Scientific Research Publishing

Applied Mathematics, 2024, 15, 313-330
https://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

A Portfolio Selection Method Based on Pattern Matching with Dual Information of Direction and Distance

Xinyi He

Department of Mathematics, Jinan University, Guangzhou, China
Email: pennyhxy@stu2021.jnu.edu.cn

How to cite this paper: He, X.Y. (2024) A Portfolio Selection Method Based on Pattern Matching with Dual Information of Direction and Distance. *Applied Mathematics*, **15**, 313-330.
https://doi.org/10.4236/am.2024.155019

Received: March 29, 2024
Accepted: May 4, 2024
Published: May 7, 2024

Abstract

Pattern matching method is one of the classic classifications of existing online portfolio selection strategies. This article aims to study the key aspects of this method—measurement of similarity and selection of similarity sets, and proposes a Portfolio Selection Method based on Pattern Matching with Dual Information of Direction and Distance (PMDI). By studying different combination methods of indicators such as Euclidean distance, Chebyshev distance, and correlation coefficient, important information such as direction and distance in stock historical price information is extracted, thereby filtering out the similarity set required for pattern matching based investment portfolio selection algorithms. A large number of experiments conducted on two datasets of real stock markets have shown that PMDI outperforms other algorithms in balancing income and risk. Therefore, it is suitable for the financial environment in the real world.

Keywords

Online Portfolio Selection, Pattern Matching, Similarity Measurement

1. Introduction

In the context of global financial markets, how to allocate limited capital reasonably and maximize returns has always been a focus of attention for investors and researchers. The complexity and high interdependence of financial markets make it difficult to predict market behavior, while factors such as financial innovation, trading methods, and market regulation also increase market uncertainty. Therefore, how to complete the allocation of capital in such a complex system has become a global challenge.

Portfolio Selection (PS) is a core task in the financial field, which involves allocating capital within a given set of assets to achieve specific investment goals. The origin of this problem can be traced back to Markowitz's mean variance theory proposed in 1952 [1], which is based on mean variance analysis and selects the optimal investment portfolio by balancing expected returns and risks. However, with the continuous changes in the market and the diversification of investor demands, traditional portfolio selection methods have gradually exposed their limitations.

In recent years, with the rapid development of artificial intelligence and machine learning technologies, these technologies have provided new ideas and methods for solving portfolio selection problems. Machine learning algorithms have the ability to analyze historical data and identify patterns and trends within it, which enables them to effectively predict future market behavior. By learning and processing a large amount of historical data, machine learning models can capture complex relationships between data and make accurate predictions based on them. This process not only improves the accuracy of predictions, but also provides valuable reference information for investors and decision-makers, helping them make wiser decisions.

By combining machine learning algorithms with portfolio theory, researchers have proposed a series of high-performance online portfolio selection strategies [2] [3]. The difference between online portfolio selection and traditional portfolio selection is that it considers the dynamics and real-time nature of the market. In an online environment, investors need to continuously adjust their investment portfolios in a constantly changing market to adapt to market changes and maximize returns. This requires investors to have the ability to analyze real-time data and make quick decisions. In order to solve the problem of online portfolio selection, researchers use various machine learning algorithms to process large amounts of real-time data and make fast and accurate investment decisions. Through the application of these algorithms, investors can maintain sharp insight in the constantly changing market environment, adjust investment strategies in a timely manner, and thus obtain better investment returns.

In summary, online portfolio selection is a challenging and promising research field. By combining artificial intelligence and machine learning technologies, we can develop more intelligent and effective investment strategies to help investors achieve better returns in complex and ever-changing financial markets. Under the previous research, the methods to solve the portfolio selection problem are mainly divided into following the winner, following the loser, pattern matching and so on. In this study, we use one of the highly intuitive methods: pattern matching. In short, pattern matching is to find the most similar window in historical stock data, predict the stock price of the latest trading day, and optimize the portfolio. Therefore, this paper makes efforts on the selection method of similar sets and proposes a Portfolio Selection Method based on Pattern Matching with Dual Information of Direction and Distance (PMDI).

The remainder of the paper is structured as follows. Section 2 presents some

preliminary work. Section 3 mainly introduces several online portfolio selection strategies based on pattern matching. Section 4 introduces the basic idea and algorithm of PMDI. Section 5 conducts experiments to evaluate the algorithm. Finally, Section 6 concludes.

2. Preliminaries

Suppose there is a financial market with M assets, over which we will invest n trading periods. The non-negative *price relative vectors* $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathbb{R}_+^M$ denotes the changes of asset prices for n trading periods, where $\mathbb{R}_+^M$ denotes the M-dimensional non-negative real space. The t^{th} vector is $\mathbf{x}_t = \left(x_{(t,1)}, x_{(t,2)}, \cdots, x_{(t,M)}\right)$; the i^{th} component of it $x_{(t,i)}$ can be expressed as: $x_{(t,i)} = \dfrac{P_{(t,i)}}{P_{(t,i-1)}}$, that is, the ratio of the i^{th} asset's closing price of period t to that of period $t-1$. Thus, there are $\mathbf{P}_t = \left(P_{(t,1)}, \cdots, P_{(t,M)}\right)$. Notably, window size w is given in some algorithms; at that time, the *market window* for the t^{th} trading period is defined as $\mathbf{X}_{t-w}^{t-1} = \left(\mathbf{x}_{t-w}, \cdots, \mathbf{x}_{t-1}\right)$.

Meanwhile, the portfolio at the beginning of period t can be expressed as $\boldsymbol{b}_t \in \Delta_M$, where Δ_M denotes the M-dimensional simplex. Assuming self-financed and non-short-selling, the components of $\mathbf{b}_t$ meet the conditions of $b_{(t,i)} \geq 0$ and $\sum_{i=1}^{M} b_{(t,i)} = 1$.

