CORN: Correlation-Driven Nonparametric Learning Approach for Portfolio Selection

BIN LI, STEVEN C. H. HOI, and VIVEKANAND GOPALKRISHNAN, Nanyang Technological University

Machine learning techniques have been adopted to select portfolios from financial markets in some emerging intelligent business applications. In this article, we propose a novel learning-to-trade algorithm termed CORrelation-driven Nonparametric learning strategy (CORN) for actively trading stocks. CORN effectively exploits statistical relations between stock market windows via a nonparametric learning approach. We evaluate the empirical performance of our algorithm extensively on several large historical and latest real stock markets, and show that it can easily beat both the market index and the best stock in the market substantially (without or with small transaction costs), and also surpass a variety of state-of-the-art techniques significantly.

Categories and Subject Descriptors: J.1 [Computer Applications]: Administrative Data Processing—Financial; J.4 [Computer Applications]: Social and Behavioral Sciences—Economics; I.2.6 [Artificial Intelligence]: Learning

General Terms: Design, Algorithms, Economics, Experimentation

Additional Key Words and Phrases: Correlation coefficient, nonparametric learning, online portfolio selection

ACM Reference Format:
Li, B., Hoi, S. C. H., and Gopalkrishnan, V. 2011. CORN: Correlation-driven nonparametric learning approach for portfolio selection. ACM Trans. Intell. Syst. Technol. 2, 3, Article 21 (April 2011), 29 pages.
DOI = 10.1145/1961189.1961193 http://doi.acm.org/10.1145/1961189.1961193

1. INTRODUCTION

Recent years have witnessed machine learning being increasingly used in business applications. An active research topic in this domain is to study machine learning techniques for selecting portfolios. In general, portfolio selection [Markowitz 1952] aims to maximize performance measures such as total wealth, economic utility, or risk adjusted return, with the wealth invested in some financial markets in the long run. This problem has been extensively studied in computational finance, statistics, and information theory, and recently, it has also attracted increasing interest from the machine learning, data mining, and artificial intelligence communities. In this article, we investigate the portfolio selection problem by sequential investment (also termed online investment) strategies, which exploits information collected from the historical market and (actively) determines how a portfolio is to be distributed among a (fixed) set of assets.

Most extensive efforts from the finance domain towards this challenge can be generally classified as Fundamental Analysis (FA) and Technical Analysis (TA). FA
approaches [Graham and Dood 1934] aim to predict the expected return of a stock by measuring its *intrinsic* value based on related economic, financial, and other qualitative and quantitative factors. Instead of measuring the intrinsic values, TA approaches [Edwards and Magee 1948] believe that the historical performance of stocks are sufficient indicators of their future performance, and often adopt charts, technical indicators, and other tools to identify patterns that can help to predict future prices or suggest future activities.

In recent years, researchers in machine learning and data mining communities have attacked the portfolio selection problem by optimizing investment strategies via computer programs that are powered by intelligent learning algorithms. We refer to these approaches as the *learning-to-trade* techniques. These techniques are close to those from the TA category in finance domain in the sense that they also operate on historic price data. However, different from the heuristic trading techniques by the TA approaches, the *learning-to-trade* techniques are often well formulated by machine learning methods and solved effectively by optimization techniques. A variety of state-of-the-art learning-to-trade algorithms have recently been proposed in the literature [Cover 1991; Helmbold et al. 1998; Borodin et al. 2004; Agarwal et al. 2006; Györfi et al. 2006; 2008].

In addition, researchers have also attempted to establish theoretical foundations for the learning-to-trade approaches. A pioneering and widely studied work is the theoretical framework of *universal* portfolio selection [Cover 1991; Cover and Ordentlich 1996; Helmbold et al. 1998; Blum and Kalai 1999; Hazan 2006; Györfi et al. 2006; 2008], which provides performance guarantees of the *regret* based on information theory. While many universal algorithms theoretically achieve a nice performance guarantee, in practice they often perform no better than simple heuristic investment strategies from previous empirical studies. An intriguing and practical question that remains unresolved is “Can we develop a learning-to-trade algorithm that consistently surpasses the market and even beats the best stock in the market?”

In this article, we present a novel learning-to-trade strategy for the (sequential) portfolio selection problem, termed CORrelation-driven Nonparametric learning (CORN) algorithm. In particular, CORN seeks to locate the market windows that are similar to the latest market window via a correlation coefficient metric, and makes a log-optimum portfolio according to the idea of the best constant rebalanced portfolio [Cover 1991] strategy. CORN not only exploits effective statistical correlations between market windows, but also benefits from the exploration of powerful nonparametric machine learning techniques. Our empirical studies on historical stock markets, including datasets covering the period of the latest financial turmoil, show that CORN consistently surpasses a variety of state-of-the-art techniques. Even when transaction costs up to 0.2% are considered, CORN easily beats the market as well as the best stock in the market substantially.

Besides, our promising empirical results also provide a strong evidence to rebut the well-known Efficient Market Hypothesis (EMH) [Fama 1970] in finance theory, which asserts that no investors can consistently beat the market using information that is already known. Our empirical results on several large historical stock testbeds show that the proposed CORN algorithm can outperform the market and even beat the best stock in the market, using only the historical price information that is already known to all the market participants.

The rest of the article is organized as follows. Section 2 formally presents the portfolio selection problem, and gives some preliminaries of the background. Section 3 reviews the related work. Section 4 presents the proposed CORN algorithm for trading stocks actively. Section 5 examines the efficacy of the proposed algorithm by conducting an extensive set of empirical studies on several historical and up-to-date
stock markets. Section 6 summarizes the article and provides directions for future work.

2. PRELIMINARIES: PORTFOLIO SELECTION

In this section, we formulate the portfolio selection problem by following previous studies [Breiman 1961; Algoet and Cover 1988; Cover 1991; Ordentlich and Cover 1996].

2.1. Problem Formulation

Consider a market with \( m \) assets. Let us denote by \( \mathbf{x}_t = (x_{t,1}, \ldots, x_{t,m}) \in \mathbb{R}^m \) the price relative vector for the \( m \) assets in the \( t \text{th} \) trading day, where each element \( x_{t,i} \) equals the \( t \text{th} \) closing price of asset \( i \) divided by the \( (t-1) \text{th} \) closing price of asset \( i \), that is, \( x_{t,i} = \frac{p_{t,i}}{p_{t-1,i}} \). Given a window size \( w \), let us define the market window for the \( t \text{th} \) trading day as \( \mathbf{X}_{t-w}^t = (\mathbf{x}_{t-w}, \ldots, \mathbf{x}_{t-1}) \), which represents the latest market movement before the \( t \text{th} \) trading day.

At the beginning of \( t \text{th} \) trading day, we specify a portfolio \( \mathbf{b}_t = (b_{t,1}, \ldots, b_{t,m}) \in \mathbb{R}^m \) to allocate our wealth among \( m \) assets. Each component \( b_{t,i} \) represents the proportion of wealth invested in the \( i \text{th} \) asset at the beginning of \( t \text{th} \) trading day. One obvious constraint for a portfolio is that it must be a simplex, denoted by \( \mathbf{b}_t \in \Delta_m \), such that \( b_{t,i} \geq 0 \) and \( \sum_{i} b_{t,i} = 1 \), which means the portfolio is self-financed and no margin is allowed. The portfolio strategy for the period of \( T \) trading days is \( \mathbf{B}_T^1 = (\mathbf{b}_1, \ldots, \mathbf{b}_T) \), which is the output of the learning-to-trade strategy.

Thus, for the \( t \text{th} \) trading day, the portfolio achieves a daily return defined as \( \mathbf{b}_t \cdot \mathbf{x}_t = \sum_{i} b_{t,i} x_{t,i} \). And the total wealth achieved at the end of the \( T \text{th} \) trading day is

\[
\mathbf{S}_T = \mathbf{S}_0 \prod_{t=1}^{T} (\mathbf{b}_t \cdot \mathbf{x}_t),
\]

where \( \mathbf{S}_0 \) is the initial wealth, which is set to 1 for convenience in our study.

The goal of a learning-to-trade task is to learn a portfolio strategy that is expressed as a sequence of functions

\[
\mathbf{b}_t : (\mathbb{R}^m)^{t-1} \rightarrow \Delta_m, \quad t = 1, 2, \ldots,
\]

where \( \mathbf{b}_t(\mathbf{X}_{t-w}^{t-1}) \) represents the portfolio vector made by the investor at the beginning of the \( t \text{th} \) trading day upon observing the past behavior of the market. As a sequential investment strategy, the learning-to-trade strategy produces one portfolio vector every trading day. All of these vectors form the portfolio strategy for the entire trading period.

In the preceding, we make several general assumptions for the portfolio selection model.

1. **Transaction cost.** No transaction cost exists in the preceding model.
2. **Market liquidity.** Each asset is arbitrarily divisible, and we can buy and sell the desired quantities at the last closing price of any given trading period.
3. **Impact cost.** The market behavior is not affected by any decision made by the learning-to-trade strategy.

2.2. Performance Criteria

One natural and common performance metric is the total wealth factor achieved during some trading period by the learning-to-trade strategy. The total wealth factor equals the wealth achieved at the end of the trading period divided by the initial wealth. In our study, we simply set the initial wealth \( \mathbf{S}_0 = 1 \), and use the same notation \( \mathbf{S}_T \) to denote
the total wealth factor for convenience. Another equivalent metric is the *Annualized Percentage Yield* (APY) that takes account of the compounding effect, that is,

$$\text{APY} = \left( S_T \right)^{\frac{1}{T}} - 1,$$

where \( y \) is the number of years corresponding to the \( T \) trading periods. APY measures the average wealth increment per year achieved by a *learning-to-trade* strategy. Typically, the higher the value of total wealth factor or APY, the more preferable the trading strategy.

