4-2011

CORN: Correlation-driven Nonparametric Learning Approach for Portfolio Selection

Bin LI
Nanyang Technological University

Steven C. H. HOI
Singapore Management University, CHHOI@smu.edu.sg

Vivekanand GOPALKRISHNAN
Nanyang Technological University

Citation

CORN : Correlation-driven Nonparametric Learning Approach for Portfolio Selection

BIN LI
STEVEN C.H. HOI
VIVEKANAND GOPALKRISHNAN
School of Computer Engineering, Nanyang Technological University

Machine learning techniques have been adopted to select portfolios from financial markets in some emerging intelligent business applications. In this paper, we propose a novel learning to trade algorithm termed the **COR**relation-driven **N**onparametric learning strategy (CORN) for actively trading stocks, which 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, in which the encouraging results show that the proposed new algorithm can easily beat both the market index and the best stock in the market substantially (without or with small transaction costs), and also surpasses a variety of state-of-the-art techniques significantly.

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

General Terms: Design,Algorithms,Economics,Experimentation

Additional Key Words and Phrases: Online Portfolio Selection,Nonparametric Learning,Correlation Coefficient

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 some relevant 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 interests from the machine learning, data mining, and artificial intelligence communities. In this paper, 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 of the 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 1996] 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 et al. 2007] believe that the historical performance of stocks and markets 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; Györfi et al. 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; Györfi et al. 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 a simple heuristic investment strategy 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 paper, we present a novel learning to trade strategy for the (sequential) portfolio selection problem, termed the **COR**relation-driven **N**onparametric learning (CORN) algorithm. In particular, CORN seeks to locate the market windows that are similar to the latest market window, and makes a log-optimum portfolio according to the idea of the best Constant Rebalanced Portfolio strategy [Cover and Gluss 1986]. 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 show that CORN easily beats the market as well as the best stock in the market substantially (without or with reasonable transaction costs), and also consistently surpasses a variety of state-of-the-art techniques. Even when faced with the recent financial turmoil, the proposed CORN strategy still achieves an excellent performance, which is considerably better than the performance of other existing approaches.

Besides, our promising empirical results also provide a strong evidence to rebut the well-known Efficient Market Hypothesis (EMH) [Timmermann and Granger 2004] in finance theory, which states that the markets are informatively efficient, i.e., prices of assets traded on the markets reflect all known information. EMH asserts that no investors can consistently beat the market using information that is already known. However, 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 paper is organized as follows. Section 2 formally formulates 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 paper and provides directions for future work.

2. PRELIMINARIES: PORTFOLIO SELECTION

In this section, we formulate the portfolio selection problem by following previous studies [Cover 1991; Ordentlich and Cover 1996; Helmbold et al. 1998; Borodin et al. 2004; Györfi et al. 2006].

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 tth trading day, where each element $x_{(t,i)}$ equals the tth closing price of asset i divided by the $(t-1)^{th}$ closing price of asset i, i.e., $x_{(t,i)} = \frac{P_{(t,i)}}{P_{(t-1,i)}}$. Given a window size w, let us define the *market window* for the tth trading day as $\mathbf{X}_{t-w}^{t-1} = (\mathbf{x}_{t-w}, \ldots, \mathbf{x}_{t-1})$, which represents the latest market movement before the tth trading day.

At the beginning of tth 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 ith asset at the beginning of tth trading day. One obvious constraint for a portfolio is that it must be a simplex, denoted by $\mathbf{b}_t \in \triangle_m$, 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 $\mathbf{B}_1^T = (\mathbf{b}_1, \ldots, \mathbf{b}_T)$, which is the output of the *learning to trade* strategy.

Thus, for the tth 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 Tth trading day is,

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

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 for portfolio selection is to learn a portfolio strategy that is expressed as a sequence of functions,

$$\mathbf{b}_t : (\mathbb{R}_+^m)^{t-1} \rightarrow \triangle_m, \quad t = 1, 2, \ldots,$$

where $\mathbf{b}_t(\mathbf{X}_1^{t-1})$ represents the *portfolio* vector made by the investor at the beginning of the tth 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 above, we make several general assumptions for the portfolio selection model:

(1) Transaction cost: no transaction cost exists in the above portfolio selection 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 for portfolio selection.

2.2   Performance Criteria

For portfolio selection, an important issue is to define appropriate criteria for evaluating the performance of the portfolio strategy. One natural and probably the most prominent approach is to adopt some functions of the total wealth achieved by the strategy over the trading period, i.e, $\mathbf{U}(\mathbf{S}_T)$, where $\mathbf{U}(\cdot)$ is some standard economic utility function with respect to the total wealth $\mathbf{S}_T$. Besides, it is possible to adopt some other process-dependent economic utility functions [Moody et al. 1998]. Below we discuss several performance criteria widely used for portfolio selection.

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*) [Elton et al. 2003] that takes account of the compounding effect, i.e.,

$$\mathbf{APY} = (\mathbf{S}_T)^{\frac{1}{y}} - 1, \qquad (2)$$

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.

For a process-dependent investor, an important concern is the evaluation of *risk* and *risk-adjusted return* of the portfolios [Sharpe 1994]. 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) [Elton et al. 2003] 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,

$$\mathbf{SR}_T = \frac{\mathbf{APY} - R_f}{\sigma_p}, \qquad (3)$$

where R_f is the risk-free return (typically the return of Treasury bills, set at 4% in our study), and σ_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 [Magdon-Ismail and Atiya 2004], which measures the decline from a historical peak of the total wealth achieved by a trading strategy. Formally, let $\mathbf{S}(\cdot)$ denote the process of the total wealth achieved by a trading strategy, i.e., $\{\mathbf{S}_1, \ldots, \mathbf{S}_t, \ldots, \mathbf{S}_T\}$. The *drawdown* at any time t, denoted as $\mathbf{DD}(t)$, is defined as: $\mathbf{DD}(t) = \sup \left[0, \sup_{i \in (0,t)} \mathbf{S}(i) - \mathbf{S}(t) \right]$. The *maximum drawdown* (*MDD*) till 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 $\mathbf{MDD}(T)$, is defined as:

$$\mathbf{MDD}(T) = \sup_{\tau \in (0,T)} \left[\sup_{t \in (0,\tau)} \mathbf{S}(t) - \mathbf{S}(\tau) \right]. \tag{4}$$

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 often be taken into consideration. Below 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 proposed in Blum and Kalai [1999] and Borodin et al. [2004]. Specifically, consider a transaction cost rate $\gamma \in (0,1)$, an action of rebalancing the portfolio has to incur transaction cost for both buy and sell operations. At the beginning of the tth trading day, the portfolio manager rebalances the portfolio from the previous closing price adjusted portfolio $\hat{\mathbf{b}}_{t-1}$ to a new portfolio $\mathbf{b}_t$. The transaction cost will be charged according to $\frac{\gamma}{2} \times \sum_i \left| b_{(t,i)} - \hat{b}_{(t-1,i)} \right|$, where the initial portfolio is set to $(0, \ldots, 0)$. Thus, with transaction cost rate γ, the total wealth achieved by the end of the Tth trading day, denoted as $\mathbf{S}_T^{c(\gamma)}$, is expressed as:

$$\mathbf{S}_T^{c(\gamma)} = \mathbf{S}_0 \prod_{t=1}^{T} \left[(\mathbf{b}_t \cdot \mathbf{x}_t) \times \left(1 - \frac{\gamma}{2} \times \sum_i \left| b_{(t,i)} - \hat{b}_{(t-1,i)} \right| \right) \right]. \tag{5}$$