Hence, the cumulative wealth $\mathbf{S}_t$ at the end of the t^{th} trading period is updated by an increasing factor $\mathbf{b}_t^{\mathrm{T}}\mathbf{x}_t$: that is: $\mathbf{S}_t = \mathbf{S}_{t-1} \cdot \left(\mathbf{b}_t^{\mathrm{T}}\mathbf{x}_t\right)$. For convenience, suppose the initial wealth is $\mathbf{S}_0$ and $\mathbf{S}_0 = \$1$; thus, the total wealth eventually achieved is:

$$\mathbf{S}_n = \mathbf{S}_0 \prod_{t=1}^{n} \left(\mathbf{b}_t^{\mathrm{T}}\mathbf{x}_t\right) = \prod_{t=1}^{n} \left(\mathbf{b}_t^{\mathrm{T}}\mathbf{x}_t\right). \tag{1}$$

Finally, the online portfolio selection problem has become a sequential decision problem through previous setting and modeling, aiming to maximize $\mathbf{b}_t^{\mathrm{T}}\mathbf{x}_t$ in each period to maximize the final accumulated wealth.

Notably, the above mathematical model is based on the following general assumptions:

- No transaction cost;
- Liquid market: In this market, assets are easily convertible without apparent price fluctuations, and the decline in the value of assets is minimal. There are always both buyers and sellers on the market. They can buy and sell any quantity of assets at the trading period's closing price;
- No market impact cost.

The method based on pattern matching entails comparing the price relationships of the previous windows to yield a similarity set C:

$$C\left(\mathbf{X}_1^t, w\right) = \left\{w < i < t+1 : G\left(\mathbf{X}_{t-w+1}^t\right) = G\left(\mathbf{X}_{i-w}^{i-1}\right)\right\}, \tag{2}$$

where G is a corresponding discretization function.

Given a similar set $C\left(\mathbf{X}_1^t\right)$, there are price relative vectors $\mathbf{x}_i, i \in C\left(\mathbf{X}_1^t\right)$, and

the logarithmic optimal utility function is the weighted average logarithmic return of its probability P_i:

$$U_L\left(\mathbf{b}, C\left(\mathbf{X}_1^t\right)\right) = E\left(\log \mathbf{b}\cdot\mathbf{x}\right) = \sum_{i\in C\left(\mathbf{X}_1^t\right)} P_i \log \mathbf{b}\cdot\mathbf{x}_i \tag{3}$$

Therefore, the innovation points of existing pattern-matching-based methods focus primarily on modifying the similarity set's measurement method to improve performance and mathematical interpretation. For example, according to the convention, we generally choose the uniform portfolio if the similarity set is empty in the following algorithms.

3. Pattern-Matching Based Approaches

This section introduces benchmark strategy (which will be used for comparative experiments in subsequent sections) and pattern-matching strategy.

3.1. Benchmarks

- BAH. Buy-and-Hold strategy, in which assets are bought and held until the end of the period with the initial weight $\mathbf{b}$, and the warehouse is not adjusted during the trading period;
- Market. A uniform BAH strategy with initial weight $\mathbf{b} = \left(\frac{1}{M}, \frac{1}{M}, \cdots, \frac{1}{M}\right)$;
- Best-stock. It is also a BAH strategy, a hindsight strategy that invests all the money into assets that perform best in hindsight;
- CRP. Constant Rebalanced Portfolios (CRP) maintain the portfolio's initial weight through constant warehouse adjustments, as the value of each asset fluctuates over time;
- UCRP. Uniform Constant Rebalanced Portfolios (UCRP) rebalance to the same uniform weight in each trading period;
- BCRP. It is also a CRP strategy, a hindsight strategy with potfolio $\mathbf{b}^*$ to maximize the final wealth.

3.2. Method Based on Pattern Matching

3.2.1. B^K

Györf *et al.* proposed Nonparametric *kernel-based* log-optimal strategy (B^K) [4] in 2006, which combines kernel-based sample selection with a log-optimal utility function. This strategy obtains a similar pair of market windows by calculating the L2 norm, also known as the Euclidean distance:

$$C_K\left(\mathbf{X}_1^t, w\right) = \left\{w < i < t+1 : \left\|\mathbf{X}_{t-w+1}^t - \mathbf{X}_{i-w}^{i-1}\right\| \le \frac{c}{\ell}\right\}, \tag{4}$$

where c and ℓ are the threshold parameters used to control the number of similar samples.

3.2.2. B^{NN}

Györf *et al.* proposed Nonparametric *nearest neighbor-based* empirical portfolio

selection strategy (B^{NN}) [5] in 2008, which measures the similarity of two market windows using the ℓ nearest neighbor [6] in terms of Euclidean distance:

$$C_N\left(\mathbf{X}_1^t, w\right) = \left\{w < i < t+1 : \mathbf{X}_{i-w}^{i-1} \text{ is among the } \ell\text{NNs of } \mathbf{X}_{t-w+1}^t\right\}, \tag{5}$$

where ℓ is a threshold parameter.

3.2.3. CORN

Li *et al.* proposed *Correlation-driven* Nonparametric Learning Approach for Portfolio Selection [7] in 2011, which uses the correlation coefficient to measure the similarity between two market windows:

$$C_C\left(\mathbf{X}_1^t, w\right) = \left\{w < i < t+1 : \frac{cov\left(\mathbf{X}_{i-w}^{i-1}, \mathbf{X}_{t-w+1}^t\right)}{std\left(\mathbf{X}_{i-w}^{i-1}\right) std\left(\mathbf{X}_{t-w+1}^t\right)} \geq \rho\right\}, \tag{6}$$

where ρ is a correlation coefficient threshold.