In portfolio management, an important concern is the evaluation of risk and risk-adjusted return of the portfolios. A common way to achieve this is to adopt the *annualized standard deviation* of daily returns to measure the volatility risk, and the *annualized Sharpe Ratio* (SR) [Sharpe 1966] to evaluate the risk-adjusted return. For the portfolio risk, we calculate the standard deviation of the daily returns, and multiply by \( \sqrt{252} \) (here 252 is the average number of trading days per year) to obtain the annualized standard deviation. For the risk-adjusted return, we calculate annualized Sharpe Ratio according to the following formula

$$\text{SR}_T = \frac{\text{APY} - R_f}{\sigma_p},$$

where \( R_f \) is the risk-free return (typically the return of Treasury bills, set at 4% in our study), and \( \sigma_p \) is the annualized standard deviation of daily returns. Typically, the higher the annualized Sharpe Ratio, the more preferable the trading strategy.

For portfolio management, another risk evaluation is *DrawDown (DD)* analysis, which measures the decline from a historical peak of the total wealth achieved by a trading strategy. Formally, let \( S(\cdot) \) denote the process of the total wealth achieved by a trading strategy, that is, \( \{ S_1, \ldots, S_t, \ldots, S_T \} \). The drawdown at any time \( t \), denoted as \( \text{DD}(t) \), is defined as:

$$\text{DD}(t) = \sup_{i \in (0,t)} S(i) - S(t).$$

The *Maximum DrawDown (MDD)* until the end of the trading period is the maximum of the drawdown over the history of the total wealth achieved by a learning-to-trade strategy. MDD is a good way to measure the inherent risk of different trading strategies. More formally, the maximum drawdown for a horizon \( T \), denoted as \( \text{MDD}(T) \), is defined as

$$\text{MDD}(T) = \sup_{\tau \in (0,T)} \left[ \sup_{t \in (0,\tau)} S(t) - S(\tau) \right].$$

The smaller the maximum drawdown value, the more risk-tolerable the trading strategy.

### 2.3. Some Practical issues in Portfolio Selection

In a real-world portfolio selection task, there are some practical issues that should be taken into consideration. Next we discuss two practical issues, and relax our previous formulation to address these issues properly.

In reality, an important and unavoidable issue is *transaction cost*. In our study, we adopt the *proportional transaction cost* model following Blum and Kalai [1999] and Borodin et al. [2004], that is, an action of rebalancing the portfolio has to incur transaction cost for both buy and sell operations. At the beginning of the \( t \)th trading day, the portfolio manager rebalances the portfolio from the previous closing price adjusted portfolio \( \hat{b}_{t-1} \) to a new portfolio \( b_t \). Specifically, we consider a transaction cost rate \( \gamma \in (0, 1) \), so the transaction cost will be charged according to \( \frac{\gamma}{2} \times \sum_i |b_{t,i} - \hat{b}_{t-1,i}| \), where the initial portfolio is set to \((0, \ldots, 0)\). Thus, with transaction cost rate \( \gamma \),
the total wealth achieved by the end of the $T^{th}$ trading day, denoted as $S_{T}^{c(y)}$, is expressed as

$$S_{T}^{c(y)} = S_0 \prod_{t=1}^{T} \left[ (b_t \cdot x_t) \times \left( 1 - \frac{y}{2} \times \sum_i |b_{t,i} - \hat{b}_{(t-1),i}| \right) \right].$$

Another practical issue is margin buying, which allows portfolio managers to buy securities with cash borrowed from security brokers. Margin buying magnifies the profit as well as the loss invested in the securities. Following the previous studies [Cover 1991; Helmbold et al. 1998; Agarwal et al. 2006], we include this constraint in our model. In our study, the margin setting is assumed to be 50% down and 50% loan, and the interest rate of the borrowed money is $c$, which is simply set to $c = 0.000233$ in our study, or equivalently, an annual interest rate of 6%. Thus, for each security in the asset pool, we create a new asset named “Margin Component”. Following the down and loan percentage, the price relative for the “Margin Component” of asset $i$ would be $2 \times x_{(t,i)} - 1 - c$, where $x_{(t,i)}$ is the price relative of the $i^{th}$ asset for the $t^{th}$ trading day. By adding this “Margin Component”, we magnify both the potential profit and loss of the trading strategy.

3. RELATED WORK
We now review a variety of learning-to-trade techniques for the portfolio selection problem.

3.1. Natural Baseline Strategies
One common baseline for portfolio selection is the Buy-And-Hold (BAH) strategy, that is, one invests the money among a set of assets according to the initial portfolio $b$, and holds the portfolio without any change during the entire trading period. The BAH strategy with an uniform portfolio, that is, $b = (\frac{1}{m}, \ldots, \frac{1}{m})$, is known as the uniform BAH strategy. In our study, we refer to uniform BAH as the Market strategy that generates the market index.

Contrary to the static BAH strategy, active trading strategies often change portfolios regularly during the trading periods. A classical strategy is Constant Rebalanced Portfolios (CRP), which adjusts the portfolio to keep a fixed fraction of the investor’s total wealth in each of the underlying investments at every trading day. Formally, given a predefined portfolio strategy $b$ for CRP, the total wealth achieved by CRP at the end of the $T^{th}$ trading day is, $S_T = S_0 \prod_{t=1}^{T} (b \cdot x_t)$. A special case of CRP is to uniformly redistribute the total wealth to all investments, that is, $b = (\frac{1}{m}, \ldots, \frac{1}{m})$, which is known as Uniform CRP (UCRP). The best possible CRP strategy is often called Best CRP (BCRP), whose total wealth can be represented as $S_{T}^{\ast} = \max_{b \in \Delta_m} S_T$. Apparently, BCRP is only a hindsight strategy, which is practically not applicable.

3.2. Follow-The-Leader Strategies
The follow-the-leader strategies often attempt to achieve the same wealth as some offline best experts. Typically, the best expert is often based on the Best Constant Rebalanced Portfolios (BCRP). Formally, follow-the-leader strategies aim to minimize the regret between the strategy $A$ and the BCRP strategy at the horizon $n$: $\text{Regret}_n(A) = \sum_{t=1}^{n} \log(b^* \cdot x_t) - \sum_{t=1}^{n} \log(b_t \cdot x_t)$. Example techniques in this category include Cover’s universal portfolios [Cover 1991], exponential gradient strategy [Helmbold et al. 1998], and online Newton step strategy [Agarwal et al. 2006].

Cover [1991] proposed Universal Portfolio (UP) strategy, where the portfolio is the historical performance weighted average of all constant rebalanced portfolio experts.
The regret achieved by Cover’s UP is $O(m \log T)$, and its runtime complexity is $O(T m^2)$, where $m$ denotes the number of stocks and $T$ denotes the number of trading days. The implementations are exponential in the number of stocks which restricts the number of assets used in experiments. Kalai and Vempala [2002] presented a time-efficient implementation of Cover’s UP based on nonuniform random walks that are rapidly mixing, which requires poly running time $O(m^7 T^8)$. Following their works, Cover and Ordentlich [1996] developed universal procedures in the case where side information is taken into account as a finite number of values. Belentepe [2005] presented a statistical view of Cover’s UP, showing that it is approximately equivalent to a constrained sequential portfolio optimization, which connects Cover’s UP with traditional mean-variance portfolio theory.

Another famous learning-to-trade approach is the Exponential Gradient (EG) strategy [Helmbold et al. 1998] for the online portfolio selection problem using multiplicative updates. In general, the EG strategy tries to maximize the expected logarithmic portfolio daily return (approximated using the last price relative), and minimize the deviation between the expected portfolio and last portfolio. The regret achieved by EG strategy is $O(\sqrt{T \log m})$ with $O(T m^2)$ running time. While its regret is not as tight as Cover’s UP, its linear time complexity substantially surpasses the latter.

Recently convex optimization has been applied to resolve the PS problem. Examples include the Online Newton Step (ONS) strategy [Agarwal et al. 2006], which aims to maximize the expected logarithmic cumulative wealth (approximated using historical price relatives) and minimize the deviation of the expected portfolio. ONS exploits the second-order information of the log wealth function and applies it to the online scenario. It theoretically achieves the regret $O(m \log T)$ that is the same as Cover’s UP, and has running time complexity of $O(T m^2)$. Following this work, Hazan and Seshadhri [2009] recently proposed a new adaptive-regret approach, which is essentially also an ONS-based strategy though they provide more decent theoretical results.

### 3.3. Similarity-Driven Strategies

The similarity-driven learning-to-trade strategies usually optimize the trading strategy by mining potentially similar information from historical market sequences. Example techniques in this category include the Anticor algorithm [Borodin et al. 2004], the nonparametric kernel-based moving window learning strategy [Györfi et al. 2006], and the nonparametric nearest neighbor learning strategy [Györfi et al. 2008].

Borodin et al. [2004] proposed an algorithm named Anticor, which seeks to explore the statistical relations between all pairs of stocks in the market. It actually makes bets on the consistency of positive lagged cross-correlation and negative autocorrelation. Unlike previous approaches, this heuristic algorithm does not try to pursue any target strategy. Although it does not have theoretical guarantee, Anticor outperforms all other existing strategies in most cases. Our algorithm is partially inspired by the idea of statistical correlation adopted in this work.

In addition, Györfi et al. [2006] recently introduced a framework of nonparametric kernel-based moving window ($B^K$) learning strategies for PS based on nonparametric prediction techniques [Györfi and Schäfer 2003]. Their algorithm first identifies a list of similar historical price relative sequences whose Euclidean distances to the recent market windows are smaller than a threshold, and then it optimizes the portfolio with respect to the list of similar sequences. Under the same framework, Györfi et al. [2007] proposed another variant called nonparametric kernel-based semi-log-optimal strategy, which is actually an approximation of the $B^K$ strategy, mainly to improve the computational efficiency. Following the same framework, the nonparametric nearest-neighbor learning ($B^{NN}$) strategy proposed by Györfi et al. [2008] aims to search for
the $\ell$ nearest neighbors in the historical price relative sequences rather than price relatives within a specified Euclidean ball. This method has been empirically shown to be a rather robust trading strategy.

3.4. Time-Series-Prediction-Based Strategies

In financial engineering, there are a number of well-studied time-series prediction models [Tsay 2002]. These models may be adapted to the portfolio selection tasks, although they were not proposed to optimize this problem. These models can be generally categorized as linear and nonlinear models. 

