UP	0.105	0.169	−0.074	0.107	0.741	0.279	1.143
Anticor-1	**0.150**	0.231	0.073	0.509	0.253	0.280	1.218
CORN-U	−0.061	−0.184	−0.224	0.044	0.752	0.329	0.555
EMA	−0.173	−0.019	0.069	0.465	0.136	0.025	0.830
RMR	0.130	0.137	−0.007	0.600	−0.001	0.010	1.232
PPT	−0.165	−0.020	−0.066	0.574	**1.371**	0.517	1.825
TPPT	−0.264	−0.042	0.000	0.504	−0.146	0.645	1.628
APPT	0.008	0.101	−0.100	0.523	−0.135	0.711	1.756
OAAT	0.057	**0.272**	**0.330**	**2.135**	0.865	**1.271**	**2.015**

Note: The bold value represents the algorithm that has the most cumulative wealth growth in different data sets.

wealth of various portfolio data sets. Among 10 types of algorithm, the OAAT algorithm shows a relatively significant advantage in several test sets, maintaining its superiority until the end. The asset under OAAT algorithm shows a good growth trend, which indicates that OAAT algorithm is an effective and robust portfolio algorithm. Compared with other algorithms in the data set, OAAT algorithm is more effective than other algorithms in balancing the benefits and risks. Under the evaluation of accumulated wealth, AR, SR, and CR, although OAAT algorithm does not perform best in each data set, it has obvious advantages in multiple data sets, with an indication of that OAAT algorithm is an efficient and feasible OLPS algorithm with strong generalization ability.

5. Statistic Test

Based on the aforementioned empirical comparative analysis, it is apparent that the OAAT algorithm outperforms certain OLPS algorithms in terms of performance. However, it does not provide conclusive evidence to establish the OAAT algorithm as the ultimate best performer across various data sets and algorithms. Hence, we will now evaluate the overall performance of the OAAT algorithm, selecting two indicators, CW and SR, to assess its profit-generating ability and risk aversion through a non-parametric test.

5.1. Nonparametric tests of CW

Nonparametric statistical tests were conducted on 10 OLPS algorithms under 7 portfolio data sets to obtain the average order value of accumulated wealth and the Friedman statistic, where $N = 7$, $k = 10$, $\chi^2 = 18.83$, and the Friedman statistic at the level of significance α =0.05 is F_f=2.56~F(9,54). Due to the Friedman statistic is greater than the critical value of the F-distribution 2.06, the assumption that these nine OLPS algorithms have similar performance as the OAAT algorithm is rejected. After that, we employ the Denferroni–Dunn test to evaluate and compare the performance of the OAAT algorithm with the other nine OLPS algorithms comprehensively. Additionally, it determines the critical value range of the difference between the average order value:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{32}$$

In the Denferroni–Dunn test, it is assumed that if the disparity between the mean rankings of any two algorithms surpasses the critical threshold (CD), it is considered that one algorithm is superior to the other under the corresponding confidence level. At the confidence level of 0.10: CD $=2.529\sqrt{\frac{10\times11}{6\times7}}$, comparing the first-order values of various algorithms, the difference between the OAAT algorithm and the other three OLPS algorithms (UP, CORN-U, EMA, PPT, TPPT) is higher than the critical value of 4.09, which shows that the performance of these three OLPS algorithms is worse than that of the OAAT algorithm. Although the remaining algorithms do not meet the alternative hypothesis of Denferroni–Dunn test, the average performance of OAAT algorithm ranks first and performs best in multiple data sets. Therefore, the OAAT algorithm remains the best algorithm among other OLPS algorithms in terms of comprehensive performance, as shown in Table 8.