Generally, CORN involves two steps. The first is defining experts $\left\{\varepsilon(w,\rho) : w \geq 1, -1 \leq \rho \leq 1\right\}$, where w is the window size and ρ is the correlation coefficient threshold. The duty of each expert is to identify comparable historical price relationships and learn how to construct an optimal portfolio based on these comparable historical price relationships. The second step combines the expert-calculated portfolios to create the final portfolio effectively. The final portfolio for the t^{th} trading day can be calculated as follows:

$$\mathbf{b}_t = \frac{\sum_{w,\rho} q(w,\rho) s_{t-1}(w,\rho) \mathbf{b}_t(w,\rho)}{\sum_{w,\rho} q(w,\rho) s_{t-1}(w,\rho)}, \tag{7}$$

where $s_{t-1}(w,\rho)$ is the historical performance of each expert, $q(w,\rho)$ is a probability distribution function, and $\mathbf{b}_t(w,\rho)$ is the portfolio output by each expert $\varepsilon(w,\rho)$.

In addition, they present CORN-U and CORN-K variants. The CORN-U algorithm treats $q(w,\rho)$ as a uniform distribution; that is, $q(w,\rho) = \frac{1}{W}$, where W is the maximum number of windows that combine all experts uniformly. The second algorithm, CORN-K, assigns a uniform distribution of $q(w,\rho) = \frac{1}{K}$ to the set of top-K best experts, while the weights of the remaining experts are set to 0.

4. A Portfolio Selection Method Based on Pattern Matching with Dual Information of Direction and Distance

When investors make investments, it is difficult to invest in thousands of stocks at once. Usually, they need to choose a few or dozens of stocks (risk assets) from thousands of stocks and invest proportionally. Therefore, the key to portfolio problems lies in how to scientifically and proportionally allocate investment funds to multiple assets, in order to achieve maximum returns and minimum risks. In this chapter, we propose a new strategy that belongs to the pattern

matching type of investment portfolio selection method. By finding a series of time periods with w as the cycle and the most similar to the recent w period stock price changes on each trading day, we obtain a similarity set of stocks, apply optimization algorithms to it, obtain the required investment portfolio for the next trading day, and extract effective information such as direction and distance from the historical stock prices. This strategy is named: A Portfolio Selection Method based on Pattern Matching with Dual Information of Direction and Distance (PMDI).

4.1. Basic Idea

Early non-parametric learning methods, such as B^K and B^{NN}, essentially used Euclidean distance to measure the similarity between current and historical market windows. However, the main drawback of using Euclidean distance is that it does not take into account the direction information of market window movement, which may include some useless or harmful relative prices. Therefore, CORN uses the Pearson product moment correlation coefficient to measure it. In fact, the Pearson correlation coefficient is the cosine value of the vector angle between two standardized sets of data, mainly used to measure the direction of information between the data.

However, the CORN strategy is not perfect. The historical similarity set defined by it covers all market vectors whose correlation coefficient with the current market vector is not lower than the preset threshold c. If the threshold value c is improperly selected, whether it is too high or too low, it may cause those vectors that are significantly different from the current market vector to be wrongly included in the historical similar set, thus greatly reducing the effectiveness of the corn strategy.

In order to further optimize the previous algorithm, PMDI recommends a more complex selection of similar sets to achieve better results. Therefore, this paper studies the Euclidean distance, Chebyshev Distance, Correlation Coefficient and other distance measurement methods as well as K-Nearest Neighbors (KNN) algorithm [6]:

1) Euclidean Distance

Euclidean distance is a widely used and easy to understand distance calculation method. Firstly, assuming the existence of two n-dimensional variables $X=(x_1,\cdots,x_n)$ and $Y=(y_1,\cdots,y_n)$, the Euclidean distance can be expressed as follows:

$$D_{M2}=\sqrt{\sum_{i=1}^{n}|x_i-y_i|}. \tag{8}$$

In an investment portfolio, Euclidean distance is sensitive to changes in each stock because it calculates linear distance, so small changes in each dimension will affect the final distance. If the correlation between stock prices is high, using Euclidean distance may be more appropriate.

In reality, changes in stock prices are both related to the overall market situa-

tion in the financial market (such as the impact of the epidemic on the entire financial market). Therefore, when doing similarity combinations, we need to take into account the Euclidean distance.

2) Chebyshev Distance

The Chebyshev distance between two points $A(x_1, y_1)$ and $B(x_2, y_2)$ in a two-dimensional plane:

$$D_{AB} = \max\left(\left|x_2 - x_1\right|, \left|y_2 - y_1\right|\right). \tag{9}$$

So, the Chebyshev distance between two n-dimensional vectors $X = (x_1, \cdots, x_n)$ and $Y = (y_1, \cdots, y_n)$ is:

$$D_{M3} = \max_i \left|x_i - y_i\right|, \tag{10}$$

Specifically, Chebyshev distance, also known as chessboard distance, measures the maximum numerical difference between two points in various coordinate dimensions.

In investment portfolios, paying attention to the Chebyshev distance between relative price vectors at different time points can help investors understand the changes in stock prices across various dimensions. If the Chebyshev distance is large, it indicates that the stock price of a certain stock fluctuates greatly in the relative price vector, which may indicate that the stock has significant risks or opportunities. On the contrary, if the Chebyshev distance is small, it indicates that the price changes of each stock are relatively stable.