**AutoRegressive Moving Average (ARMA)** [Box et al. 1970] is considered among the most important linear models. Combining an AutoRegressive (AR) model with a Moving Average (MA) model, this model is often denoted as ARMA($p$, $q$), where $p$ is the order of the autoregressive part and $q$ is the order of the moving average part. Other ARMA variants include **AutoRegressive Integrated Moving Average** (ARIMA) models and **AutoRegressive Fractionally Integrated Moving Average** (ARFIMA) models, etc. On the other hand, there are also some well-studied nonlinear models, such as **AutoRegressive Conditional Heteroskedasticity** (ARCH) models [Engle 1982], which represent the changes of variance along time. One of the most widely used representation of ARCH models is **Generalized AutoRegressive Conditional Heteroskedasticity** (GARCH) [Bollerslev 1986], which considers past variances for the future explanation of future variances, and thus is used to model the serial dependence of volatility. It is often denoted as GARCH($p$, $q$), where $p$ denotes the order of the variance forecast and $q$ is the order of the white noise disturbance.

4. CORN: CORRELATION-DRIVEN NONPARAMETRIC LEARNING STRATEGY

In this section, we present a new learning-to-trade strategy termed CORrelation-driven Nonparametric learning algorithm (CORN).

4.1. Motivation

The general idea for the similarity-driven learning-to-trade strategies is to optimize the trading strategy by mining similar patterns/information from historical market sequences. Among the existing similarity-driven learning strategies, Anticor [Borodin et al. 2004] attempts to find statistical relations between pairs of stocks, while the nonparametric learning strategies [Györfi et al. 2006; 2008] attempt to discover similar appearances or market windows. Although Anticor is successful in mining the statistical relations between pairs of stocks, they ignore the price movements of the whole market which are crucial for portfolio selection. Besides, the portfolio strategy learned by Anticor is rather heuristic, which could lead to suboptimal solutions.

On the other hand, the existing nonparametric learning strategies [Györfi et al. 2006; 2008] rely on Euclidean distance for similarity measure between the latest market window and the historical market windows. However, the main limitation of Euclidean measure is that it does not exploit the direction information of the market windows’ movements. As a result, it may detect some similar appearances, but it often includes some potentially useless or even harmful price relatives, and at the same time excludes many beneficial price relatives.

To better understand the drawbacks of using Euclidean distance for measuring the similarity between different market windows, we give an intuitive motivating example in Figure 1. Assume a market consists of only one asset, and the window size is fixed to 2. Let the latest market window for the $n^{th}$ trading day $X_{n-1}^{n-2} = (1.10, 1.10)$ and the radius of Euclidean norm ball $r = 0.2$. Consider three possible market windows A:($1.10, 1.05$), B:($0.96, 0.96$), and C:($1.2, 1.3$) as shown in Figure 1. In Figure 1, the
Fig. 1. A motivating example to illustrate the limitation of the Euclidean measure. The left diagram represents the absolute price movements of market windows A, B, C, and \(X_{n-2}^{n-1}\) in the consecutive three days (here the first price is only for the calculation of the price relatives). The starting prices of all of them are set to 100. The numbers in the parentheses of A, B, and C show their three-day prices, and the latest price for \(X_{n-2}^{n-1}\) is \(X_{n-2}^{n-1} = (100, 110, 121)\). The right diagram shows the corresponding price relative movements of the four market windows for the two trading days. The numbers in the parentheses of A, B, C, and \(X_{n-2}^{n-1}\) are their price relative vectors.

The left figure shows the virtual price movement trends adjusted with the same starting price $100 for all the market windows, while the right figure shows the corresponding price relatives for all the market windows. According to the principle of locating similar market windows that have the most similar moving trends as the latest market window, we should locate the market windows A and C that have the similar upward moving trends, and avoid including window B that has the dissimilar downward moving trend as indicated in the left figure. However, the Euclidean-based approach, that is, \(\|X_{t-1} - X_{n-2}^{n-1}\| \leq 0.2\), indicates that market windows A and B are detected as most similar to the latest market window \(X_{n-2}^{n-1}\), while market window C is excluded from the similar set. As a consequence, the power of subsequent optimizing the trading strategy from the resulting set of market windows will considerably suffer irrelevant or even harmful market windows (such as market window B) and the neglect of beneficial market windows (such as market window C). This motivates us to overcome the limitation by exploring more effective approaches.

4.2 Basic Idea and Definition

CORN is mainly inspired by the idea of exploiting statistical correlations between market windows in the historical stock market, and also driven by the consideration of exploring powerful nonparametric learning techniques to effectively optimize the portfolio.

Traditionally portfolio selection methods in finance try to estimate a target function based on past data and build the portfolios based on the learned function. However, since the financial market is complex and accurately modeling its movement is a difficult task, we adopt a nonparametric learning approach (or instance-based learning, case-based learning) [Aha et al. 1991; Aha 1991; Cherkassky and Mulier 1998]. Nonparametric learning makes no assumption on the distribution, and tries to capture the knowledge from the stored training data without building a target function. In particular, CORN locates similar price appearances among all past price relatives at the beginning of every trading day, and then builds the portfolio directly based on these appearances. This is achieved without estimating any global function of the market movement, but by estimating a target value of the next price relative.

To overcome the limitation of Euclidean measure in mining historical market windows and the negligence of the whole market movement of the existing strategies, we propose to employ the Pearson product-moment correlation coefficient, which is an
effective tool for measuring statistical correlations. It is also worth noting that the proposed CORN strategy measures the statistical correlations between market windows of all stocks rather than pairs of stocks as Anticor does. Since market windows of all stocks represent market movements in the specific time frames, it could be more effective to match the similar price relatives regarding the whole market.

We declare a correlation-similar set as one that contains historical price relatives whose previous market windows are statistically correlated to the latest market window, and formally define it as

\[
C_t(w, \rho) = \left\{ w < i < t - 1 \left| \frac{\text{cov}(X_{t-w}^{i-1}, X_{t-w}^{i-1})}{\text{std}(X_{t-w}^{i-1}) \cdot \text{std}(X_{t-w}^{i-1})} \geq \rho \right. \right\},
\]

where \( w \) is the market window size, \(-1 \leq \rho \leq 1\) is a parameter of correlation coefficient threshold, \( \text{cov}(A, B) \) denotes the covariance between market windows \( A \) and \( B \), and \( \text{std}(A) \) denotes the standard deviation of market window \( A \). If either std term equals 0, that is, the market is of zero volatility in the specific market window, we will then simply set its correlation coefficient to 0. In the calculation of preceding formula, both market windows \( X_{t-w}^{i-1} \) and \( X_{t-w}^{i-1} \) are concatenated into \( m \times w \)-dimensional vectors, and we can obtain the univariate correlation coefficient between the two market windows.

The correlation coefficient distinguishes the proposed CORN algorithm from the previous nonparametric learning strategies in the following aspects. Firstly, all the competing approaches use the price relatives, that is, the changes of the absolute prices. As shown in the motivating example, their drawback is that the Euclidean distance only considers the strength of the difference between the two price relatives, without considering the direction information. To overcome this drawback, we propose a correlation coefficient to measure the similarity between the two price relatives. With this important direction information, we can better identify the similar price relatives, thus leading to the superior performance of the proposed strategy. The Euclidean distance may be used to measure the direction information directly, for example, by using the slope of the centralized points (refer to Section 4.4 for details). However, used this way, it only measures the direction information but ignores the strength information. Thus, the proposed correlation coefficient is advantageous in that it not only considers the strength information but also the direction information of the price relatives, which are balanced appropriately.

Second, it is worth noting that in the calculation of univariate correlation coefficients, we will calculate the arithmetic mean return of all the \( m \times w \)-dimensional vectors. This mean return is uniformly distributed among \( m \) stocks, which is the same as the market strategy. As a result, the mean return actually reflects the whole market movements during the windows. The correlation coefficient measures the linear dependency between the two market windows, during which the mean return of the two market windows represents the whole market movements. This distinguishes the proposed CORN strategy from the previous Anticor strategy and the nonparametric learning strategies, all of which ignore the whole market information.

Third, the correlation coefficient not only reflects the degree of linear dependence or similarity, but also takes into account the directions of the vectors indicated by the signs. Although \(-\rho\) and \(\rho\) (\(\rho > 0\)) intuitively correspond to equivalent strength of linear dependence or similarity, they are in the opposite directions, that is, one is same-trending and the other is opposite-trending with respect to the target market window. We choose \(\rho\) as the threshold, as we are interested in market windows which are similar in terms of both strength and direction. The direction information also distinguishes the proposed CORN algorithm from the previous nonparametric learning strategies.
4.3. Algorithm

Next we present the proposed CORrelation-driven Nonparametric learning (CORN) algorithm, which aims to exploit the correlation-similar set in optimizing the portfolios for active trading. In general, CORN has two major steps. The first step is to define experts whose tasks are to locate the similar historical price relatives and learn to find an optimal portfolio based on the similar historical price relatives. The second step is to effectively combine the portfolios produced by the experts to form the final portfolio.

We first start by defining a set of infinite experts, each expert indexed by \((w, \rho)\), that is, \(\{E(w, \rho): w \geq 1, -1 \leq \rho \leq 1\}\). The expert \(E(w, \rho)\) is identified by its window size \(w\) and correlation coefficient threshold \(\rho\). Empirically, the infinite set of experts could be fixed to a finite number \(W \times P\), where \(W\) represents the maximum window size and \(P\) represents the number of correlation coefficient thresholds. In general, we can define an expert \(E(w, \rho)\) as \(E(w, \rho) = \mathbf{b}(w, \rho)\).