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 previous 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 $2 * x_{(t,i)} - 1 - c$, where $x_{(t,i)}$ is the price relative of the ith asset for the tth 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, i.e., one invests the money among a set of assets according to the initial portfolio $\mathbf{b}_1$, and holds the portfolio without any change during the entire trading period. The *BAH* strategy with an uniform portfolio, i.e., $\mathbf{b}_1 = (\frac{1}{m}, \ldots, \frac{1}{m})$, 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*) [Cover and Gluss 1986], 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 $\mathbf{b}$ for CRP, the total wealth achieved by CRP at the end of the Tth trading day is, $\mathbf{S}_T = \mathbf{S}_0 \prod_{t=1}^{T} (\mathbf{b} \cdot \mathbf{x}_t)$. A special case of CRP is to uniformly redistribute the total wealth to all investments, i.e., $\mathbf{b} = (\frac{1}{m}, \ldots, \frac{1}{m})$, 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 $\mathbf{S}_T^\star = \max_{\mathbf{b} \in \triangle_m} \mathbf{S}_T$. 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, the follow-the-leader strategies aim to minimize the *regret* between the strategy A and the BCRP strategy at the horizon n: $Regret_n(A) = \sum_{t=1}^{n} \log(\mathbf{b}^* \cdot \mathbf{x}_t) - \sum_{t=1}^{n} \log(\mathbf{b}_t \cdot \mathbf{x}_t)$. Example techniques in this category include Cover's Universal Portfolios [Cover 1991], the Exponential Gradient strategy [Helmbold et al. 1998], and the 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 run time complexity is $O(T^m)$, 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 an time-efficient implementation of Cover's UP based on non-uniform 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 *Exponential Gradient* (EG) strategy [Helmbold et al. 1998] for 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 $O(Tm)$ running time. While its regret is not as tight as Cover's UP, however, its linear time complexity substantially surpasses the latter.

Recently convex optimization has been applied to resolve the PS problem [Agarwal et al. 2006]. 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 variation 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 $O(Tm^3)$. Following this work, Hazan and Seshadhri [2009] very recently proposed a new adaptive-regret approach, which is essentially also an ONS based strategy though they provide more descent theoretical results.

3.3   Similarity-driven Strategies

The similarity-driven learning to trade strategies usually optimize the trading strategy by mining potential similarity 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* ($\mathbf{B}^K$) learning strategies for PS based on nonparametric prediction techniques [Györfi and Schäfer 2003]. In their approach, the algorithm first identifies a list of similar historical price relative sequences whose Euclidean distances with the recent market windows are smaller than a threshold; it then 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 $\mathbf{B}^K$ strategy, mainly to improve the computational efficiency. Following the same framework as the $\mathbf{B}^K$ strategy, the *Nonparametric Nearest Neighbor learning* ($\mathbf{B}^{NN}$) strategy proposed by Györfi et al. [2008] aims to search for the ℓ nearest neighbors in the historical price relative sequences rather than searching price relatives within a specified Euclidean ball. This method has been empirically shown to be a rather robust trading strategy for PS.

3.4   Time-Series Prediction based Strategies

In finance engineering, there are a number of well-studied time series prediction models [Tsay 2005]. These models may be adapted to the portfolio selection tasks, although they were not proposed to optimize the portfolio selection problem. In general, there are mainly two categories of models for time series prediction in finance, i.e., linear and nonlinear models.

For linear models, *autoregressive moving average* (ARMA) [Box et al. 1994] is one of the most important 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, for nonlinear models, there are also some well-studied 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.
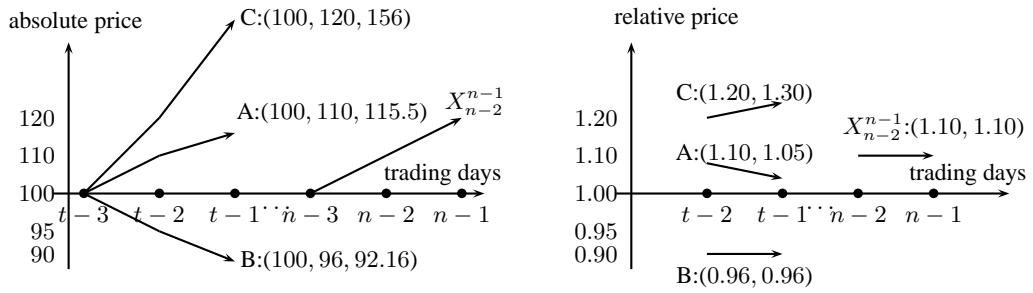
Fig. 1: A motivating example to illustrate the limitation of the Euclidean measure. The left diagram represents the absolute price movements of market windows A, B, C, and X_{n-2}^{n-1} in the consecutive three days (here the first price is only for the calculation of the price relatives). The starting prices of all of them are set to 100. The numbers in the parenthesis 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 parenthesis of A, B, C, and X_{n-2}^{n-1} are their price relative vectors.

4. CORN: CORRELATION-DRIVEN NONPARAMETRIC LEARNING STRATEGY

In this section, we present a new *learning to trade* strategy termed **COR**relation-driven **N**onparametric 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 pattern/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; Györfi et al. 2008] attempt to discover the 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; Györfi et al. 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 Fig. 1. Assume a market consists of only one asset, and the window size is fixed to 2. Let the latest market window for the nth trading day $\mathbf{X}_{n-2}^{n-1} = (1.10, 1.10)$ 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 Fig. 1. In Fig. 1, 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, by the Euclidean based approach, i.e., $\|\mathbf{X}_{t-w}^{t-1} - \mathbf{X}_{n-w}^{n-1}\| \leq 0.2$ indicates that market windows A and B are detected as most similar to the latest market window $\mathbf{X}_{n-2}^{n-1}$, 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 from irrelevant or even harmful market windows (such as market window B) and the ignorance 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 the powerful nonparametric learning techniques to effectively optimize the portfolio. 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* [Rodgers and Nicewander 1988], 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 the 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 define a *correlation-similar* set $\mathcal{C}_t(w, \rho)$ that contains the historical price relatives whose previous market windows are statistically correlated to the latest market window. In particular, the *correlation-similar* set $\mathcal{C}_t(w, \rho)$ is formally defined as follows:

$$\mathcal{C}_t(w, \rho) = \left\{ w < i < t - 1 \middle| \frac{\mathrm{cov}(\mathbf{X}_{i-w}^{i-1}, \mathbf{X}_{t-w}^{t-1})}{\mathrm{std}(\mathbf{X}_{i-w}^{i-1})\mathrm{std}(\mathbf{X}_{t-w}^{t-1})} \geq \rho \right\}, \tag{6}$$

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

The correlation coefficient distinguished the proposed CORN algorithm from the previous nonparametric learning series in the following aspects.

First, in this paper, all the methods equivalently use the price relatives, i.e., the changes of the absolute prices. As shown in the motivating example, the drawback of existing Euclidean based methods comes from that Euclidean only considers the strength of the difference of the two price relatives, where no direction information is considered. To overcome this drawback, we propose correlation coefficient to measure the difference of the two price relatives. With such important direction information, we can better identify the similar price relatives, which contribute to the excellent performance of the proposed

strategy. Some readers may argue if we can use Euclidean to measure the direction information directly, for example, the slope of the centralized points (detail can be seen in Section 4.4). However, by 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 series, all of which ignore the whole market information.

Third, the correlation coefficient not only concerns about the degree of linear dependence or similarity, but also cares about the directions of the vectors indicated by the signs. Although $-\rho$ and ρ ($\rho > 0$) intuitively correspond to equivalent strength of linear dependence or similarity, they are in the opposite directions, i.e., either one is up-trend and the other is down-trend with respect to the target market windows. We choose ρ as the threshold, as in the stock market we are interested in the market windows with the similar appearances in terms of both strength and directions. The direction information also distinguishes the proposed CORN algorithm from the previous nonparametric learning strategies with Euclidean measure to locate the similar appearance, which ignore the direction information.

4.3 Algorithm

Next we present the proposed **COR**relation-driven **N**onparametric 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, ρ), i.e., $\{\mathcal{E}(w, \rho) : w \geq 1, -1 \leq \rho \leq 1\}$. The expert $\mathcal{E}(w, \rho)$ is identified by its window size w and correlation coefficient threshold ρ. 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 $\mathcal{E}(w, \rho)$ as $\mathcal{E}(w, \rho) = \mathbf{b}(w, \rho)$.