Table 8
Nonparametric tests of cumulative wealth

Project	HK50	HSCIIG	HSCIT	HSCIU	SP500-A	SP500-B	NDX	Avg rank	Diff
UBAH	4	5	9	9	**1**	6	6	5.71	3.86
UP	2	4	8	8	4	8	9	6.14	4.29
Anticor-1	**1**	**1**	2	4	6	5	7	3.71	1.86
CORN-U	6	10	10	10	5	4	10	7.86	6.00
EMA	10	8	3	3	7	9	5	6.43	4.57
RMR	9	2	5	2	8	10	**1**	5.29	3.43
PPT	7	7	6	5	3	7	8	6.14	4.29
TPPT	8	9	4	7	10	3	4	6.43	4.57
APPT	5	6	7	6	9	2	3	5.43	3.57
OAAT	3	3	**1**	**1**	2	**1**	2	**1.86**	**0.00**

Note: The bold value represents the algorithm that has the most cumulative wealth growth in different data sets.

Table 9
Nonparametric tests of Calmar ratio

Project	HK50	HSCIIG	HSCIT	HSCIU	SP500-A	SP500-B	NDX	Avg rank	Diff
UBAH	5	4	8	9	2	6	5	5.57	3.86
UP	3	3	7	8	5	8	8	6.00	4.29
Anticor-1	1	2	2	5	6	7	7	4.29	2.57
CORN-U	7	10	10	10	4	5	10	8.00	6.29
EMA	9	7	3	7	7	9	9	7.29	5.57
RMR	2	5	5	2	8	10	6	5.43	3.71
PPT	8	8	6	3	1	4	2	4.57	2.86
TPPT	10	9	4	6	10	3	4	6.57	4.86
APPT	6	6	9	4	9	2	3	5.57	3.86
OAAT	4	**1**	**1**	**1**	3	**1**	**1**	**1.71**	**0.00**

Note: The bold value represents the algorithm that has the most cumulative wealth growth in different data sets.

5.2. Nonparametric tests of the CR

Similar to the previous test, 10 algorithms in seven data sets are used to solve the average order value of CR in Friedman statistics, $N = 7, k = 10, \chi^2 = 21.02, F_f = 3.003 \sim F(9,54)$. Therefore, the Friedman statistic, which is equal to 3.003, is greater than the critical value of 2.138, so the original assumption that all of the 10 OLPS algorithms perform equally has been disproven. It can be clearly seen that the difference between OAAT algorithm and four OLPS algorithms (UP, CORN-U, EMA, TPPT) is higher than the critical value of 4.09, which shows that the performance of these OLPS algorithms is slightly inferior to that of OAAT algorithm. In the seven data sets, the OAAT algorithm stands out from the other nine algorithms by securing the highest number of first-place rankings and achieving the top average ranking across all data sets. Table 9 shows that the OAAT algorithm is proficient in effectively balancing income and risk, demonstrating exceptional comprehensive abilities. It is a trustworthy and highly generalizable OLPS algorithm.

6. Conclusion

Aiming at the problem of how to use asset historical information to mine portfolio potential returns and parameter settings in OLPS, we propose a novel online adaptive asset tracking algorithm (OAAT) for OLPS. The OAAT algorithm updates investment proportions by considering various factors such as recent peak prices, historical returns, and historical volatility. It optimizes parameters through online learning, starting with initial parameters based on the minimum sum principle of ordinal information. After each trading phase, parameter optimization is performed using the gradient descent algorithm, and the average values of the optimal parameters from the last five days are used for the next phase. The results of the empirical analysis and statistical tests indicate that the OAAT algorithm effectively determines the investment proportion to balance return and risk, demonstrating superior performance compared to the other nine OLPS algorithms. However, there are more optimization algorithms that can be tried in this experiment. Therefore, further research is needed on the OAAT algorithm.

Funding Support

This work was supported by the National Social Science Foundation of China (18BTJ029), Key Projects of National Statistical Science Research Projects (2020LZ10), and Tertiary Education Scientific Research Project of Guangzhou Municipal Education Bureau (202235324).

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicabe to this article as no new data were created or analyzed in this study.