For each expert \(E(w, \rho)\), after calculating the correlation-similar set \(C_t(w, \rho)\) at the beginning of the \(t\)th trading day, we propose to learn the optimal portfolio by following the idea of the BCRP [Cover 1991] strategy to maximize the total wealth over the sequence of price relatives, that is,

\[
\mathbf{b}_t(w, \rho) = \arg \max_{\mathbf{b} \in \Delta_m} \prod_{i \in C_t(w, \rho)} (\mathbf{b} \cdot \mathbf{x}_i), \quad (2)
\]

where \(\Delta_m\) represents a simplex with \(m\) components. In case \(C_t(w, \rho)\) becomes empty (especially for a large \(\rho\) value), we will simply adopt a uniform portfolio \(\left(\frac{1}{m}, \ldots, \frac{1}{m}\right)\). The general procedure for each expert is summarized in Algorithm 1 shown in Figure 2.
It should be noted that the correlation-similar set usually consists of a large number of correlated price relatives. Thus, if one price relative (whose correlation is within the threshold) has occurred frequently in the history, it will also appear many times in the correlation-similar set. In other words, Eq. (2) has somewhat considered the occurrence/confidence of the correlated price relatives, which avoids simply taking an extreme case in the history.

Further, we discuss the strategy for combining the outputs from the set of experts. We combine them according to the historical performance of each expert \( s_{t-1}(w, \rho) \) and a probability distribution function \( q(w, \rho) \). Specifically, the final portfolio for the \( t \)th trading day can be calculated as

\[
b_t = \frac{\sum_{w,\rho} q(w, \rho)s_{t-1}(w, \rho)b_t(w, \rho)}{\sum_{w,\rho} q(w, \rho)s_{t-1}(w, \rho)},
\]

where \( b_t(w, \rho) \) represents the portfolio output by each expert \( E(w, \rho) \) and \( s_{t-1}(w, \rho) \) represents its historical performance (in our study we use the total wealth it achieved). For an individual expert, the higher the value of the return \( s_{t-1}(w, \rho) \), the higher weight will be assigned in the combination of the final portfolio. Once we calculate \( b_t \) by the previous equation, we will output it as the desired portfolio for the \( t \)th trading day, which will be used by the portfolio manager for the portfolio selection task.

Finally, the CORN strategy updates the total wealth achieved as

\[
S_t = S_{t-1} \times (b_t \cdot x_t),
\]

where \( S_{t-1} \) represents the total wealth achieved until the \( (t-1) \)th trading day and initial capital \( S_0 = 1 \). For each expert, CORN updates its performance \( s_t(w, \rho) \) after \( t \) trading periods, which can be calculated as

\[
s_t(w, \rho) = s_{t-1}(w, \rho) \times (b_t(w, \rho) \cdot x_t),
\]

where \( s_{t-1}(w, \rho) \) represents the total wealth achieved by the expert \( E(w, \rho) \) at the end of \( (t-1) \)th trading day and the initial capital is set to 1, that is, \( s_0 = 1 \).

Therefore, it is not difficult to see that the total wealth achieved by the proposed CORN strategy after \( T \) trading periods is equivalent to the sum of the weighted return of all experts based on the probability distribution \( q(w, \rho) \), that is,

\[
S_T = \sum_{w,\rho} q(w, \rho)s_T(w, \rho).
\]

It is clear that the final result is affected by all the experts, and the portion of contribution made by each of the experts is determined by the choice of distribution \( q(w, \rho) \) and the expert’s performance \( s_T(w, \rho) \).

In terms of different expert combinations, we present two CORN variants, that is, the CORN Uniform combination algorithm (CORN-U) and the CORN top-K combination algorithm (CORN-K). The CORN-U algorithm simply considers \( q(w, \rho) \) as a uniform distribution, that is, \( q(w, \rho) = \frac{1}{W} \), where \( W \) is the maximum number of windows, which uniformly combines all the experts. In this algorithm, we assign all the experts the same weights, although such weights can be adjusted if we could obtain more information on the distribution of the experts. Moreover, CORN-U considers \( P = 1 \) and chooses a specific value of \( \rho \). The details of the CORN-U algorithm are shown in Figure 3.

The preceding uniform combination algorithm may include some poor experts, leading to degradation of the overall performance. To overcome this limitation, the second algorithm, CORN-K, combines only the top-\( K \) best experts to form the final portfolio. In particular, it chooses the \( K \) experts with best historical returns and uniformly combines them, that is, the strategy assigns the set of top-\( K \) best experts a uniform distribution.
The proposed CORN Uniform combination algorithm (CORN-U): $q(w, \rho) = \frac{1}{W}$, while the weights assigned for other experts are simply set to 0. Moreover, for the CORN-K algorithm, we set $P$ to be larger than 1. For each $W$, we assign $P$ associated experts, each having a different $\rho$ value. In our empirical study, the $\rho$ value of the $i^{th}$ expert is set to $\frac{i-1}{P}$. The CORN-K algorithm is presented in Algorithm 3 as shown in Figure 4.

4.4. Geometrical Interpretation

In this section, we analyze the principle of the CORN algorithm from an intuitive geometrical perspective. The key step of the proposed CORN algorithm is to locate the similar correlation coefficient set. For simplicity, we assume the market windows are given in a two-dimensional space. Figure 5 shows an intuitive example that corresponds the example used in Section 4.1 from a geometrical view. In the figure, the origin point $(\mu_i, \mu_t)$ denotes the mean point of price relative vectors, $X_{t-1}^{i-w}$ denotes the market window of current $t^{th}$ trading day, $X_{t-1}^{i-1}$ denotes the market window of the $i^{th}$ trading day on the historical price relative sequence, and points A, B, and C represent another three market windows on the historical price relative sequence.

From a geometrical view of point [Rodgers and Nicewander 1988], we know that the correlation coefficient between two market windows $X_{t-1}^{i-w}$ and $X_{t-1}^{i-1}$ is equivalent to the cosine of angle $\theta$ between these two vectors, that is, $\cos \theta = \text{corrcos}(X_{t-1}^{i-w}, X_{t-1}^{i-1})$. Thus, given a correlation coefficient threshold $\rho$, the approach of searching for market windows satisfying $\text{corrcos}(X_{t-1}^{i-w}, X_{t-1}^{i-1}) \geq \rho$ is equivalent to finding market windows $X_{t-1}^{i-1}$ with $|\theta| \leq \text{arccos} \rho$. When $\rho$ is simply fixed to 0, it reduces to looking for market window vectors $X_{t-1}^{i-1}$ which have angle $|\theta| \leq 90^\circ$ with respect to $X_{t-1}^{i-1}$. In another words, the CORN strategy locates all market windows $X$ that satisfy $a^\top X \geq 0$ or intuitively those points on the right-hand side of line $a^\top X = 0$, where $a$ is a unit vector that is perpendicular to the vector from $(\mu_i, \mu_t)$ to $X_{t-1}^{i-1}$.
Algorithm 3 The proposed CORN TOP-K combination Algorithm (CORN-K)

Input:
\[ X^T_i = (x_1, \ldots, x_T) \]: historical market windows, \( W \): maximum window size for experts, \( P \): maximum number of correlation coefficient thresholds, \( K \): the value of \( K \) for the TOP-K experts

Output:
\( (b_1, b_2, \ldots, b_T) \): the output portfolio strategy

Procedure
1: Initialize \( S_0 \) and \( W \times P \) experts: \( S_0 = 1, \mathcal{P} = \left\{ 0, \frac{1}{P}, \ldots, \frac{P-1}{P} \right\} \), \( q(w, \rho) = \frac{1}{W \times P} \)
2: for \( t = 1 \) to \( T \) do
3: \hspace{1em} for \( w = 1 \) to \( W \) do
4: \hspace{2em} for \( \rho \in \mathcal{P} \) do
5: \hspace{3em} CORN Expert Learning (Algorithm 1) to find the portfolio:
6: \hspace{3em} \( b_t(w, \rho) = \mathcal{E}(w, \rho) \)
7: \hspace{2em} end for
8: \hspace{1em} end for
9: Combine the TOP-K experts’ portfolios:
10: \hspace{1em} \( b_t = \frac{\sum_{w, \rho} q(w, \rho) s_{t-1}(w, \rho) b_t(w, \rho)}{\sum_{w, \rho} q(w, \rho) s_{t-1}(w, \rho)} \)
11: Update the total wealth: \( S_t = S_{t-1} \times (b_t \cdot x_t) \)
12: Update the experts: \( s_t(w, \rho) = s_{t-1}(w, \rho) \times (b_t(w, \rho) \cdot x_t) \)
13: TOP-K and expert weight updates:
14: Select top K experts \( \{ \mathcal{E}(w, \rho) \} \) w.r.t. \( s_{t-1}(w, \rho) \)
15: Set weights for the top K experts: \( q(w, \rho) = \frac{1}{K} \)
16: Set weights for other experts: \( q(w, \rho) = 0 \)
17: end for

Fig. 4. The proposed CORN TOP-K combination algorithm (CORN-K).

Fig. 5. Geometrical interpretation of the proposed CORN strategy in comparison to the nonparametric kernel-based moving window (\( B^K \)) learning strategy.
On the other hand, the nonparametric learning strategy $B^K$ aims to locate market windows $X_{i-w}^{-1}$ within a Euclidean ball centered at $X_{i-w}^{-1}$ with radius $r_{k,l}$, that is, $||X_{i-w}^{-1} - X_{i-w}|| \leq r_{k,l}$. In contrast to the correlation coefficient approach used by the CORN approach, the major limitations of the Euclidean-based approach are twofold. First, from the geometrical view, it is clear that it neglects the directional information. As a result, it may include some irrelevant or negative market windows. For example, according Euclidean measurement, point $B:(0.96, 0.96)$ is within the Euclidean norm ball and hence is regarded as the similar case. However, this is a harmful window as its trend is completely different in direction from the latest market window $X_{t-w}^{-1}$. Moreover, it may also exclude some informative and beneficial market windows. For example, point $C:(1.20, 1.30)$ is excluded by the Euclidean approach, but is considered as an important market window as it is highly positively correlated with $X_{t-w}^{-1}$. Second, the Euclidean-based approach clearly does not consider the market information, which is represented by point $(\mu_i, \mu_t)$ in the figure.

Note that the preceding analysis could be easily extended to multidimensional vectors in general scenarios, that is, $w \times m$ dimensions where $w$ is the window size and $m$ is the number of stocks. The previous geometrical analysis again validates the importance and efficacy of the CORN algorithm.