For each expert $\mathcal{E}(w, \rho)$, after calculating the *correlation-similar* set $\mathcal{C}_t(w, \rho)$ at the beginning of tth trading day, we propose to learn the optimal portfolio by maximizing the total wealth over the sequence of price relatives by following the similar idea of the BCRP strategy [Cover and Gluss 1986], i.e.,

$$\mathbf{b}_t(w, \rho) = \arg\max_{\mathbf{b} \in \triangle_m} \prod_{i \in \mathcal{C}_t(w, \rho)} (\mathbf{b} \cdot \mathbf{x}_i), \tag{7}$$

where $\triangle_m$ represents a simplex with m components. It is possible that $\mathcal{C}_t(w, \rho)$ is empty

for a large ρ value, for which we will simply adopt a uniform portfolio $(\frac{1}{m}, \ldots, \frac{1}{m})$. The general procedure for each expert is summarized in Algorithm 1 shown in Fig. 2. Moreover, the *correlation-similar* set usually consists of a large number of correlated price relatives. Thus, if some price relative (whose correlation is within the threshold) has occurred frequently in the history, it will also appear multiple times in the *correlation-similar* set. In other words, Eq. (7) has somewhat considered the occurrence/confidence of the correlated price relatives, which would avoid simply taking an extreme case in the history.

Algorithm 1 The CORN Expert Learning Procedure ($\mathcal{E}(w, \rho)$)

Input:

      t: Index of current trading day, $\mathbf{X}_1^{t-1}$: Historical market windows, w: Window size for the expert, ρ: Correlation coefficient threshold

Output:

      $\mathbf{b}_t$: Expert's portfolio for the tth trading day

Procedure

  1: Initialize: $\mathcal{C}_t(w, \rho) = \emptyset$
  2: **if** $t \leq w + 1$ **then**
  3:      return $\mathbf{b}_t = (\frac{1}{m}, \ldots, \frac{1}{m})$
  4: **end if**
  5: **for** $i = w + 1$ **to** $t - 1$ **do**
  6:      **if** corrcoef$(\mathbf{X}_{i-w}^{i-1}, \mathbf{X}_{t-w}^{t-1}) \geq \rho$ **then**
  7:        $\mathcal{C}_t(w, \rho) = \mathcal{C}_t(w, \rho) \cup \{i\}$
  8:      **end if**
  9: **end for**
10: **if** $\mathcal{C}_t(w, \rho) == \emptyset$ **then**
11:      return $\mathbf{b}_t = (\frac{1}{m}, \ldots, \frac{1}{m})$
12: **else**
13:      Search for the optimal portfolio:
         $\mathbf{b}_t = \arg \max_{\mathbf{b} \in \triangle_m} \prod_{i \in \mathcal{C}_t(w, \rho)} (\mathbf{b} \cdot \mathbf{x}_i)$
14: **end if**

Fig. 2: The proposed CORN Expert Learning Procedure.

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 tth trading day can be calculated as follows:

$$\mathbf{b}_t = \frac{\sum_{w,\rho} q(w, \rho) s_{t-1}(w, \rho) \mathbf{b}_t(w, \rho)}{\sum_{w,\rho} q(w, \rho) s_{t-1}(w, \rho)}, \tag{8}$$

where $\mathbf{b}_t(w, \rho)$ represents the portfolio output by each individual expert $\mathcal{E}(w, \rho)$ 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 $\mathbf{b}_t$ by the above equation, we will output it as the desired portfolio for the tth 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 follows:

$$\mathbf{S}_t = \mathbf{S}_{t-1} \times (\mathbf{b}_t \cdot \mathbf{x}_t), \tag{9}$$

where $\mathbf{S}_{t-1}$ represents the total wealth achieved till the $(t-1)^{th}$ trading day and initial capital $\mathbf{S}_0 = 1$. For each expert, CORN updates its performance $s_t(w, \rho)$ after t trading periods, which can be calculated as follows:

$$s_t(w, \rho) = s_{t-1}(w, \rho) \times (\mathbf{b}_t(w, \rho) \cdot \mathbf{x}_t), \tag{10}$$

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

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 $q(w, \rho)$, i.e.,

$$\mathbf{S}_T = \sum_{w, \rho} q(w, \rho) s_T(w, \rho). \tag{11}$$

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.

In terms of different expert combinations, we present two CORN variants, i.e., 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, i.e., $q(w, \rho) = \frac{1}{W}$, 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 ρ. The details of the CORN-U algorithm are shown in Fig. 3.

The above uniform combination algorithm may include some poor experts, leading to the degradation of the overall performance. To overcome the 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, i.e., the strategy assigns the set of top K best experts a uniform distribution $q(w, \rho) = \frac{1}{K}$, while the weights assigned on the 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 has a different ρ value. In our empirical study, the ρ value of the ith expert is set to $\frac{i-1}{P}$. The CORN-K algorithm is presented in Algorithm 3 as shown in Fig. 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 2-dimensional space. Fig. 5 shows an intuitive example that corresponds the example used in Section 4.1 from a geometrical view. In the figure, the origin point (μ_i, μ_t) denotes the mean point of price relative vectors, $\mathbf{X}_{t-w}^{t-1}$ denotes the market window of current t^{th} trading day, $\mathbf{X}_{i-w}^{i-1}$ denotes the market window of the ith trading day on the historical price

---

Algorithm 2 The proposed CORN Uniform Combination Algorithm (CORN-U)

Input:

$\mathbf{X}_1^T = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$: Historical market windows, W: Maximum window size for experts, ρ: Correlation coefficient threshold

Output:

$(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_T)$: Portfolio strategy

Procedure

1: Initialize $\mathbf{S}_0$ and W experts: $\mathbf{S}_0 = 1$, $q(w, \rho) = \frac{1}{W}$
2: **for** $t = 1$ **to** T **do**
3:     **for** $w = 1$ **to** W **do**
4:         CORN Expert Learning (Algorithm 1) to find the portfolio: $\mathbf{b}_t(w, \rho) = \mathcal{E}(w, \rho)$
5:     **end for**
6:     Combine the experts' portfolios:

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

7:     Update the total wealth: $\mathbf{S}_t = \mathbf{S}_{t-1} \times (\mathbf{b}_t \cdot \mathbf{x}_t)$
8:     Update the experts: $s_t(w, \rho) = s_{t-1}(w, \rho) \times (\mathbf{b}_t(w, \rho) \cdot \mathbf{x}_t)$
9: **end for**

Fig. 3: The proposed CORN Uniform combination Algorithm (CORN-U).

relative sequence, and point A, B and C represent another three market windows on the historical price relative sequence.

From a geometrical view of point, we know that the correlation coefficient between two market windows $\mathbf{X}_{t-w}^{t-1}$ and $\mathbf{X}_{i-w}^{i-1}$ is equivalent to the *cosine* of angle θ between these two vectors [Rodgers and Nicewander 1988], i.e., $\cos \theta = \text{corrcoef}(\mathbf{X}_{i-w}^{i-1}, \mathbf{X}_{t-w}^{t-1})$. Thus, given a correlation coefficient threshold ρ, the approach of searching for market windows satisfying $\text{corrcoef}(\mathbf{X}_{i-w}^{i-1}, \mathbf{X}_{t-w}^{t-1}) \geq \rho$ is equivalent to finding market windows $\mathbf{X}_{i-w}^{i-1}$ with $|\theta| \leq \arccos \rho$. When ρ is simply fixed to 0, it reduces to looking for market window vectors $\mathbf{X}_{i-w}^{i-1}$ which have angle $|\theta| \leq 90°$ with respect to $\mathbf{X}_{t-w}^{t-1}$. In another words, the CORN strategy locates all market windows $\mathbf{X}$ that satisfy $\mathbf{a}^\top \mathbf{X} \geq 0$ or intuitively those points on the righthand side of line $\mathbf{a}^\top \mathbf{X} = 0$, where $\mathbf{a}$ is a unit vector that is perpendicular to the vector from (μ_i, μ_t) to $\mathbf{X}_{t-w}^{t-1}$.