References

Bertsekas, D. P. (1997). Nonlinear programming. *Journal of the Operational Research Society*, *48*(3), 334–334. https://doi.org/10.1057/palgrave.jors.2600425

Borodin, A., El-Yaniv, R., & Gogan, V. (2004). Can we learn to beat the best stock. *Journal of Artificial Intelligence Research*, *21*, 579–594. https://doi.org/10.1613/jair.1336

Brahma, P. P., Wu, D., & She, Y. (2016). Why deep learning works: A manifold disentanglement perspective. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(10), 1997–2008. https://doi.org/10.1109/TNNLS.2015.2496947

Cover, T. M. (1991). Universal portfolios. *Mathematical Finance*, *1*(1), 1–29. https://doi.org/10.1111/j.1467-9965.1991.tb00002.x

Dai, H. L., Liang, C. X., Dai, H. M., Huang, C. Y., & Adnan, R. M. (2022). An online portfolio algorithm based on trend promote price tracing ensemble learning algorithm. *Knowledge-Based Systems*, *239*, 107957. https://doi.org/10.1016/j.knosys.2021.107957

Dai, H. L., Huang, C. Y., Dai, H. M., Lai, F. T., Lv, X. T., & Adnan, R. M. (2023). A novel adjusted learning algorithm for online portfolio selection using peak price tracking approach. *Decision Analytics Journal*, *7*, 100256. https://doi.org/10.1016/j.dajour.2023.100256

Delgado, M. A., & Song, X. (2018). Nonparametric tests for conditional symmetry. *Journal of Econometrics*, *206*(2), 447–471. https://doi.org/10.1016/j.jeconom.2018.06.010

Dong, Z., & Zhou, Y. (2019). Momentum effect timing strategy and investment value – An empirical study based on agricultural stock market. *Agricultural Economics and Management*, *2019*(03), 5–18.

Esfahani, H. N., Sobhiyah, M. H., & Yousefi, V. R. (2016). Project portfolio selection via harmony search algorithm and modern portfolio theory. *Procedia-Social and Behavioral Sciences*, *226,* 51–58. https://doi.org/10.1016/j.sbspro.2016.06.161

Frost, J., Gambacorta, L., & Gambacorta, R. (2020). The Matthew effect and modern finance: On the nexus between wealth inequality, financial development and financial technology. *SSRN*. https://doi.org/10.2139/ssrn.3659973

Györfi, L., Lugosi, G., & Udina, F. (2006). Nonparametric kernel–based sequential investment algorithms. *Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics*, *16*(2), 337–357. https://doi.org/10.1111/j.1467-9965.2006.00274.x

Huang, D., Zhou, J., Li, B., Hoi, S. C., & Zhou, S. (2016). Robust median reversion algorithm for online portfolio selection. *IEEE Transactions on Knowledge and Data Engineering*, *28*(9), 2480–2493. https://doi.org/10.1109/TKDE.2016.2563433

Lai, Z. R., Dai, D. Q., Ren, C. X., & Huang, K. K. (2018). A peak price tracking-based learning system for portfolio selection. *IEEE Transactions on Neural Networks and Learning Systems*, *29* (7), 2823–2832. https://doi.org/10.1109/TNNLS.2017.2705658

Li, B., & Hoi, S. C. H. (2018). *Online portfolio selection: Principles and algorithms*. USA: CRC Press.

Li, B., Sahoo, D., & Hoi, S. C. (2016). OLPS: A toolbox for on-line portfolio selection. *Journal of Machine Learning Research*, *17*(35), 1–5.