3) Correlation Coefficient

Pearson correlation coefficient is a statistical indicator used to measure the degree of linear correlation between two samples. Its value range is $[-1,1]$, where -1 represents complete negative correlation, 0 represents uncorrelated, and 1 represents complete positive correlation. The Pearson correlation coefficient calculation formula for samples X and Y is as follows:

$$P_{XY} = \frac{E(XY) - E(X)(Y)}{\sqrt{E\left(X^2\right) - E^2(X)} \cdot \sqrt{E\left(Y^2\right) - E^2(Y)}}, \tag{11}$$

where $E(X)$ represents the expected value of variable X, and the others represent the same.

In an investment portfolio, the Pearson correlation coefficient measures the degree of linear dependence between two relative price vectors.

If the Pearson correlation coefficient is close to 1, it indicates a strong positive linear relationship between the price vectors at two time points. That is, as the price vector at one time point increases, the price vector at the other time point also tends to increase. If the Pearson correlation coefficient is close to -1, it indicates a strong negative linear relationship between the price vectors at two time points, that is, as the price vector at one time point increases, the price vector at the other time point tends to decrease. If the Pearson correlation coefficient is close to 0, it indicates that there is no significant linear relationship between the price vectors at two time points.

Therefore, when making similarity combinations, we will also consider the Pearson correlation coefficient, and we are more inclined to choose relative price vectors with positive Pearson correlation coefficients.

4) K-Nearest Neighbors Algorithm

Strictly speaking, KNN is not a measure of distance, but as a classification algorithm in machine learning, it is very suitable for selecting similar sets in pattern matching.

Its principle is that when it is necessary to predict the category of a new sample, the algorithm will search for the K nearest sample points in the training dataset to the new sample, and determine the category of the new sample based on these K nearest neighbor categories. The "K" here is a key parameter that determines the number of neighboring samples participating in decision-making. In the KNN algorithm, Euclidean distance or Manhattan distance are usually used as distance metrics to calculate the similarity or distance between sample points.

The following Figure 1 is a very clear explanation: assuming that the small dashed circle and the large dashed circle in the figure represent K = 3 and K = 5, respectively, the red point in the center of the figure is the point we want to predict. So the KNN algorithm will find the three closest points to it and see which category has more. When K = 3, there are 2 blue crosses and 1 green star around the predicted point, and the predicted point should belong to the "blue cross" category; When K = 9, there are 4 blue crosses and 5 green stars around the predicted point, so the predicted point belongs to the "green star" category.

In fact, when using this algorithm for classification in class domains with small sample sizes, it is easy to generate misclassification. From the diagram, it can also be seen that the above figure did not provide a good classification result for the predicted points. On the contrary, this algorithm is more suitable for automatic classification of class domains with larger sample sizes, and is therefore very suitable as a classification method for pattern matching in portfolio selection, used to process a large amount of historical stock data.

In pattern matching, assuming we have a series of historical relative price

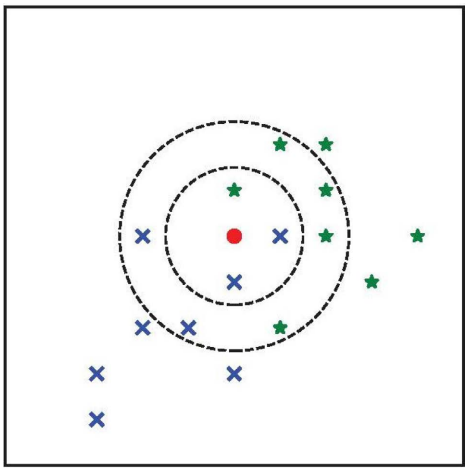

Figure 1. KNN algorithm.

vectors, we can use the KNN algorithm to predict the categories or trends of future price vectors. In this process, the KNN distance refers to the distance between the current price vector and the K nearest neighbor points in the historical dataset. If the current price vector is very close to the price vector of a certain period in history (*i.e.*, the KNN distance is small), it may indicate that the market is repeating past behavior patterns, so trend analysis can be conducted based on this.

Based on the above analysis, this article ultimately chooses to combine various ranging methods in different ways to form a historical similarity set with the highest similarity in historical moments.

In this algorithm, we mainly use the Correlation Coefficient and Chebyshev Distance as the distance measures. First, we filter out the historical stock prices that are close to the direction information provided in the current trading period and the distance information between individual stocks. Then, we use the KNN algorithm in terms of Euclidean distance as the measure to filter the similar sets twice, so as to extract the distance information, direction information and the distance information between individual stocks in the historical stock prices.

First, at the beginning of the T trading day, the parameter combination (W, m, ρ) is used to determine a relatively large similar set C_1, which meets the Chebyshev Distance limit or the Correlation Coefficient limit $\mathbf{X}_{i-w}^{i-1}$ will be selected into the similar set:

$$C_1 = \left\{ w < i < t-1 : \left\| \mathbf{X}_{t-w}^{t-1} - \mathbf{X}_{i-w}^{i-1} \right\|_\infty \le m \text{ or } \frac{cov\left(\mathbf{X}_{i-w}^{i-1}, \mathbf{X}_{t-w}^{t-1}\right)}{std\left(\mathbf{X}_{i-w}^{i-1}\right) std\left(\mathbf{X}_{t-w}^{t-1}\right)} \ge \rho \right\} \quad (12)$$

where w is the market window size, $m \ge 0$ refers to the Chebyshev Distance threshold, $-1 \le \rho \le 1$ is a parameter of Correlation Coefficient threshold, $cov(A, B)$ denotes the covariance between market windows A and B, and $std(A)$ denotes the standard deviation of market window A. In addition, $\|\cdot\|_\infty$ represents calculating the L_∞ norm, which is the Chebyshev Distance.