On the other hand, the nonparametric learning strategy $\mathbf{B}^K$ aims to locate market windows $\mathbf{X}_{i-w}^{i-1}$ within a Euclidean ball centered at $\mathbf{X}_{i-w}^{i-1}$ with radius $r_{k,l}$, i.e., $\|\mathbf{X}_{i-w}^{i-1} - \mathbf{X}_{t-w}^{t-1}\| \leq r_{k,l}$. 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, which however is a harmful window as the moving trend is completely different from the latest market window $\mathbf{X}_{t-w}^{t-1}$. Moreover, it may also exclude some informative and beneficial market windows. For example, point C:$(1.20, 1.30)$ that is excluded by the Euclidean approach is considered as an important market window that is highly positive correlated with $\mathbf{X}_{t-w}^{t-1}$. Second, the Euclidean based approach clearly does not consider the market information, which is represented by point (μ_i, μ_t) in the figure.

Algorithm 3 The proposed CORN TOP-K combination Algorithm (CORN-K)

Input:

$\mathbf{X}_1^T = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$: 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:

$(\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_T)$: the output portfolio strategy

Procedure

1: Initialize $\mathbf{S}_0$ and $W \times P$ experts: $\mathbf{S}_0 = 1, \mathbb{P} = \left\{ 0, \frac{1}{P}, \ldots, \frac{P-1}{P} \right\}, q(w, \rho) = \frac{1}{W \times P}$

2: **for** $t = 1$ **to** T **do**

3:     **for** $w = 1$ **to** W **do**

4:         **for** $\rho \in \mathbb{P}$ **do**

5:             CORN Expert Learning (Algorithm 1) to find the portfolio:
            $\mathbf{b}_t(w, \rho) = \mathcal{E}(w, \rho)$

6:         **end for**

7:     **end for**

8:     Combine the TOP-K experts' portfolios:

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

9:     Update the total wealth: $\mathbf{S}_t = \mathbf{S}_{t-1} \times (\mathbf{b}_t \cdot \mathbf{x}_t)$

10:     Update the experts: $s_t(w, \rho) = s_{t-1}(w, \rho) \times (\mathbf{b}_t(w, \rho) \cdot \mathbf{x}_t)$

11:     TOP-K and expert weight updates:
    Select top K experts $\{\mathcal{E}(w, \rho)\}$ w.r.t. $s_{t-1}(w, \rho)$
    Set weights for the top K experts: $q(w, \rho) = \frac{1}{K}$
    Set weights for other experts: $q(w, \rho) = 0$

12: **end for**

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

Note that the above analysis could be easily extended to multidimensional vectors in general scenarios, i.e., $w \times m$ dimensions where w is the window size, and m is the number of stocks. The above 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 ρ and the window size w. Below we analyze how they affect the algorithms.

As shown in the motivating example in Section 4.1, the correlation coefficient threshold ρ is critical to the *correlation-similar* set. If ρ is negative, the *correlation similar* set would contain may negatively related price relative vectors or irrelevant price relative vectors. On the other hand, if ρ 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 the 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
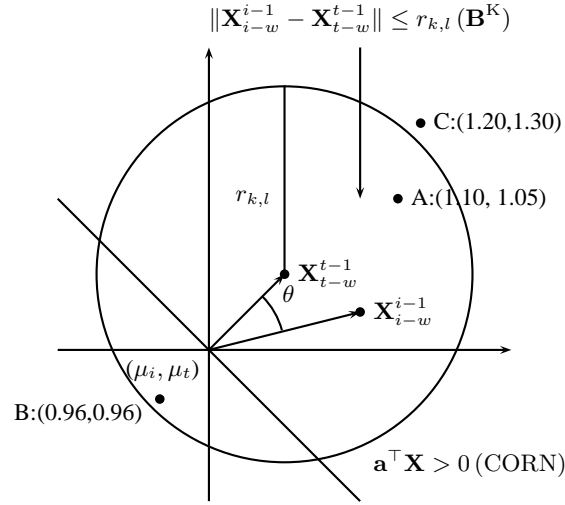
Fig. 5: Geometrical interpretation of the proposed CORN strategy in comparison to the Nonparametric Kernel-based Moving Window ($\mathbf{B}^{\mathrm{K}}$) learning strategy.

ρ value is dataset dependent, but often close to 0. Numerical verification will be provided in Section 5.9.

Moreover, we note that CORN would reduce to some special case when setting $\rho \to 1$. In particular, when the correlation coefficient threshold $\rho \to 1$, the CORN algorithm reduces to the Uniform CRP (UCRP) strategy. It is straightforward to verify this by noting that when $\rho \to 1$, fewer market windows are highly positive correlated to the latest window vector; in the extreme case of $\rho = 1$, $\mathcal{C}_t(w, \rho)$ becomes almost empty, which thus reduces to the Uniform CRP strategy. Numerical verification will be shown 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 to 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, i.e., 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 maximum window size.

Remarks. Although the proposed algorithms for portfolio selection are simple and effective, readers may still want to figure out whether it is reasonable to make portfolio using only the market price information, and what are the basic assumptions and reasons for this method to achieve excellent performance based on the historical prices. Below gives some justifications to answer the question. First of all, it is often a debate whether it is reasonable to make a portfolio selection decision based on only the historical market price information as some may believe that the market price is only a sign of economic behaviors. In fact, this is related to the long-standing combat between fundamental analysis and technical analysis in finance. Our goal is not to completely resolve such a great challenging debate, but instead we provide some empirical evidences to endorse the effectiveness of technical analysis methods. Second, as a method belonging to the category of technical analysis, the success of our method thus depends on three basic assumptions that are common for most technical analysis methods, including (1) market action discounts everything, i.e., techni-

cal analysis assumes stock price at any given time reflects everything that has/could affect the company including fundamental factors; (2) price move in trends, and (3) history tends to repeat itself. With these assumptions, it is not difficult to understand the principles of the proposed CORN strategy, which considers only the market prices, adopts the correlation coefficient to find trend information in the historical information, and attempts to locate the repeated patterns from the correlation similar set.

5. EXPERIMENTS

5.1 Experimental Testbed on Real Data

In our experiments, we perform numerical evaluations on four real datasets[1] 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; Györfi et al. 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 Dec 31st 1984. In our experiments, we refer it to as "NYSE (O)". For consistency, we also collected another latest dataset in the New York Stock Exchange (NYSE)[2] market from Jan 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 of the previous 36 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 that was used in 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 Jan 2nd 1998 to Jan 31st 2003. The fourth dataset is a collection of global equity indices collected from MSCI[3]. It contains three indices which represent the equity markets of Pacific, North America, and Europe, ranging from Sept 9th 2005 to Sept 7th 2009 with a total of 1042 trading days.

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

Table I: Summary of four real datasets.

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 markets.

---

[1]All datasets can be downloaded from http://www.cais.ntu.edu.sg/~libin/portfolios.
[2]We collected the data from Yahoo finance. http://finance.yahoo.com.
[3]We collected the data from MSCI Barra. http://www.mscibarra.com.

5.2   Experimental Setup and Metrics

One salient merit of the proposed CORN algorithm is its nonparametric nature, i.e., it is almost parameter-free. However, in reality, there are two possible parameters that affect the performance, i.e., the correlation coefficient threshold ρ 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 examine the influence of these parameters in Section 5.9.