### 4.5. Analysis of Parameters

In the CORN expert learning procedure, there are two key parameters: the correlation coefficient threshold $\rho$ and the window size $w$. Next we analyze how they affect the algorithms.

As shown in the motivating example in Section 4.1, the correlation coefficient threshold $\rho$ is critical to the correlation similar set. If $\rho$ is negative, the correlation similar set would contain some negatively related price relative vectors or irrelevant price relative vectors. On the other hand, if $\rho$ is too large, for example, $\rho \geq 0.5$, the correlation similar set would neglect some positively correlated price relative vectors. Since the correlation similar set is crucial for the selection of optimal portfolios, it would harm the learning performance if it either contains negatively related price relative vectors/irrelevant price relative vectors or discards positively correlated price relative vectors. Empirically, we found that the optimal $\rho$ value is dataset dependent, but often close to 0, which will be verified in Section 5.9.

Moreover, we note that CORN would reduce to a special case when setting $\rho \rightarrow 1$. In particular, when the correlation coefficient threshold $\rho \rightarrow 1$, the CORN algorithm reduces to the Uniform CRP (UCRP) strategy. It is straightforward to verify this by noting that when $\rho \rightarrow 1$, fewer market windows are highly positively correlated to the latest window. In the extreme case of $\rho = 1$, $C(w, \rho)$ becomes almost empty, which thus reduces to the Uniform CRP strategy. This will be numerically verified in Section 5.9.

Another key parameter for the CORN expert learning process is the window size. Since the calculation of correlation coefficient treats the market windows as a vector, the window size does not have a significant effect on the final portfolio. In the situation when certain experts give very bad predictions, the final result tends to be relatively stable since the proposed combination methods, namely, CORN-U and CORN-K, will reduce the impact of the bad predictions and thus provide a stable final result. We will numerically analyze the effect of the maximum window size in Section 5.9, which shows that there is only a trivial effect of this parameter.

Remarks. The simplicity and effectiveness of CORN raises the question whether it is reasonable to select a portfolio using only the market price information. While our goal is not to resolve the philosophical debate between fundamental and technical analysts, we believe this work goes a long way to provide empirical evidence that...
Table I. Summary of Four Real Datasets

| Dataset  | Market | Region | Time frame | # Trading days | # Assets |
|----------|--------|--------|------------|----------------|---------|
| NYSE (O) | Stock  | US     | July 3rd 1962 - Dec 31st 1984 | 5651           | 36      |
| NYSE (N) | Stock  | US     | Jan 1st 1985 - Jun 30th 2009 | 6179           | 23      |
| SP500    | Stock  | US     | Jan 2nd 1998 - Jan 31st 2003 | 1276           | 25      |
| MSCI     | Index  | Global | Sept 9th 2005 - Sept 7th 2009 | 1042           | 3       |

endorses the effectiveness of the technical analysis approach. Moreover, it should be noted that the success of CORN depends on three basic assumptions that form the basis of most technical analysis methods, including (1) market action discounts everything, that is, technical analysis assumes stock price at any given time reflects everything that has/could affect the company including fundamental factors; (2) price moves in trends, and (3) history tends to repeat itself. These assumptions allow us to construct a portfolio using only similar trends of historical market prices, without caring about other factors.

5. EXPERIMENTS

5.1. Experimental Testbed on Real Data

In our experiments, we perform numerical evaluations on four real datasets by comparing the proposed CORN algorithm with a number of competing learning-to-trade algorithms. The information of the four datasets is summarized in Table I.

The first dataset is the NYSE dataset, which has been widely used in many previous studies [Cover 1991; Helmbold et al. 1998; Borodin et al. 2004; Agarwal et al. 2006; Györfi et al. 2006; 2008]. It contains 5651 daily price relatives of 36 stocks in the New York Stock Exchange (NYSE) for a 22-year period from July 3rd 1962 to December 31st 1984. In our experiments, we refer to it as “NYSE (O)”. To demonstrate current relevance, we also collected newer data from the NYSE market from January 1st 1985 to June 30th 2009, which contains 6179 trading days. We refer to this dataset as “NYSE (N)”. It is worth noting that this dataset consists of 23 stocks rather than the previous 36 stocks owing to the amalgamation and bankruptcy of some previous stocks. All price relatives are adjusted for splits and dividends, which is consistent with the previous NYSE (O) dataset. The third dataset is the SP500 dataset used by Borodin et al. [2004]. It consists of 25 stocks from S&P500 which have the largest market capitals. This dataset contains price relatives of 1276 trading days, ranging from January 2nd 1998 to January 31st 2003. The fourth dataset, MSCI, is a collection of global equity indices. It contains three indices which represent the equity markets of Pacific, North America, and Europe, ranging from September 9th 2005 to September 7th 2009 with a total of 1042 trading days.

The diverse datasets in our testbed have witnessed several cycles of the stock markets, especially during the dot-com bubble from 1995 to 2000 and the subprime mortgage crisis from 2007 to 2009. The first three datasets are used to test the capability of the CORN on stock markets while the fourth dataset is used to test the capability of the CORN on global indices which may be potentially applicable for “Fund on Fund” (FOF). Note that although the CORN algorithm is numerically tested on stock markets, it could be applied on any kind of financial market.

1All datasets can be downloaded from http://www.cais.ntu.edu.sg/~libin/portfolios.
2We collected the data from Yahoo finance. http://finance.yahoo.com.
3We collected the data from MSCI Barra. http://www.mscibarra.com.
5.2. Experimental Setup and Metrics
There are two possible parameters that affect the performance, that is, the correlation coefficient threshold $\rho$ and the maximum window size $W$. In our experiment, we simply fix $\rho = 0.1$ for the CORN-U algorithm without any tuning, which is not the best parameter as shown in our subsequent evaluations. For the CORN-K algorithm, in theory, $W$, $P$, and $K$ in principle can be determined from the data. In practice, due to computational concerns, we simply fix $W = 5$, $P = 10$, and $K = 5$ in all experiments. We will later examine the influence of these parameters.

To compare the performances of different learning-to-trade algorithms, we adopt the total wealth, the Annualized Percentage Yields (APYs), and the annualized Sharpe Ratio. In general, the higher the values of these measures, the better the performance of the learning-to-trade algorithm. In addition, we also adopt the Maximum DrawDown (MDD) for the drawdown analysis of the learning-to-trade strategy. The smaller the MDD value, the more preferable the trading algorithm concerning downside risk.

5.3. Approaches Compared
We implemented two variants of the proposed CORN strategy as well as a variety of existing strategies described in Section 3 and listed as follows.4

(1) Market. Market strategy (the uniform BAH approach);
(2) Best-Stock. Best stock in the market that is a hindsight strategy;
(3) BCRP. Best Constant Rebalanced Portfolios strategy in hindsight;
(4) UP. Cover’s Universal Portfolios implemented as Kalai and Vempala [2002], its parameters are set to $\delta_0 = 0.004$, $\delta = 0.005$, $m = 100$, $S = 500$;
(5) EG. Exponential Gradient (EG($\eta$)) algorithm with the best parameter $\eta$ fixed to 0.05 as suggested by Helmbold et al. [1998];
(6) ONS. Online Newton Step (ONS($\eta$, $\beta$, $\gamma$)) with the best parameters set as the same suggested by Agarwal et al. [2006], that is, $\eta = 0$, $\beta = 1$, $\gamma = 1/8$;
(7) Anticor. BAH$_{30}$ (Anticor) as a variant of Anticor to smooth the volatility, which is a better solution proposed by Borodin et al. [2004];
(8) $B^K$. Nonparametric kernel-based moving window ($B^K(c)$) strategy with the parameter setting $W = 5$, $L = 10$, $c = 1.0$ that has the best empirical performance according to Györfi et al. [2006];
(9) $B^{NN}$. Nonparametric nearest-neighbor-based strategy with parameter $W = 5$, $L = 10$, $p_\ell = 0.02 + 0.5 \frac{\ell - 1}{L - 1}$ suggested by Györfi et al. [2008].

5.4. Experiment 1: Evaluation of Total Wealth
The first experiment evaluates the total wealth achieved by different learning-to-trade algorithms without considering transaction cost, which will be investigated in Section 5.12. For each algorithm, we invest an initial asset $S_0 = \$1$ over all the stocks in the market.

Table II summarizes the total wealth achieved by various algorithms on the four datasets. Several observations can be drawn from the results. First of all, we find that all learning-to-trade algorithms can beat the market index, that is, the uniform BAH strategy, on all the datasets. This shows that it is promising to investigate learning-to-trade algorithms for portfolio selection. Second, most existing trading algorithms, except Anticor, do not always outperform the best stock in the market on the four datasets. Third, we observe that the regular follow-the-leader approaches (UP, EG, ONS) often perform substantially worse than other state-of-the-art approaches.

4We can adjust the parameters of competitors for better performance, but that is beyond the scope of this article.
Table II. Total Wealth Achieved by Various Strategies on Four Real Datasets

| Strategies | NYSE (O)  | NYSE (N)  | SP500  | MSCI  |
|------------|-----------|-----------|--------|-------|
| Market     | 14.50     | 14.84     | 1.34   | 0.92  |
| Best-stock | 54.14     | 63.47     | 3.78   | 0.97  |
| BCRP       | 250.60    | 93.25     | 4.07   | 0.99  |
| UP         | 27.41     | 24.76     | 1.64   | 0.97  |
| EG         | 27.09     | 24.14     | 1.63   | 0.97  |
| ONS        | 109.19    | 23.65     | 3.34   | 1.09  |
| Anticor    | 1.71E+07  | 7.37E+04  | 5.55   | 2.45  |
| $B^K$      | 1.08E+09  | 9.5E+02   | 2.26   | 1.27  |
| $B^{KN}$   | 3.35E+11  | 5.59E+04  | 3.09   | 37.43 |
| CORN-U     | 1.48E+13  | 3.32E+05  | 6.35   | 31.51 |
| CORN-K     | 6.29E+13  | 4.38E+05  | 8.56   | 48.72 |

The numbers in boldface represent the top two achievements on each dataset.