Then, ℓ historical relative price sequences closer to $\mathbf{X}_{t-w}^{t-1}$ are selected by KNN Algorithm in the similarity set to obtain the historical similarity set C_2. The specific method is: after calculating the Euclidean distance between the market vectors $\mathbf{X}_{t-w}^{t-1}$ and the $\mathbf{X}_{i-w}^{i-1}$ of C_1, arrange these distances in ascending order, and take the market vector corresponding to the first ℓ distances to construct the historical similarity set C_2:

$$C_2 = \left\{ i \in C_1 : \mathbf{X}_{i-w}^{i-1} \text{ is among the } \ell\text{NNs of } \mathbf{X}_{t-w}^{t-1} \right\}. \quad (13)$$

Therefore, the algorithm process of PMDI is shown in Algorithm 1.

Specifically, in the actual operation of the algorithm, ℓ cannot be fixed, because with the increase of trading period, many market vectors with high similarity to the current market vector will be excluded. Therefore, we need to construct a parameter ℓ about the number of neighbors that increases with the increase of trading period: $\ell = s_{ell} \times t$, Where s_{ell} is the scale factor.

From an experimental perspective, expert learning algorithms are very slow,

so PMDI will not use them, but instead choose fixed parameters through a large number of experiments. Specifically, if there is no historical price sequence that satisfies the similarity condition, the final calculated similarity set is an empty set: $C = \varnothing$, then a uniform investment portfolio $\mathbf{b}_t = \left(\frac{1}{M}, \frac{1}{M}, \cdots, \frac{1}{M}\right)$.

4.2. Portfolio Optimization

The first two sections proposed a method for selecting similar sets in pattern matching. Then, after calculating the similar set C_2 on each trading day, the optimal investment portfolio is obtained by maximizing the total return of all relative price vectors in that similar set. The overall process of portfolio optimization is shown in the Algorithm 2.

By using this algorithm to output $\mathbf{b}_t$ on each trading day, we will use it as the investment portfolio to complete the next trading day. By using this recursion, the final investment return can be calculated:

Algorithm 1: Algorithm of PMDI

Data: $\mathbf{X}_1^{t-1}$: Historical market Windows
Data: w: Window size
Data: m: Chebyshev Distance threshold
Data: ρ: Correlation Coefficient threshold
Data: l: Number of neighbors
Result: C_2: Similarity set
Initialize: $C_1 = \emptyset, C_2 = \emptyset$
for $i = w+1$ *to* $t-1$ **do**
  if $\| \mathbf{X}_{t-w}^{t-1} - \mathbf{X}_{i-w}^{i-1} \|_\infty \le m$ *or* $corrcoef(\mathbf{X}_{i-w}^{i-1}, \mathbf{X}_{t-w}^{t-1}) \ge \rho$ **then**
    $C_1 = C_1 \cup \{i\}$
  end
end
for $i \in C_1$ **do**
  if $\mathbf{X}_{i-w}^{i-1}$ *is among the* ℓNNs *of* $\mathbf{X}_{t-w}^{t-1}$ **then**
    $C_2 = C_2 \cup \{i\}$
  end
end

Algorithm 2: Portfolio optimization process

Data: $\mathbf{X}_1^{t-1}$: Historical price series
Data: w: Window size
Data: t: Current trading day
Data: C_2: Similarity set
Result: $\mathbf{b}_t$: Portfolio
if $t \le w+1$ **then**
  $\mathbf{b}_t = (\frac{1}{M}, \frac{1}{M}, ..., \frac{1}{M})$
end
if $C_2 = \emptyset$ **then**
  $\mathbf{b}_t = (\frac{1}{M}, \frac{1}{M}, ..., \frac{1}{M})$
else
  $\mathbf{b}_t = \arg\max_{\mathbf{b} \in \triangle_M} \prod_{i \in C_2} \mathbf{b} \cdot \mathbf{x}_i$
end

$$\mathbf{S}_n = \mathbf{S}_0 \prod_{t=1}^{n} \left(\mathbf{b}_t^{\mathrm{T}} \mathbf{x}_t \right) = \prod_{t=1}^{n} \left(\mathbf{b}_t^{\mathrm{T}} \mathbf{x}_t \right). \tag{14}$$

where the initial funds $\mathbf{S}_0 = 1$.

The key step of the above algorithm 2 is how to optimize the investment portfolio:

$$\mathbf{b}_t = \arg\max_{\mathbf{b} \in \Delta_M} \prod_{i \in C_2} \mathbf{b} \cdot \mathbf{x}_i \tag{15}$$

The Sequential Quadratic Program (SQP) algorithm [8] has a superlinear convergence speed, and the number of function and gradient evaluations is small. It can transform complex nonlinear constrained optimization problems into relatively simple quadratic programming (QP) problems, and is one of the most effective methods for solving nonlinear optimization problems with equality and bound constraints. Therefore, this article will use this algorithm to optimize the investment portfolio.

According to the different methods for solving subproblems in quadratic programming, the SQP method can be divided into line search SQP method and trust region SQP method. Both of these methods are important for ensuring global convergence, determining displacement and new iteration points in each iteration. The difference between the two lies in the fact that the line search method first determines the search direction, and then selects the search step size; The trust region method directly determines the direction step size and generates new iteration points.