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 trading algorithm. In addition, we also adopt the *Maximum drawdown* (MDD) for the drawdown analysis of the learning to trading 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 below[4]:

(1) Market: the Market strategy (the uniform BAH approach);

(2) Best-Stock: the best stock in the market that is a hindsight strategy;

(3) BCRP: the 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(η)) algorithm with the best parameter η fixed to 0.05 as suggested by the authors [Helmbold et al. 1998];

(6) ONS: Online Newton Step (ONS(η, β, γ)) with the best parameters set as the same suggested by the authors in Agarwal et al. [2006], i.e., $\eta = 0, \beta = 1, \gamma = 1/8$;

(7) Anticor: BAH_{30}(Anticor) as a variant of Anticor to smooth the volatility, which is a better solution proposed by the authors [Borodin et al. 2004];

(8) $\mathbf{B}^K$: Nonparametric kernel-based moving window ($\mathbf{B}^K(c)$) 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) $\mathbf{B}^{NN}$: Nonparametric nearest neighbor based strategy with parameter $W = 5, L = 10, p_\ell = 0.02 + 0.5\frac{\ell - 1}{L - 1}$ as the authors suggested [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 $\mathbf{S}_0 = \$1$ 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, i.e., the uniform BAH strategy, on all the

---

[4]We can adjust the parameters of comparators for better performance, but that is beyond the scope of this paper.

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
$\mathbf{B}^K$	1.08E+09	9.5E+02	2.26	1.27
$\mathbf{B}^{NN}$	3.35E+11	5.59E+04	3.09	**37.43**
CORN-U	**1.48E + 13**	**3.32E + 05**	**6.35**	31.54
CORN-K	**6.29E + 13**	**4.38E + 05**	**8.56**	**48.72**

Table II: Total wealth achieved by various strategies on four real datasets. The numbers in boldface represent the top two achievements on each dataset.

datasets. This shows that it is promising to investigate learning to trade algorithms for portfolio selection. Second, except Anticor, most existing trading algorithms do not always outperform the best stock in the market on the four datasets, except Anticor. Third, we observe that the regular follow-the-leader approaches (UP, EG, ONS) often perform substantially worse than the other state-of-the-art approaches. Finally, among all compared algorithms, the proposed CORN-U and CORN-K algorithms always achieved 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 $\mathbf{B}^{NN}$ algorithm that achieves \$335-billion and the $\mathbf{B}^K$ algorithm that achieves \$1.16-billion. On the NYSE (N) datasets that consists of up-to-date data, both the CORN-U and CORN-K strategies also achieved over \$332-thousand and \$438-thousand, respectively, while the existing state-of-the-art strategy achieved about \$73.7-thousand by the Anticor strategy and \$55.9-thousand by the $\mathbf{B}^{NN}$ strategy. 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 found 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 achieved considerably better results than the market index, the best stock in the market, as well as all the state-of-the-art strategies. 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.

Besides the above results, we are also interested in examining how the total wealth achieved by various strategies change over different trading periods. Fig. 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 the 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 the NYSE (O) and NYSE (N) datasets where the growth rate after 2500 trading days is much higher than the previous trading periods. Such phenomenon establishes that when more historical data are available for the learning to trade task, the
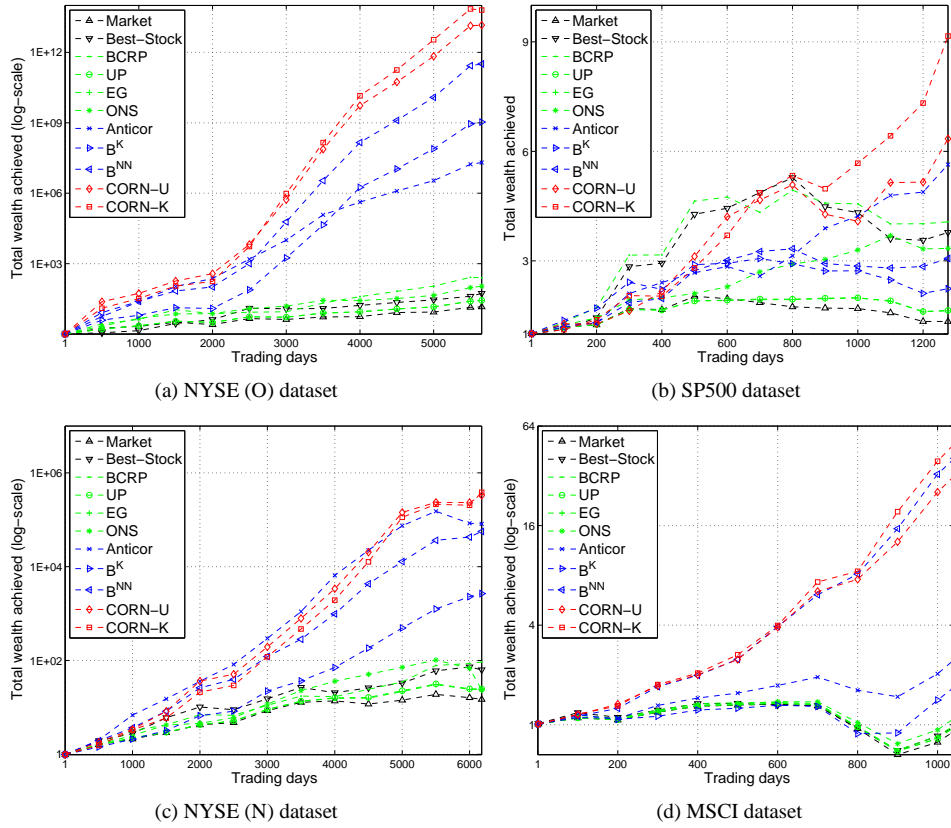
Fig. 6: Total wealth achieved by various trading strategies over different trading days.

CORN algorithms are able to perform more effective trading by exploiting statistical correlation with the powerful nonparametric learning approach. 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. Table III summarizes the results of APYs, the Risks and annualized Sharpe Ratios for all the strategies. 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 Shape Ratio. For example, for the Market strategy on NYSE (O) dataset, the APY of the Market strategy is 13%, its Risk or annualized standard deviation of daily return is 15%, and its annualized Sharpe Ratio is 60%. From Table III, we observe that on the NYSE (O), NYSE (N) and SP500 datasets, both the CORN algorithms achieved the highest APY and annualized Sharpe Ratio 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 excellent as the other state-of-the-art strategies. Similar to the common fact of no pain no gain in financial markets, i.e., a higher return is often associated with a higher risk, the risk of

Strategies	NYSE (O)	NYSE (N)	SP500	MSCI
Market	$13\% \pm 15\%$	$12\% \pm 18\%$	$6\% \pm 24\%$	$-2\% \pm 20\%$
	60%	43%	8%	-31%
Best-stock	$20\% \pm 24\%$	$18\% \pm 29\%$	$30\% \pm 51\%$	$-1\% \pm 25\%$
	65%	50%	52%	-19%
BCRP	$29\% \pm 31\%$	$20\% \pm 24\%$	$32\% \pm 42\%$	$0\% \pm 22\%$
	80%	69%	67%	-20%
UP	$16\% \pm 14\%$	$13\% \pm 19\%$	$10\% \pm 22\%$	$0\% \pm 20\%$
	91%	51%	28%	-24%
EG	$16\% \pm 13\%$	$14\% \pm 19\%$	$11\% \pm 22\%$	$-1\% \pm 20\%$
	91%	52%	30%	-24%
ONS	$24\% \pm 18\%$	$14\% \pm 34\%$	$27\% \pm 24\%$	$2\% \pm 20\%$
	110%	29%	98%	-8%
Anticor	$113\% \pm 29\%$	$58\% \pm 33\%$	$41\% \pm 38\%$	$25\% \pm 21\%$
	378%	163%	97%	100%
$\mathbf{B}^{K}$	$158\% \pm 36\%$	$32\% \pm 25\%$	$17\% \pm 33\%$	$6\% \pm 20\%$
	422%	113%	40%	10%
$\mathbf{B}^{NN}$	$234\% \pm 40\%$	$56\% \pm 27\%$	$25\% \pm 39\%$	$\mathbf{147}\% \pm 25\%$
	571%	189%	55%	**580**%
CORN-U	$\mathbf{297}\% \pm 49\%$	$\mathbf{68}\% \pm 33\%$	$\mathbf{45}\% \pm 41\%$	$137\% \pm 25\%$
	600%	**194**%	**97**%	533%
CORN-K	$\mathbf{324}\% \pm 52\%$	$\mathbf{70}\% \pm 32\%$	$\mathbf{54}\% \pm 40\%$	$\mathbf{164}\% \pm 27\%$
	619%	**204**%	**123**%	**602**%

Table III: APYs, Risks and Sharpe Ratios for various strategies on the four datasets. The upper row of each cell shows APY$\pm$ Risk and the second row shows Sharpe Ratio. The top two ratios on each dataset are highlighted.