Finally, among all compared algorithms, the proposed CORN-U and CORN-K algorithms always achieve the best total wealth on all datasets, and are substantially better than the market index and the best stock in the market. For example, on the NYSE (O) dataset after trading for 22 years, the total wealth achieved by the CORN-U strategy and the CORN-K strategy impressively increases from $1 to almost $14.5-trillion and $63-trillion, respectively, which are much higher than the state-of-the-art $B^{NN}$ algorithm that achieves $335-billion and the $B^K$ algorithm that achieves $1.16-billion.

On the SP500 and MSCI datasets, due to the tough market conditions (the market index of MSCI dataset actually decreases) and relatively shorter trading period, we find that the total wealth achieved by the learning-to-trade strategies is significantly smaller than that of the two NYSE datasets. But, we also observe that both CORN-U and CORN-K still achieve considerably better results than the market index, the best stock in the market, as well as all the state-of-the-art strategies.

Besides the preceding results, we are also interested in examining how the total wealth achieved by various strategies change over different trading periods. Figure 6 shows the changes of total wealth achieved by the various strategies on the four datasets. From the figure, we first observe that the two CORN algorithms consistently outperform other algorithms over most trading periods. Further, we find that when more trading days are engaged, the growth rate of the wealth achieved by CORN tends to increase, which is particularly obvious on NYSE (O) and NYSE (N) datasets where the growth rate after 2500 trading days is much higher than the previous trading periods. This phenomenon establishes that when more historical data are available for the learning-to-trade task, the CORN algorithms are able to perform more effective trading by exploiting statistical correlation with the powerful nonparametric learning approach. It is also interesting to note that, although the market drops sharply due to the financial downturn in 2008, the proposed CORN algorithms are still able to achieve encouraging returns, which is especially more impressive in the later part of the MSCI dataset. All these impressive results reiterate the efficacy and robustness of the proposed learning-to-trade algorithm.

5.5. Experiment 2: Evaluation of APY, Risk and Sharpe Ratio

In this experiment, we evaluate the performance of APYs, risks, and annualized Sharpe Ratios of the compared strategies and summarize the results in Table III. For each cell in the table, the two numbers of the first row represent APY and risk (volatility risk), respectively, and the number of the second row represents the annualized Sharpe Ratio. For example, on the NYSE (O) dataset, the market strategy’s APY is 13%, risk or annualized standard deviation of daily return is 15%, and annualized Sharpe Ratio...
is 60%. From Table III, we observe that on the NYSE (O), NYSE (N), and SP500 datasets, both CORN algorithms achieved the highest APYs and annualized Sharpe Ratios among all learning-to-trade strategies. On the MSCI dataset, the CORN-K strategy achieved the highest APY value and the highest annualized Sharpe Ratio while the CORN-U strategy is also as excellent as the other state-of-the-art strategies. Similar to the common fact of no pain no gain in financial markets, that is, a higher return is often associated with a higher risk, the risk of our CORN algorithm is also higher than other strategies since the return of the proposed algorithms are much higher than the others. Nonetheless, the impressive annualized Sharpe Ratios achieved by CORN strongly support the advantages of the proposed trading strategy.

5.6. Experiment 3: Evaluation of Quarterly or Monthly Returns

We are also interested in whether the proposed CORN strategies outperform the benchmark quarterly or monthly. Empirically, Figure 7 shows the quarterly return distributions of the CORN-U strategy on NYSE (O) and NYSE (N) datasets and monthly return distributions of the CORN-U strategy on SP500 and MSCI datasets. For comparison, the corresponding market return distributions are shown in the figure as the benchmark. In Figure 7(a) and Figure 7(b), most of the quarterly returns with the CORN-U strategy on NYSE (O) and NYSE (N) datasets are higher than the quarterly returns of market index. More specifically, during 72 quarters out of total 86 quarters
Table III. APYs, Risks and Sharpe Ratios for Various Strategies on the Four Datasets

| Strategies | NYSE (O) | NYSE (N) | SP500 | MSCI |
|------------|---------|---------|-------|------|
| Market     | 13% ± 15% | 12% ± 18% | 8% ± 24% | −2% ± 20% |
|            | 60%      | 43%     | 52%   | −31% |
| Best-stock | 20% ± 24% | 18% ± 29% | 30% ± 51% | −1% ± 25% |
|            | 65%      | 50%     | 52%   | −19% |
| BCRP       | 29% ± 31% | 20% ± 24% | 32% ± 42% | 0% ± 22% |
|            | 80%      | 69%     | 67%   | −20% |
| UP         | 16% ± 14% | 13% ± 19% | 16% ± 22% | 0% ± 20% |
|            | 91%      | 51%     | 28%   | −24% |
| EG         | 16% ± 13% | 14% ± 19% | 11% ± 22% | −1% ± 20% |
|            | 91%      | 52%     | 30%   | −24% |
| ONS        | 24% ± 18% | 14% ± 34% | 27% ± 24% | 2% ± 20% |
|            | 110%     | 28%     | 98%   | −8% |
| Anticor    | 113% ± 29% | 58% ± 33% | 41% ± 38% | 25% ± 21% |
|            | 378%     | 163%    | 97%   | 100% |
| B^k        | 158% ± 36% | 32% ± 25% | 17% ± 33% | 6% ± 20% |
|            | 422%     | 113%    | 40%   | 10% |
| B^NN       | 234% ± 40% | 56% ± 27% | 25% ± 39% | 147% ± 25% |
|            | 511%     | 189%    | 55%   | 580% |
| CORN-U     | 297% ± 49% | 68% ± 33% | 45% ± 41% | 137% ± 25% |
|            | 609%     | 194%    | 97%   | 533% |
| CORN-K     | 324% ± 52% | 70% ± 32% | 54% ± 40% | 164% ± 27% |
|            | 619%     | 204%    | 123%  | 602% |

The upper row of each cell shows APY ± Risk and the second row shows Sharpe Ratio. The top two ratios on each dataset are highlighted.

(85%) for the NYSE (O) dataset, 76 out of 94 quarters (81%) for NYSE (N) dataset, the CORN-U strategy outperforms the market. In the Figure 7(c), during 37 months out of 58 months (64%), the CORN-U strategy outperforms the market. In the Figure 7(d), during 46 out of 48 months (96%), the CORN-U strategy outperforms the market index. As a summary, for most of the time slices, the CORN-U strategy could outperform the market strategy, which again verifies the proposed CORN is stable and robust.

5.7. Statistical Evaluation of Performance

Besides the preceding results, we are also interested in evaluating the CORN strategy statistically [Katz and McCormick 2000]. Since our datasets are just samples for the entire stock market population, we try to validate the strategy for the future. We conduct a student t-test to determine the likelihood that the observed profitability is due to chance alone (under the assumption that the system was not profitable in the population from which our datasets were drawn). Since the sample profitability of the proposed CORN is being compared with no profitability, 0 is subtracted from the sample mean profit/loss. It is worth noting that daily profit/loss equals daily return minus 1. The standard error of the mean is calculated as the standard deviation divided by the square root of the number of trading days. The \( t \)-statistic is the sample profit mean divided by the sample standard error to obtain the value of the \( t \)-statistic, that is, \( t \)-statistic = \( \frac{\text{sample profit mean} - 0}{\text{sample standard error}} \). Finally, the probability of getting the \( t \)-statistic by chance alone is calculated with the degree of freedom, which is the number of trading days minus 1. It is worth noting that the assumption of the student t-test is that the underlying distribution of the data is normal. According to the central limit theorem, as the number of cases in the sample increases, the distribution of the sample mean approaches normal. Considering that each of our datasets contains a large number of trading samples, we can regard the distribution of the profit/loss as normal, so the statistical analysis regarding the mean is meaningful.
Fig. 7. Quarterly return on NYSE (O) & NYSE (N) and monthly return on SP500 & MSCI. CORN-U beats the market in 85%, 81%, 64%, and 96% periods on the NYSE (O), NYSE (N), SP500, and MSCI datasets, respectively.

Table IV. Statistical Analysis of Mean Profit/Loss for CORN-U Strategy

| Statistical Attributes | NYSE (O) | NYSE (N) | SP500 | MSCI |
|-------------------------|----------|----------|-------|------|
| Size                    | 5651     | 6179     | 1276  | 1042 |
| Mean                    | 0.0062   | 0.0023   | 0.0020| 0.0039|
| SD                      | 0.0325   | 0.0204   | 0.0255| 0.0168|
| SE of the mean          | 4.32E-04 | 2.59E-04 | 7.13E-04| 5.20E-04|
| t-Statistic (P/L > 0)   | 14.3395  | 8.8730   | 2.8046| 7.5031|
| P-value (Significance)  | 0        | 0        | 0.0026| 0    |

The statistical analysis is on daily basis to test whether the success of CORN is due to chance.

Table IV summarizes the statistical analysis for the result of mean profit/loss achieved by the CORN-U algorithm. Since our strategy dynamically changes the portfolio every trading day, we analyze it on a daily basis. From the table, we can see that the t-statistics for the four datasets are so large that the significance in each case approaches zero. The results show that it is almost impossible to attribute the success of the CORN-U strategy to chance alone.
5.8. Experiment 4: Evaluation of Drawdown

In finance, drawdown analysis measures the decline from a historical peak of the total wealth achieved (refer to Section 2.2 for details). This section shows that the drawdown for the proposed CORN strategy is acceptable.

Figure 8 shows the drawdown analysis on the four datasets. For comparison, the maximum drawdown for market strategy, best-stock strategy, and state-of-the-art \( \text{BNN} \) strategy are also presented. From the table, we can conclude that the maximum drawdowns for the proposed CORN strategies, especially the CORN-K strategy, are quite impressive. The CORN-K strategy achieves the lowest maximum drawdowns on each dataset. It is worth noting that during the financial crisis from 2007 to 2009, there is a huge drawdown on the MSCI dataset, for example, the MDD for market strategy is 59.17%. However, the CORN strategies with MDDs of 18.77% and 14.91%, respectively, still perform much better than the market. Since drawdown is an important measure of the downside risk, this drawdown analysis strongly demonstrates that the risk of the proposed CORN strategies is acceptable even though we designed the strategy solely on the utility function of total wealth.