This article uses the line search SQP algorithm. To perform an accurate line search, let G^k be the descent direction at the iteration point $\mathbf{b}^k$, and the step size is:

$$\alpha^k = \arg\min_{\alpha > 0} \left\{ \Omega(\alpha) = f\left(\mathbf{b}^k + \alpha G^k \right) \right\} \tag{16}$$

But in reality, this approach requires too much computation and does not require such precision, so we use the Armijo criterion for line search: Let G^k be the descent direction at the iteration point $\mathbf{b}^k$. Given the constant $c \in (0,1)$, if there is

$$f\left(\mathbf{b}^k + \alpha G^k \right) \le f\left(\mathbf{b}^k \right) + c\alpha \nabla f\left(\mathbf{b}^k \right) G^k \tag{17}$$

then the step size α satisfies the Armijo criterion.

In summary, the general process of using the line search SQP algorithm to solve portfolio optimization problems is:

First, set $f(\mathbf{b}) = -\prod_{i \in C} \mathbf{b} \cdot \mathbf{x}_i$, $\mathbf{a} = (1,1,\cdots,1) \in \mathbb{R}_+^M$, $b = 1$, $h(\mathbf{b}) = \mathbf{a} \cdot \mathbf{b} - b$, $l = (0,0,\cdots,0) \in \mathbb{R}_+^M$, $u = (1,1,\cdots,1) \in \mathbb{R}_+^M$, so solving the optimization portfolio problem requires solving the following nonlinear constrained optimization problem:

$$\begin{aligned} \min \quad & f(\mathbf{b}) \\ \text{s.t.} \quad & h(\mathbf{b}) = 0 \\ & l \le \mathbf{b} \le u \end{aligned} \tag{18}$$

The above constraints include: Equation constraint $h(\mathbf{b}) = \mathbf{a} \cdot \mathbf{b} - b = 0$, represents that the sum of the components of $\mathbf{b}$ is 1, and the value range of $\mathbf{b}$ constrains $l \le \mathbf{b} \le u$, indicating that the investment weight of each asset is between 0 and 1.

Secondly, through Taylor expansion, we approximate the objective function of the nonlinear constrained optimization problem to a quadratic function at the iteration point $\mathbf{b}^k$, and linearize the constraint conditions to obtain a quadratic programming subproblem:

$$\begin{aligned} \min \quad & f(\mathbf{b}) = \frac{1}{2}\left[\mathbf{b}-\mathbf{b}^k\right]^{\mathrm{T}} \nabla^2 f\left(\mathbf{b}^k\right)\left[\mathbf{b}-\mathbf{b}^k\right] + \nabla f\left(\mathbf{b}^k\right)^{\mathrm{T}}\left[\mathbf{b}-\mathbf{b}^k\right] \\ \text{s.t.} \quad & \nabla h\left(\mathbf{b}^k\right)^{\mathrm{T}}\left[\mathbf{b}-\mathbf{b}^k\right] + h\left(\mathbf{b}^k\right) = 0 \\ & l - \mathbf{b}^k \le \mathbf{b} - \mathbf{b}^k \le u - \mathbf{b}^k \end{aligned} \tag{19}$$

where the initial value of iteration point $\mathbf{b}^k$ will be set as a uniform investment portfolio $\mathbf{b}^0 = \left(\frac{1}{M}, \frac{1}{M}, \cdots, \frac{1}{M}\right) \in \mathbb{R}_+^M$. The above optimization problem, as an approximate problem of the original problem, has a locally better solution and is not the optimal solution of the original problem. However, this solution provides a direction for the subsequent iteration process. Through continuous iteration and optimization, we can gradually approach the optimal solution of the original problem. To achieve the goal, set:

$$G = \mathbf{b} - \mathbf{b}^k \tag{20}$$

At this point, the above quadratic programming problem is transformed into an optimization problem regarding variable G, namely:

$$\begin{aligned} \min \quad & f(\mathbf{b}) = \frac{1}{2} G^{\mathrm{T}} \nabla^2 f\left(\mathbf{b}^k\right) G + \nabla f\left(\mathbf{b}^k\right)^{\mathrm{T}} G \\ \text{s.t.} \quad & \nabla h\left(\mathbf{b}^k\right)^{\mathrm{T}} G + h\left(\mathbf{b}^k\right) = 0 \\ & l - \mathbf{b}^k \le G \le u - \mathbf{b}^k \end{aligned} \tag{21}$$

Thirdly, by instructing:

$$\begin{aligned} H &= \nabla^2 f\left(\mathbf{b}^k\right) \\ C &= \nabla f\left(\mathbf{b}^k\right) \\ A &= \nabla h\left(\mathbf{b}^k\right)^{\mathrm{T}} \\ B &= h\left(\mathbf{b}^k\right) \\ D &= l - \mathbf{b}^k \\ E &= u - \mathbf{b}^k \end{aligned} \tag{22}$$

Transform the optimization problem into a general form of a quadratic programming problem:

$$\begin{aligned} \min \quad & \frac{1}{2} G^{\mathrm{T}} H G + C^{\mathrm{T}} G \\ \text{s.t.} \quad & AG + B = 0 \\ & D \le G \le E \end{aligned} \tag{23}$$

The Lagrange function for optimization problems is

$$\mathcal{L}(G,\lambda)=\frac{1}{2}G^{\mathrm{T}}HG+C^{\mathrm{T}}G+\lambda(AG+B) \tag{24}$$

From the extremum conditions of multivariate functions $\mathcal{L}(G,\lambda)=0$, it can be concluded that:

$$\begin{aligned} HG+C+A^{\mathrm{T}}\lambda&=0\\ AG+B&=0 \end{aligned} \tag{25}$$

Write in matrix form as:

$$\begin{pmatrix} H & A^{\mathrm{T}}\\ A & 0 \end{pmatrix}\begin{pmatrix} G\\ \lambda \end{pmatrix}=\begin{pmatrix} -C\\ -B \end{pmatrix} \tag{26}$$

The above equation can be understood as a linear system of equations with $[G,\lambda]^{\mathrm{T}}$ as the variable, where both the number of variables and equations are $M+1$. The equations either have no solution or have a unique solution.