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, Fig. 7 shows the quarterly return distribution of the CORN-U strategy on NYSE (O) and NYSE (N) datasets and monthly return distribution of the CORN-U strategy on SP500 and MSCI datasets. For comparison, the corresponding market return distribution is shown in the figure as the benchmark. In the Fig. 7a and Fig. 7b, most of the quarterly returns with the CORN-U strategy on NYSE (O) and NYSE (N) dataset are higher than the quarterly return of market index. More specifically, during the 72 quarters out of total 86 quarters (85%) for NYSE (O) dataset, 76 out of 94 quarters (81%)for NYSE (N) dataset, the returns acquired with the CORN-U strategy outperform the market returns. In the third Fig. 7c, during 37 months out of 58 months (64%), the returns accumulated with the CORN-U strategy outperform the market returns. In the last Fig. 7d, 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 method is stable and robust.

5.7 Statistical Evaluation of Performance

Besides the above results, we also interested in evaluating the CORN strategy statistically [Katz and McCormick 2000]. Since our datasets are just samples for the entirely stock markets population, we try to validate the strategy for future. We conduct a *stu-*

(a) NYSE (O) dataset

(b) NYSE (N) dataset
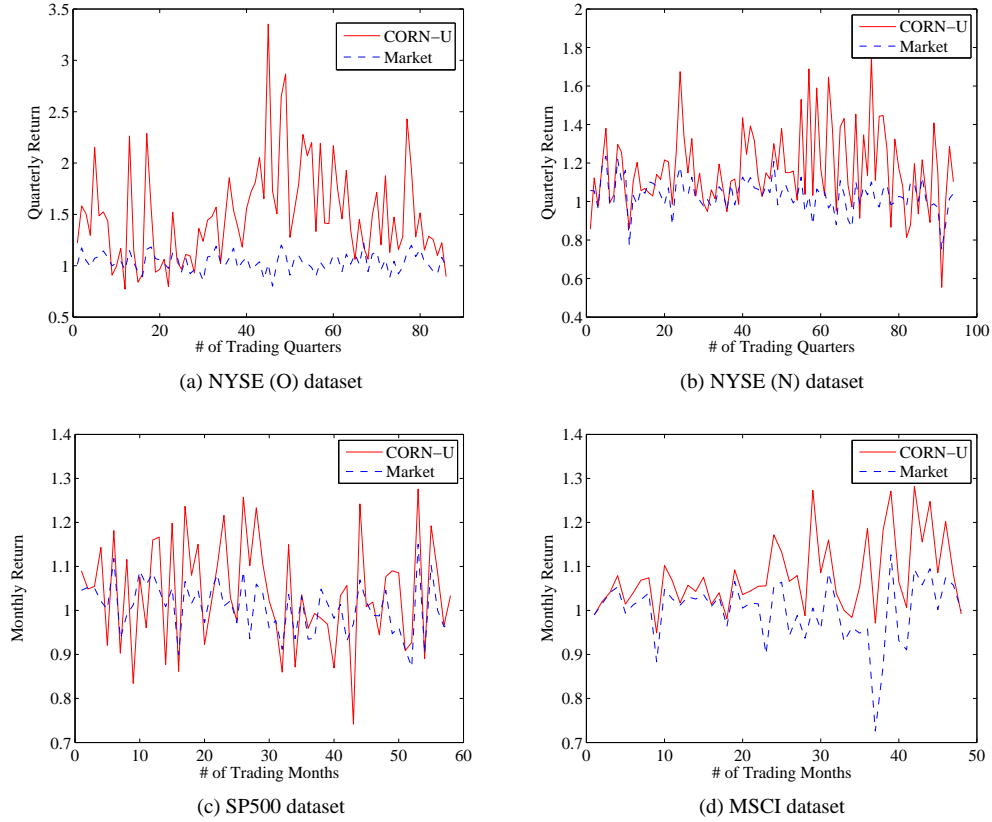
(c) SP500 dataset

(d) MSCI dataset

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.

dent 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, zero 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. The equation to calculate the t-statistic is $t\text{-statistic} = \frac{\text{Sample Profit Mean} - 0}{\text{Sample Standard Error}}$. 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 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. Concerning that each of our dataset contains such a large number of trading samples, we could regard the distribution of the profit/loss as normal, such that the statistical analysis regarding the mean is meaningful.

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 the daily basis. And the profit/loss is in the sense of absolute value, i.e., the return for the trading day is compared with 1.00. From the table, we can see that the t-statistics for the four datasets are so large that the significance for the four datasets approach zero. The results fantastically show that it is almost impossible to contribute the success of the CORN-U strategy to chance along.

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

Table IV: Statistical analysis of mean profit/loss for CORN-U strategy. The statistical analysis is on daily basis to test whether the success of CORN is due to chance.

5.8 Experiment 4: Evaluation of Drawdown

In finance, drawdown analysis is to measure of the decline from a historical peak in the total wealth achieved. The background knowledge is described in Section 2.2. This section is to show that the drawdown for the proposed CORN strategy is acceptable.

Fig. 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 $\mathbf{B}^{NN}$ 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 almost achieves the lower maximum drawdowns on the four datasets. It is worth noting that even with the financial crisis from 2007 to 2009, there is a huge drawdown on the MSCI dataset, i.e., the MDD for market strategy is 59.17%. However, the CORN strategies still perform much better than the market, the MDDs for the CORN strategies are 18.77% and 14.91%, respectively. Since drawdown is an important measure for the downside risk, this drawdown analysis strongly demonstrates that the risk on the proposed CORN strategies is acceptable even we design the strategy with the utility function of total wealth.

5.9 Experiment 5: Evaluation of Parameters

Following the intuitive analysis in Section 4.5, in this section, we experimentally evaluated the effects of the two parameters, correlation coefficient threshold ρ and maximum window size W.

To evaluate the effect of correlation coefficient threshold ρ, we analyze the performance of CORN algorithm by varying parameter ρ from -1.0 to $+1.0$ with fixed $W = 5$. Fig. 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 ρ approaches to 1. Further, the empirically optimal value of ρ is 0.3 for datasets NYSE (N), while results on other datasets show the optimal values of ρ is between 0 to 0.5. Finally, we found that CORN performs considerably poor when ρ is too large (excluding many informative cases) or too small (including
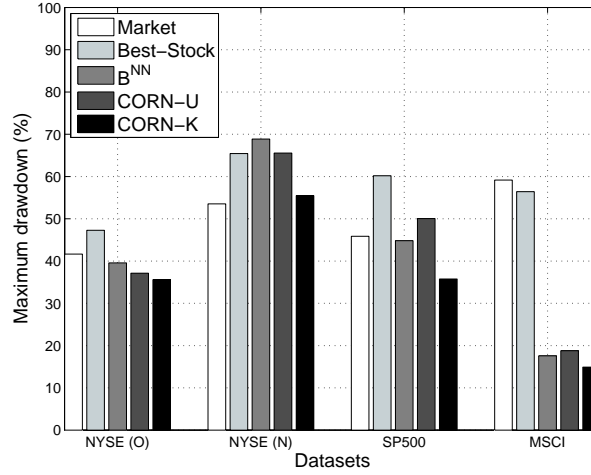
Fig. 8: Drawdown analysis of varied strategies on the four datasets. For comparison, the corresponding ratios for Market, best-stock and $\mathbf{B}^{NN}$ strategies are also provided. CORN-U and CORN-K are plotted on the rightmost of bar figure.

too much irrelevant cases), which is consistent with the motivation of the proposed CORN strategy.