5.9. Experiment 5: Evaluation of Parameters

Following the intuitive analysis in Section 4.5, we experimentally evaluated the effects of the two parameters, that is, correlation coefficient threshold \( \rho \) and maximum window size \( W \).

To evaluate the effect of correlation coefficient threshold \( \rho \), we analyze the performance of the CORN algorithm by varying parameter \( \rho \) from \(-1.0\) to \(+1.0\) with fixed \( W = 5 \). Figure 9 shows the effects of varied threshold values for the CORN-U algorithm on the four datasets. Several observations can be drawn from the empirical results. First of all, the results verify the statement that CORN reduces to UCRP when \( \rho \) approaches to 1. Further, the empirically optimal value of \( \rho \) is 0.3 for NYSE (N) dataset, while results on other datasets show the optimal values of \( \rho \) lie between 0 to 0.5. Finally, we found that CORN performs considerably poorly when \( \rho \) is too large (excluding many informative cases) or too small (including too many irrelevant cases), which is consistent with our motivation.
Another important parameter is the maximum window size, $W$. We notice that different $W$s may affect the performance of the proposed CORN algorithm. This experiment aims to examine the effect of varied $W$s with fixed $\rho = 0.1$. Figure 10 shows the evaluation results of varied $W$ ranging from 2 to 15 on the four datasets with the CORN-U algorithm. We have several observations from the empirical results. First of all, we found that window size does affect the performance of the proposed algorithm. Second, we nonetheless do not see any consistent trend from the figure, which is consistent with the analysis in Section 4.5. Finally, we found that on all cases of various $W$s (from 2 to 15), the proposed algorithm always outperforms the best stock and market.

As both CORN-U and CORN-K algorithms combine the experts, the final performance are affected by experts' individual performances. We conduct experiments to further examine the proportion of contribution made by these experts, which are based on Algorithm 1 with a maximum window size $W = 10$. We then rank the experts with respect to their performances, and show their corresponding proportions. Figure 11 illustrates the results on the four datasets. It is clear that the proportions of contribution made by the experts are different. But their performances generally fall in the normal range, that is, the majority of contributions range from 5% to 30%, which shows that the predictions of all these experts are rather robust. In reality, it is possible that certain experts may give very bad predictions, leading to very small contributions. Our proposed combination methods, however, would reduce the impact of such bad predictions. Finally, as the distribution of CORN-U is uniform over all the experts while the
Fig. 10. Effect of window size on the total wealth achieved. With correlation coefficient threshold fixed to 0.1, the maximal window size ranges from 2 to 15 consecutively.

Fig. 11. Proportion of contribution to the final performance made by a set of 10 experts.
Table V. Risk Adjusted Return (Annualized Sharpe Ratio) Achieved by Various Strategies without and with Margin

| Dataset | Market | BCRP | Anticor | B<sup>NN</sup> | CORN-U | CORN-K |
|---------|--------|------|---------|---------------|--------|--------|
| NYSE (O) | 0.60   | 0.80 | 3.78    | 5.71          | 6.00   | 6.19   |
|         | 0.36   | 0.93 | 4.89    | 9.74          | 9.95   | 12.17  |
| NYSE (N) | 0.43   | 0.69 | 1.63    | 1.89          | 1.94   | 2.94   |
|         | 0.25   | 0.64 | 1.39    | 1.81          | 2.00   | 2.16   |
| SP500   | 0.08   | 0.67 | 0.97    | 0.55          | 0.98   | 1.23   |
|         | 0.09   | 0.58 | 0.82    | 0.31          | 0.85   | 1.36   |
| MSCI    | -0.31  | -0.20| 1.00    | 5.80          | 5.33   | 6.02   |
|         | -0.44  | -0.20| 0.78    | 11.07         | 10.60  | 12.80  |

The number in the upper row shows results without margin, while the number in the lower row shows results with margin.

Table VI. Effect of Randomizing the Initial Trading Day and Trading Period on the Performance of CORN, with Respect to the Market Strategy and the Best Stock

| Dataset | Avg. Length (All cases) | Against the Market | Against the Best Stock |
|---------|-------------------------|--------------------|------------------------|
|         |                         | Win Rate | Avg. Length | Win Rate | Avg. Length | Win Rate | Avg. Length |
| NYSE (O) | 1464                | 94%       | 678        | 1513     | 76%        | 432      | 1789       |
| NYSE (N) | 1652                | 87%       | 338        | 1848     | 66%        | 461      | 2265       |
| SP500   | 294                 | 74%       | 17         | 391      | 21%        | 217      | 582        |
| MSCI    | 289                 | 98%       | 31         | 298      | 91%        | 26       | 319        |

"Win Rate" denotes the percentage of cases when CORN-U beats the baselines, and "Avg. Length" denotes the average length of the trading period that CORN-U is run, in each case.

distribution of CORN-K is uniform over the best K experts, the final result of CORN-K would be consistently better than that of CORN-U under the same parameters, as observed in Table II.

5.10. Experiment 6: Evaluation of Portfolio with Margin Buying

We also tested the proposed method on the cases where we are allowed to buy stocks with margin based on the margin model described in Section 2.3. Table V shows the risk adjusted returns of the proposed CORN strategies without and with margin buying. For the sake of comparison, we list the results of market strategy, BCRP, state-of-the-art Anticor strategy, and B<sup>NN</sup> strategy. It is shown that with the benefit of margin, in most cases, except CORN-U on SP500 dataset, the annualized Sharpe Ratios achieved by the proposed CORN algorithms increase (on 3 out of 4 datasets for CORN-U and all datasets for CORN-K), while the ratios achieved by other strategies often decrease (on 3 out of 4 datasets for Anticor algorithm and on 2 out of 4 datasets for B<sup>NN</sup>). The same as the results without margin, the proposed CORN algorithms with margin achieve the highest annualized Sharpe Ratios. The experiment again indicates that CORN is an effective and practical approach. It can take the advantage of margin and gain return growth corresponding to the associated risk.

5.11. Experiment 7: Portfolio with Random Periods

We shall now demonstrate the robustness of the CORN strategy by eliminating the impact of specific trading periods. We randomly choose the initial trading dates and the duration of the trading period, and run CORN-U for selecting portfolios. The quality of 100 portfolios generated from such random periods on each of the four datasets is summarized in Table VI. The table shows the percentage of cases ("Win Rate") where CORN-U beats the baselines, and the average duration or length of the trading period to create the portfolio ("Avg. Length") when CORN-U wins or loses. The results clearly
show that on three out of four datasets (NYSE (O), NYSE (N), and MSCI), portfolios generated by CORN-U from random periods outperform the market index and the best stock with a high probability. The win rate for CORN-U on the SP500 dataset with respect to the best stock is not quite satisfying because the best stock in SP500 is so strong that it beats most competitors, as shown in Table II. By observing sample length statistics, we can find that when the portfolio generated by CORN loses, the length of the trading periods (Avg. Size (Loss)) are relatively short. For example, on the NYSE (O) dataset, the average sample trading length is 1464, while the average trading length when CORN loses to the market is relatively short – 678, and the average sample trading length when CORN wins against the market is relatively long – 1513. Similar statistics appear over the four datasets. These results are consistent with the nonparametric learning process of CORN strategies, that is, the more historical price relatives for learning, the more effective the proposed CORN strategies.

5.12. Experiment 8: Evaluation of Transaction Cost

Another important and unavoidable issue in portfolio selection is transaction cost. In the experiments, we adopt the proportional transaction cost model stated in Section 2.3. We conduct the experiments on both situations with and without transaction cost. In particular, we evaluate the performances of the proposed CORN algorithm by varying transaction cost $\gamma$ from 0% to 1.0% on the four datasets. It would be interesting if CORN can still outperform the two baselines, that is, the market and the best stock, in the market when there is a nontrivial transaction cost. Figure 12 shows our experimental results based on the CORN-U strategy. As we can observe, when the transaction cost increases, the total wealth achieved by CORN-U drops considerably. However, we found that on the four datasets, even with a rather high transaction cost, the CORN-U strategy still performs convincingly well. The proposed CORN strategy is rather robust on all except the SP500 dataset. The break-even commission rates of the total wealth achieved between the CORN and the two baselines on NYSE (O), NYSE (N), and MSCI datasets range from 0.3% to 0.6%, which are significantly impressive. On the other hand, on the SP500 dataset, the break-even rate between CORN and the best stock is about 0.08%, and the break-even rate between CORN and the market is about 0.18%. Even though this is not as impressive as on the other three datasets, such break-even rates are still acceptable in practice. As previously seen, the reason for such results is that the best stock on SP500 is simply so strong that it actually beats most existing methods, as shown in Table II.

5.13. Experiment 9: Evaluation of Computation Time

The following experiment evaluates the time efficiency of the proposed CORN strategy. In general, the CORN strategy is quite computationally intensive. To be specific, the major computation time costs are twofold: (1) the time cost for the selection of correlation similar set; (2) the time cost required for performing the optimization process. We run all of our experiments on a desktop PC equipped with Intel Core 2 at 2.33 GHz at MATLAB. On the NYSE (O) dataset, CORN-U and CORN-K strategies took about 4 hours and 25 hours, respectively, for all the trading periods, while the other two nonparametric learning algorithms $B^K$ and $B^{NN}$ took 26 hours and 14 hours on the same dataset, respectively. On the MSCI dataset consisting of 3 equity indices in 1042 trading days, CORN-U and CORN-K took about 15 minutes and 50 minutes, respectively, while $B^K$ and $B^{NN}$ took 20 minutes and 75 minutes, respectively. These results show that our method is computationally comparable with the previous two state-of-the-art algorithms.
Total wealth achieved by CORN-U on the four datasets with proportional commission rate $\gamma$ varying from 0%, 0.1%, ..., 1.0%. Among the four datasets, the effect of transaction costs on NYSE (O), NYSE (N), and MSCI datasets are log-scaled for proper display. The break-even commission rates with the market index are about 0.6%, 0.3%, 0.2%, and 0.5% for NYSE (O), NYSE (N), SP500, and MSCI datasets, respectively.