Finally, in the fourth step, the optimal solution G^{*} of the quadratic programming problem is taken as the next search (descent) direction G^{k} of the original problem. The search step size α^{k} is determined using the Armijo criterion. Then, the step size of α^{k} is moved in the G^{k} direction to obtain the next iteration point:

$$\mathbf{b}^{k+1}=\mathbf{b}^{k}+\alpha^{k}G^{k} \tag{27}$$

By iterating repeatedly until G^{k} satisfies a certain convergence accuracy, an approximate solution $\mathbf{b}^{k+1}$ for the original constraint problem can be obtained.

5. Experiments

5.1. Preparation

In order to test the empirical effect of the proposed strategy, we conducted extensive experiments on 2 real-world stock market data sets, which are summarized in Table 1.

For the convenience of obtaining and easy to reproduce, this article uses daily closing price data from the stock market. One dataset is widely used by other scholars in the field. NYSE (O) was first used by Cover (1991) [9] and has been reused by many later scholars. It consists of the closing prices of 36 stocks on the New York Stock Exchange over 5651 trading days.

In addition to the classic data sets NYSE (O), we collect an additional data set. The data set is the list of 19 high-market stocks in the China Securities 500 Index. The data spans 2405 trading days, or ten years, from December 25th 2012 to

Table 1. Summary of the 2 real data sets in our numerical experiments.

Dataset	Region	Time Frame	Trading days	Assets
CSI500	CN	Dec. 25th 2012 - Nov. 18th 2022	2405	19
NYSE(O)	US	Jul. 3rd 1962 - Dec. 31st 1984	5651	36

November 18th 2022. We call this data set "CSI500". This is a very new dataset that also takes into account the pandemic period of the past three years. The purpose of collecting this dataset is to determine whether our algorithm can adapt to the economic development of the new era and handle the impact of the epidemic on the economy. Also since the data sets are stored as relative prices, we do not need to consider exchange rates.

In addition, for comparative experiments, the benchmarks and 3 state-of-the-art portfolio-selected systems are selected, and the preliminary settings are as follows:

1. Market;

2. Best-Stock [10];

3. BCRP [11];

4. B^K. The parameters are set to $W=5$, $L=10$, $c=1.0$. According to the research of Györf *et al.*, it has the best empirical performance at this time;

5. B^{NN}. The parameters are set to $W=5$, $L=10$, $p_\ell=0.02+0.5\frac{\ell-1}{L-1}$ suggested by Györf *et al.*;

6. CORN. The CORN Uniform combination algorithm with the parameter setting $W=5$, $L=1$, $c=0.1$.

Simultaneously, the PMDI's parameters were set to $w=5$, $\rho=0.2$, $m=0.05$, $\ell=0.5\times t$. The performance of the above strategies was primarily evaluated by the following metrics: 1 and 2 are investment performance metrics, 3 and 4 are risk metrics. That is:

1. CW (Cumulative Wealth)

Cumulative wealth is the most important indicator for evaluating investment effectiveness, and the initial investment amount is set to $\mathbf{S}_0=1$, the formula is:

$$\mathbf{S}_n=\prod_{t=1}^{n}\left(\mathbf{b}_t^{\mathrm{T}}\mathbf{x}_t\right). \tag{28}$$

This has been calculated in chapter 2 and will not be repeated here.

2. APY (Annual Percentage Yield)

Given Cumulative wealth of $\mathbf{S}_n$, then the Annual Percentage Yield (APY) can be expressed as:

$$\mathrm{APY}=\left(\mathbf{S}_n\right)^{\frac{1}{Y}}-1, \tag{29}$$

where Y represents the number of years corresponding to n trading cycles. APY measures the average annual wealth increment achieved by trading strategies. In general, the higher the Cumulative Wealth or APY value, the more desirable the trading strategy is.

3. Risk (Annualized Standard Deviation of Daily return)

The annualized standard deviation σ is obtained by multiplying the standard deviation of daily returns σ_d by $\sqrt{252}$ (252 is the average number of trading days per year):

$$\sigma=\sigma_d\times\sqrt{252}. \tag{30}$$

The annualized standard deviation is an important indicator for measuring the volatility of investment portfolios, which can reflect the risk level of investment portfolios and help us better manage investment risks.

4. SR (Sharpe Ratio)

The Sharpe ratio [12] is one of the three classic indicators that can comprehensively consider both returns and risks. As a risk adjusted rate of return, it can eliminate the adverse effects of risk factors on performance evaluation. The formula for calculating the Sharpe ratio is:

$$\text{Sharpe Ratio} = \frac{\text{APY} - R_f}{\sigma}, \tag{31}$$

Wherein, R_f is the annualized risk-free interest rate, which generally adopts the interest rate of treasury bond in the same period. The purpose of this formula is to calculate how much excess return will be generated per unit of total risk undertaken by the investment portfolio.

5.2. Cumulative Wealth

The first experiment evaluates the CW obtained by various strategies without considering transaction costs.

Table 2 summarizes the CW obtained by various algorithms on four datasets, with bold numbers indicating the top ranked achievement on each dataset.

On the CSI500 dataset, the PMDI algorithm achieved the best results, which proves the superiority of the PMDI algorithm. Although the performance of the PMDI algorithm on the NYSE (O) dataset is not as good as other pattern matching algorithms, it is worth mentioning that the PMDI algorithm does not use expert algorithms, which greatly accelerates the running speed of the algorithm compared to other pattern matching algorithms. It is not easy to achieve a revenue of nearly 100 million dollars in this situation.