Another important parameter is the maximum window size, W. We notice that different Ws may affect the performance of the proposed CORN algorithm. This experiment aims to examine the effect of varied Ws with fixed $\rho = 0.1$. Fig. 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 the window size does affect the performances of the proposed algorithm. Second, we however 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 Ws (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, their final performances are affected by their 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 performances of the experts, and show their corresponding proportions. Fig. 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, i.e., the majority of contribution ranges from 5% to 30%, which shows that the predictions of all these experts are rather robust. In reality, it is possible that some 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 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

Following the studies in Cover [1991] and Helmbold et al. [1998], we also tested our portfolio selection method on the cases where we are allowed to buy stocks on margin. We
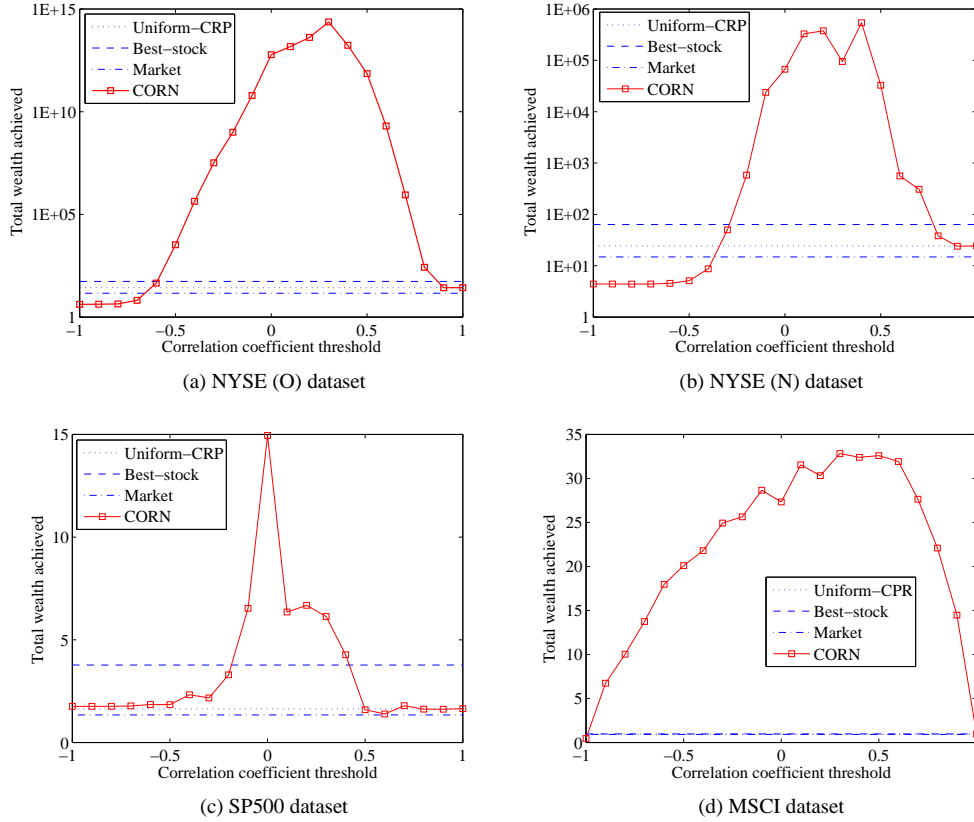
Fig. 9: Effect of correlation coefficient threshold on the total wealth achieved. With window size fixed to 5, the correlation coefficient threshold ranges from -1 to 1 with interval fixed to 0.1.

use the margin model described in Section 2.3. Table V shows the performances of the proposed CORN strategies without and with margin buying. For the sake of comparison, the performances of Market strategy, BCRP and state-of-the-art Anticor strategy and $\mathbf{B}^{NN}$ strategy are listed. It is shown that with the benefit of margin, almost all the strategies, especially the CORN strategies, gain a rapid profit growth on all the datasets. For example, on the NYSE (O) dataset, the total wealth of the CORN-U strategy increases from $1.48E+12$ to $1.05E+22$, which is $0.81E+10$ times the return of the CORN strategy without margin. On the recently NYSE(N) dataset, the total wealth of the CORN-U strategy increases from $3.32E+05$ to $9.39E+08$, which is about $2.82E+03$ times the return without margin. On the SP500 dataset, the total wealth achieved by the CORN strategies grow huge compared with other strategies, from 8.56 to 38.51, which is about 4.50 times the return without margin. For the MSCI dataset, the total wealth achieved by CORN-K strategy increases from 48.72 to 3585.82. The same as Experiment 5.4, the CORN strategies gain the highest total return on all the datasets with or without margin.

The experiment again indicates that the CORN strategies are effective and practical algorithms for the portfolio selection problem. It can take the advantages of margin and gain a explicit profit growth.
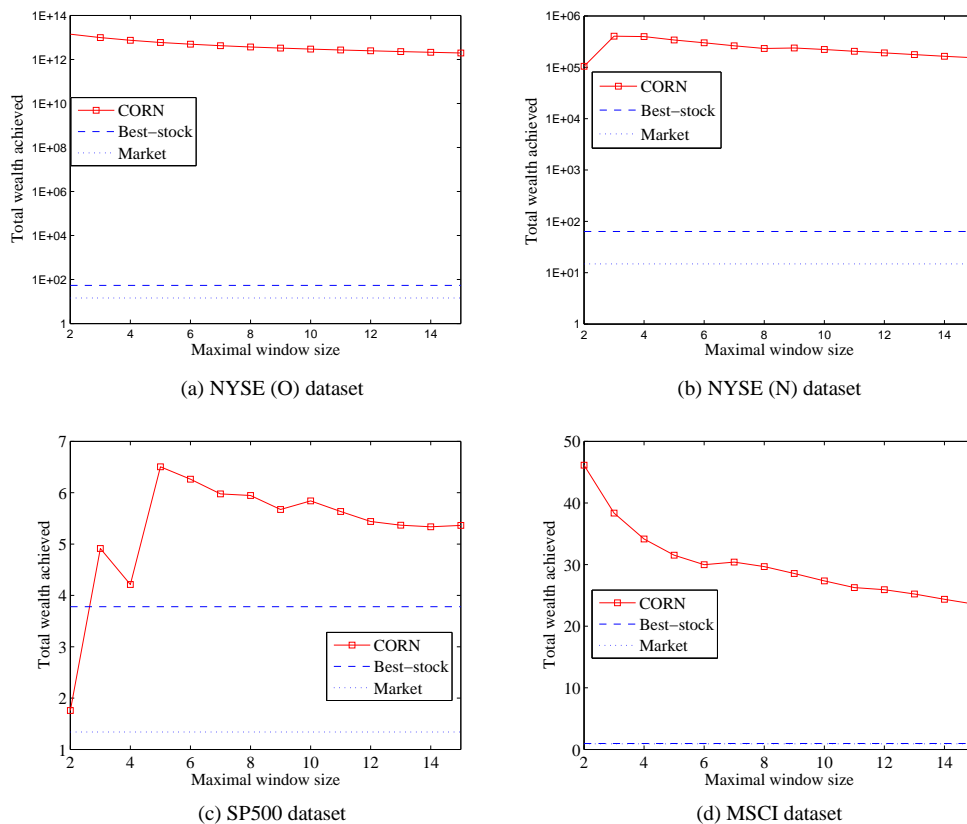
Fig. 10: Effect of window size on the total wealth achieved. With correlation coefficient threshold fixed to 0.1, the maximal window size ranges from 2 to 15 consecutively.
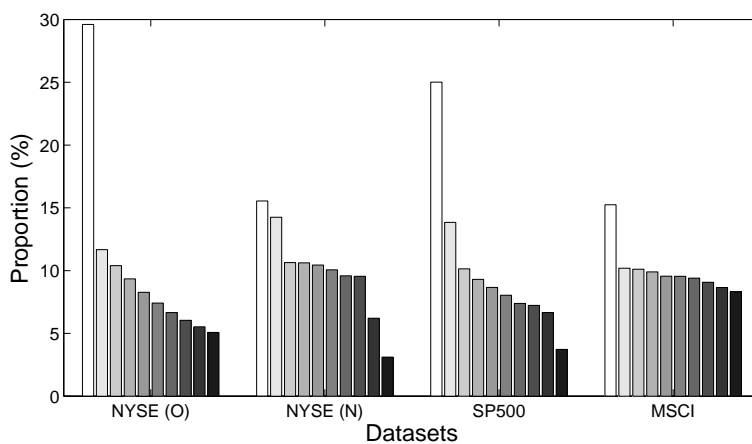

Fig. 11: Proportion of contribution to the final performance made by a set of 10 experts.