5.14. Experiment 10: Comparison to Well-Known Time-Series Methods

As there are some well-studied time series prediction methods, it would be interesting to compare CORN against these methods. In this section, we compare CORN against the well-known ARMA and GARCH methods for time-series predictions on the stock datasets. Since they were not specifically proposed for portfolio selection tasks, we sequentially make predictions for each stock and select the stock with the highest prediction, that is, we put all the money in the best stock based on the prediction results. We choose parameters for ARMA(p, q) according to the previous work proposed by Biau et al. [2010], that is, set (p, q)=(1, 1). Similarly, we set parameters for GARCH(p, q) to the default values, that is, (p, q)=(1, 1). Unlike the previous experiments, since our goal is to evaluate the prediction performance, traditional performance measures cannot be used. Thus, we should consider some different measures in this experiment. As we typically care about the profitability of the daily return with respect to the market strategy, we compare its daily performance with that of the market strategy. This produces two criteria for performance measures. The first one represents the accuracy of the profitability: percentage of the days for which the strategy surpasses the market strategy. The second criterion denotes the strength of the profitability: average ratio of the daily wealth gained by the strategy over that achieved by the market.
Table VII. Comparison of the Proposed CORN Strategy against Two Time Series Prediction Methods (ARMA & GARCH)

| Dataset | ARMA | GARCH | CORN-U | CORN-K |
|---------|------|-------|--------|--------|
| NYSE (O) | 47.67% | 46.82% | 53.78% | 54.04% |
|         | 1.0002 | 1.0002 | 1.0053 | 1.0056 |
| NYSE (N) | 48.11% | 49.44% | 52.82% | 53.52% |
|         | 0.9999 | 1.0000 | 1.0018 | 1.0019 |
| SP500   | 47.85% | 49.69% | 52.90% | 52.82% |
|         | 0.9992 | 1.0001 | 1.0014 | 1.0016 |
| MSCI    | 49.81% | 51.73% | 64.01% | 63.63% |
|         | 0.9997 | 1.0001 | 1.0034 | 1.0039 |

For each dataset, the first row denotes the accuracy, and the second row is the strength of profitability.

strategy. In practice, the first criterion denotes the chance how likely the prediction-based strategy is able to produce profit better than the market strategy, and the second criterion denotes the ratio between the profit produced by the prediction strategy over that of the market. The higher the values of these criteria, the better performance the algorithm achieves on the sequential time-series prediction tasks.

Table VII shows the results, which clearly indicate CORN significantly beats the well-known time-series prediction algorithms, namely, ARMA and GARCH. For the first criterion, the prediction accuracy of CORN significantly surpasses that of ARMA and GARCH. In particular, all accuracies produced by ARMA are below 50%, and those produced by the GARCH strategy fluctuate around 50%, while the results obtained by CORN are above 50%. This clearly shows that CORN performs significantly better than the traditional widely used ARMA and GARCH time-series prediction techniques. For the second criterion, the strengths of the probability are always below 1 for ARMA, and those of GARCH are always floating around 1, which is consistent with the previous result that the accuracies are below 50%. On the other hand, the results of CORN are always above 1, significantly surpassing ARMA and GARCH strategies. This experiment validates the applicability and capability of CORN to the sequential time-series prediction problem.

5.15. Discussion

From the extensive experiments detailed earlier, CORN has been empirically proved an effective tool for portfolio selection, which exploits the statistical correlation information in the financial markets by a nonparametric learning approach. The success of the CORN strategy may be attributed to some hidden information in the market, which has not yet been explored by market traders. Our method using statistical correlation can exploit this hidden information, leading to encouraging performance. Although we cannot provide the exact hidden information, the success of our method does provide certain useful knowledge to enhance our understanding of portfolio selection and the stock market in financial engineering. In particular, one useful finding is that our promising result provides strong evidence to show that the market is inefficient, as explained in Section 1. Another useful piece of knowledge is that the price does often move in trends and the price relative patterns could reappear in practice. Such knowledge provides evidence to endorse the advantages of technical analysis in the long-standing debate, and indicates that it may be possible to exploit such knowledge and hidden information to build effective portfolios in real-world finance applications.

6. CONCLUSION

This article proposed a novel CORrelation-driven Nonparametric learning (CORN) strategy for portfolio selection, which effectively exploits the statistical correlation
information hidden in underlying stock market movements, and benefits from the exploration of powerful nonparametric learning techniques. The proposed CORN algorithm is simple in nature, easy to implement, and has practically very few parameters which are easy to set. Our empirical studies show that the CORN algorithm can substantially beat the market and the best stock, and also consistently surpasses a variety of state-of-the-art algorithms.

Although high return strategies are often associated with high risk, it would be more attractive to develop a strategy that can manage the risk properly without slashing the return too much. As an extension to this work, we are currently developing such risk-limiting strategies for CORN. Moreover, we are also looking at exploiting transaction volume information, which could be potentially beneficial to improve trading performance. In future, we plan to investigate theoretical insights of the algorithm, and examine the extensions of our algorithm to improve the performance with high transaction costs.

REFERENCES

AGARWAL, A., HAZAN, E., KALE, S., AND SCHAPIRE, R. E. 2006. Algorithms for portfolio management based on the newton method. In Proceedings of the International Conference on Machine Learning.

AHA, D. W. 1991. Case-Based learning algorithms. In Proceedings of the DARPA Case-Based Reasoning Workshop.

AHA, D. W., KIBLER, D., AND ALBERT, M. K. 1991. Instance-Based learning algorithms. Mach. Learn. 6, 1, 37–66.

ALGROET, P. H. AND COVER, T. M. 1988. Asymptotic optimality and asymptotic equipartition properties of log-optimimum investment. The Ann. Probab. 16, 2, 876–898.

BELENTEPE, C. Y. 2005. A statistical view of universal portfolios. Ph.D. thesis, University of Pennsylvania.

BLUM, G., BLEAKLEY, K., GYÖRFI, L., AND OPTUCSÁK, G. 2010. Nonparametric sequential prediction of time series. J. Nonparam. Statist. 22, 3, 297–317.

BLUM, A. AND KALAI, A. 1999. Universal portfolios with and without transaction costs. Mach. Learn. 35, 3, 193–205.

BOLLERSLEV, T. 1986. Generalized autoregressive conditional heteroskedasticity. J. Economet. 31, 3, 307–327.

BORODIN, A., EL-YANIV, R., AND GOGAN, V. 2004. Can we learn to beat the best stock. J. Artif. Intell. Res. 21, 579–594.

BOX, G. E. P., AND JENKINS, G. M. 1970. Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco, CA.

BREIMAN, L. 1961. Optimal gambling systems for favorable games. In Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability.

CHEKASSKY, V. S. AND MULLER, F. 1998. Learning From Data: Concepts, Theory, and Methods. John Wiley & Sons, New York.

COVER, T. M. 1991. Universal portfolios. Math. Finan. 1, 1, 1–29.

COVER, T. M. AND OBERDJICH, E. 1996. Universal portfolios with side information. IEEE Trans. Inf. Theory 42, 2, 348–363.

EDWARDS, R. D. AND MAGRE, J. 1948. Technical Analysis of Stock Trends. Stock Trend Service, Springfield, MA.

ENGLE, R. F. 1982. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. Econometrica 50, 4, 987–1007.

FAMA, E. F. 1970. Efficient capital markets: A review of theory and empirical work. J. Finan. 25, 2, 383–417.

GRAHAM, B. AND DODD, D. 1934. Security Analysis. McGraw Hill, New York.

GYÖRFI, L., LUGOSI, G., AND UDINA, F. 2006. Nonparametric kernel-based sequential investment strategies. Math. Finan. 16, 2, 337–357.

GYÖRFI, L. AND SCHAPIRE, R. 2003. Nonparametric prediction. In Advances in Learning Theory: Methods, Models and Applications, J. A. K. Suykens, G. Horvatth, S. Basu, C. Micchelli, and J. Vandewalle, Eds. IOS Press, NATO Science Series, Amsterdam, Netherlands, 339–354.

GYÖRFI, L., UDINA, F., AND WALK, H. 2008. Nonparametric nearest neighbor based empirical portfolio selection strategies. Statist. Decis. 26, 2, 145–157.

GYÖRFI, L., URBÁN, A., AND VAJDA, I. 2007. Kernel-based semi-log-optimal empirical portfolio selection strategies. Int. J. Theor App. Finan. 10, 3, 505–516.
HAZAN, E. 2006. Efficient algorithms for online convex optimization and their applications. Ph.D. thesis, Princeton University.

HAZAN, E. AND SESHADRI, C. 2009. Efficient learning algorithms for changing environments. In Proceedings of the International Conference on Machine Learning.

HELMBOLD, D. P.; SCHAPIRE, R. E.; SINGER, Y.; AND WARMUTH, M. K. 1998. On-Line portfolio selection using multiplicative updates. Math. Finan. 8, 4, 325–347.

KALAI, A. AND VEMPALA, S. 2002. Efficient algorithms for universal portfolios. J. Mach. Learn. Res. 3, 423–440.

KATZ, J. O. AND MCCORMICK, D. L. 2000. The Encyclopedia of Trading Strategies. McGraw Hill, New York.

MARKOWITZ, H. 1952. Portfolio selection. J. Finan. 7, 1, 77–91.

ORDENTLICH, E. AND COVER, T. M. 1996. On-line portfolio selection. In Proceedings of the Annual Conference on Computational Learning Theory.

RODGERS, J. L. AND NICEWANDER, A. W. 1988. Thirteen ways to look at the correlation coefficient. Amer. Statist. 42, 1, 59–66.

SHARPE, W. F. 1966. Mutual fund performance. J. Bus. 39, 1, 119.

TSAY, R. S. 2002. Analysis of Financial Time Series. John Wiley & Sons, New York.

Received April 2010; revised July 2010, October 2010; accepted October 2010