Li, Y., Chen, C., Zheng, X., Wang, Y., & Gong, B. (2023). REFER: Randomized online factor selection framework for portfolio management. *Expert Systems with Applications*, *223*, 119837. https://doi.org/10.1016/j.eswa.2023.119837

Lin, C. M., Yang, M. S., Chao, F., Hu, X. M., & Zhang, J. (2016). Adaptive filter design using type-2 fuzzy cerebellar model articulation controller. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(10), 2084–2094. https://doi.org/10.1109/TNNLS.2015.2491305

Ma, Y., Han, R., & Wang, W. (2021). Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, *165*, 113973. https://doi.org/10.1016/j.eswa.2020.113973

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77–91. https://doi.org/10.2307/2975974

Moghaddam, A. H., Moghaddam, M. H., & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science*, *21*(41), 89–93. https://doi.org/10.1016/j.jefas.2016.07.002

Ortobelli Lozza, S., Carneiro Mendes, L., & Nardelli, C. (2022). Online portfolio selection models versus mean variance optimal choices. In *Managing and Modelling of Financial Risks.*

Raitoharju, J., Kiranyaz, S., & Gabbouj, M. (2016). Training radial basis function neural networks for classification via class-specific clustering. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(12), 2458–2471. https://doi.org/10.1109/TNNLS.2015.2497286

Salas, A. (2020). *Bayesian online learning for online portfolio selection*. PhD Thesis, University of Oxford.

Shiller, R. J. (2003). From efficient markets theory to behavioral finance. *Journal of Economic Perspectives*, *17*(1), 83–104. https://doi.org/10.1257/089533003321164967

Singh, V., & Thurman, A. (2019). How many ways can we define online learning? A systematic literature review of definitions of online learning (1988–2018). *American Journal of Distance Education*, *33*(4), 289–306. https://doi.org/10.1080/08923647.2019.1663082

Soleymani, F., & Paquet, E. (2021). Deep graph convolutional reinforcement learning for financial portfolio management–Deeppocket. *Expert Systems with Applications*, *182*, 115127. https://doi.org/10.1016/j.eswa.2021.115127

Stella, F., & Ventura, A. (2011). Defensive online portfolio selection. *International Journal of Financial Markets and Derivatives*, *2*(1–2), 88–105.

Tantisripreecha, T., & Soonthomphisaj, N. (2018). Stock market movement prediction using LDA-online learning model. In *2018 19th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing,* 135–139. https://doi.org/10.1109/SNPD.2018.8441038

Tiwari, R., Srivastava, S., & Gera, R. (2020). Investigation of artificial intelligence techniques in finance and marketing. *Procedia Computer Science*, *173*, 149–157. https://doi.org/10.1016/j.procs.2020.06.019

Yao, X., & Zhang, N. (2023). Elastic-net regularized online portfolio selection with transaction costs. In *2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms,* 1495–1502. https://doi.org/10.1109/EEBDA56825.2023.10090727

Wu, W., Zhu, Y., & Huang, D. (2019). Semi-exponential gradient algorithm and empirical analysis for online portfolio selection. *Journal of Computer Applications*, *39*(8), 2462–2467. https://doi.org/10.11772/j.issn.1001-9081.2018122588

Zhang, W., Li, X., Chen, Y., Ye, N., & Zhang, X. P. (2023). Effective online portfolio selection for the long-short market using mirror gradient descent. *IEEE Signal Processing Letters*, *30*, 913–917. https://doi.org/10.1109/LSP.2023.3296913

Zhou, T. (2018). A method for portfolio selection based on joint probability of co-movement of multi-assets. *Journal of Mathematical Finance*, *8*(3), 535–548. https://doi.org/10.4236/jmf.2018.83034

Zimmert, J., Agarwal, N., & Kale, S. (2022). Pushing the efficiency-regret Pareto frontier for online learning of portfolios and quantum states. In *Proceedings of Thirty Fifth Conference on Learning Theory, 178*, 182–226.

How to Cite: Lai, F., Liang, C., Huang, C., & Dai, H. (2023). Online Adaptive Asset Tracking Algorithm with Ordinal Information. *Journal of Computational and Cognitive Engineering*. https://doi.org/10.47852/bonviewJCCE32021590