5.3. APY

The performance of APY is almost consistent with that of CW, as shown in the Table 3, so it will not be further elaborated here.

Table 2. The CW obtained through different strategies.

Strategies	CSI500	NYSE (O)
Market	4.72	14.50
Best-stock	19.51	54.14
BCRP	26.79	250.60
B^K	11.06	1.08E+09
B^{NN}	22.50	3.19E+11
CORN	30.62	1.45E+13
PDMI	61.20	3.60E+07

5.4. Risk

In this subsection, we examine the volatility of strategy returns, *i.e.* annualized standard deviation. As shown in Table 4, there is a correlation between the volatility of returns for each strategy and the cumulative and annualized returns, which further validates the investment principle of "high returns accompanied by high risks". By observing this indicator, we can have a clearer understanding of the risk levels of each strategy, thereby providing stronger basis for investment decisions.

As is well known, high risk can bring high returns. The PDMI algorithm achieved the best returns on the CSI500 dataset, but the risk indicator was not the highest. Therefore, it can be expected that the risk adjusted returns in the next section will be the highest.

5.5. Sharpe Ratios

Table 5 summarizes the performance of risk adjusted return indicators for each strategy—Annualized Sharpe Ratios, with risk-free rates generally set at 4%.

From the table, it can be seen that the Sharpe ratio of PMDI has always maintained a high level of over 100%.

6. Conclusions

There is certain reference value of the Portfolio Selection Method based on Pattern

Table 3. APYs of different strategies.

Strategies	CSI500	NYSE (O)
Market	18%	13%
Best-stock	37%	19%
BCRP	41%	28%
B^K	29%	153%
B^{NN}	39%	226%
CORN	43%	**286%**
PMDI	**54%**	117%

Table 4. Risks of different strategies.

Strategies	CSI500	NYSE (O)
Market	31%	15%
Best-stock	60%	24%
BCRP	47%	31%
B^K	36%	36%
B^{NN}	38%	40%
CORN	46%	49%
PMDI	48%	42%

Table 5. Sharpe ratios of different strategies.

Strategies	CSI500	NYSE (O)
Market	43%	58%
Best-stock	55%	64%
BCRP	79%	78%
B^K	68%	409%
B^{NN}	92%	551%
CORN	86%	578%
PMDI	103%	269%

Matching with Dual Information of Direction and Distance (PMDI) for the development of pattern matching methods in online portfolio selection algorithms.

On the one hand, this method has theoretical advantages. This article creatively proposes a method for extracting similarity sets driven by similarity measurement by analyzing various pattern matching algorithms in the past. By combining different similarity measurement methods, the KNN algorithm is used to select the best among them. After multiple screening of historical price information in stock trading, a similarity set with distance and direction information is obtained, and optimization algorithms are implemented on this similarity set.

On the other hand, we conducted extensive experiments on two datasets, including different stock data from the real world. PMDI outperformed all other comparable algorithms in one dataset. Although the results on another dataset were not as expected, we greatly reduced the runtime due to not using expert algorithms. Due to the completely different time periods and financial environments of the two datasets, it can be proven that the PDMI algorithm has robustness and adaptability in different market environments and time periods. Therefore, it is suitable for practical financial environments, including high-frequency trading.

We will continue to explore the possibilities of pattern matching based methods and contribute to future research on online portfolio selection problems.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Harry, M. (1952) Portfolio Selection. *The Journal of Finance*, **7**, 77-91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

[2] Li, B., Zhao, P., Hoi, S.C. and Gopalkrishnan, V. (2012) PAMR: Passive Aggressive Mean Reversion Strategy for Portfolio Selection. *Machine Learning*, **87**, 221-258. https://doi.org/10.1007/s10994-012-5281-z

[3] Li, B., Hoi, S.C.H., Sahoo, D. and Liu, Z.Y. (2015) Moving Average Reversion Strat-

egy for On-Line Portfolio Selection. *Artificial Intelligence*, **222**, 104-123. https://doi.org/10.1016/j.artint.2015.01.006

[4] Györfi, L., Lugosi, G. and Udina, F. (2006) Nonparametric Kernel-Based Sequential Investment Strategies. *Mathematical Finance*, **16**, 337-357. https://doi.org/10.1111/j.1467-9965.2006.00274.x

[5] Györfi, L., Udina, F. and Walk, H. (2008) Nonparametric Nearest Neighbor Based Empirical Portfolio Selection Strategies. *Statistics & Risk Modeling*, **26**, 145-157. https://doi.org/10.1524/stnd.2008.0917

[6] Cover, T. and Hart, P. (1967) Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, **13**, 21-27. https://doi.org/10.1109/TIT.1967.1053964

[7] Li, B., Hoi, S.C.H. and Gopalkrishnan, V. (2011) CORN: Correlation-Driven Nonparametric Learning Approach for Portfolio Selection. *ACM Transactions on Intelligent Systems and Technology*, **2**, 1-29. https://doi.org/10.1145/1961189.1961193

[8] Wilson, R.B. (1963) A Simplicial Algorithm for Concave Programming. Ph.D. Thesis, Harvard University, Cambridge.

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

[10] Borodin, A., El-Yaniv, R. and 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

[11] Cover, T.M. and Thomas, J.A. (1991) Elements of Information Theory. Wiley-Interscience, New York.

[12] Sharpe, W.F. (1965) Mutual Fund Performance. *The Journal of Business*, **39**, 119-138. https://doi.org/10.1086/294846