Dataset	Market	BCRP	Anticor	$\mathbf{B}^{NN}$	CORN-U	CORN-K
NYSE (O)	14.50	250.59	1.71E+07	3.35E+11	**1.48E + 13**	**6.29E + 13**
	15.7	3755.09	5.76E+12	3.17E+20	**1.05E + 22**	**1.83E + 25**
NYSE (N)	14.84	93.25	7.37E+04	5.59E+04	**3.32E + 05**	**4.38E + 05**
	14.10	662.53	1.05E+07	3.94E+07	**9.39E + 08**	**2.29E + 09**
SP500	1.34	4.06	5.55	3.09	**6.35**	**8.56**
	1.03	6.48	10.57	3.43	**14.59**	**38.51**
MSCI	0.92	0.99	2.45	**37.43**	31.54	**48.72**
	0.71	0.99	3.10	**1286.58**	1068.99	**3585.82**

Table V: Total wealth achieved by various strategies without and with margin. The number in the upper row shows the total wealth without margin, while the number in the lower row shows the total wealth with margin.

5.11 Experiment 7: Portfolio with Random Periods

To better show the robustness of the CORN strategy, and eliminate the impact from specific entry dates and time frames, i.e., to make the samples more representative, we randomly choose the entry dates and the running periods for the CORN-U strategy on the NYSE (O) dataset simply due to its relatively long trading periods. In our experiment, we randomly choose 100 samples from the NYSE (O) dataset. Among the randomly chosen trading periods, CORN-U outperforms the market index and the best stock with a probability of 94% and 76%, respectively. By observing the entry dates and sample lengths, we can find that the lengths for the losing cases are relatively short. The statistics for all the samples also verify our suspect. The sample length mean of the 100 random samples is 1464. For the losing cases, the sample length mean is relative short, 678, and the best stock losing sample length mean is 432. At the same time, the sample length mean for the winning cases is relatively long, 1513, and the best stock winning sample length mean is 1789. The results are consistent with the learning process of CORN strategies, i.e., 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 γ from 0% to 1.0% on the four datasets. It would be interesting if CORN can still outperform the two comparator, i.e., the market and the best stock, in the market when there is a nontrivial transaction cost. Fig. 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 quite convincing. The proposed CORN strategy is rather robust on the datasets except the SP500 dataset. The break-even commission rate of the total wealth achieved between the CORN and the two comparators for NYSE (O), NYSE (N) and MSCI datasets ranges from 0.2% to 0.6%, which is 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%. Event not as impressive as on the other three datasets, such break-even rate is still acceptable in practice. The reason for the results is because the best stock on SP500 is simply too strong such that it actually beats almost all the existing methods as shown in Table II.
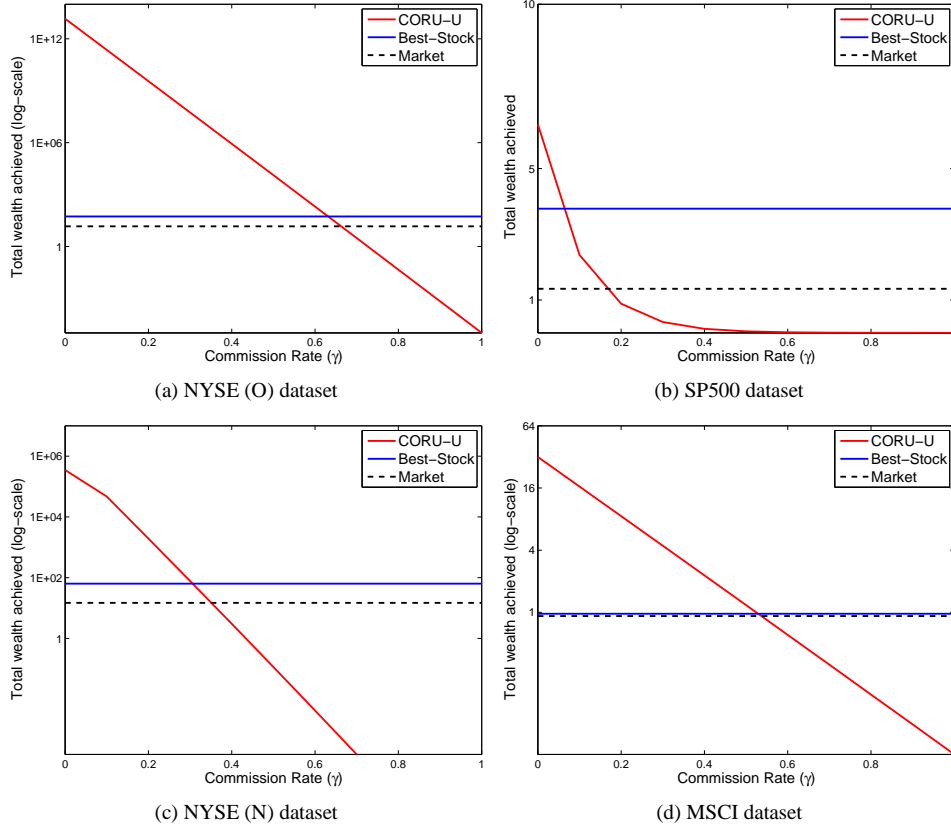
(a) NYSE (O) dataset

(b) SP500 dataset

(c) NYSE (N) dataset

(d) MSCI dataset

Fig. 12: Total wealth achieved by CORN-U on the four datasets with proportional commission rate γ varies from $0\%, 0.1\%, \ldots, 1.0\%$. Among the four dataset, the effect of transaction cost on NYSE (O), NYSE (N) and MSCI datasets are log-scaled for proper display. The break-even commission rate with the market index is about 0.6%, 0.3%, 0.2% and 0.5% for NYSE (O), NYSE (N), SP500 and MSCI datasets respectively.

5.13   Experiment 9: Evaluation of Computation Time

The following experiment is to evaluate the time efficiency of the proposed CORN strategy. In general, 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 2.33GHz at MATLAB. On the NYSE (O) dataset, CORN-U and CORN-K strategies took about 4 hours and 25 hour respectively for all the trading periods, which is better than the two other nonparametric learning algorithms $\mathbf{B}^K$ and $\mathbf{B}^{NN}$ that 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 $\mathbf{B}^K$ and $\mathbf{B}^{NN}$ 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.

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. As stated before, the proposed CORN strategy can be easily extended to make sequential time series prediction. In this section, we compare CORN against the well-known ARMA & GARCH methods for time series predictions on the stock datasets. Note that we do not develop the entire trading system based on these two methods, since they were not proposed for portfolio selection tasks, and there are various components for designing such an entire trading system. Thus, we sequentially make predictions for each stock for the next trading day and select the stock with the highest prediction, i.e., putting 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], i.e., setting (p, q)=(1, 1). Similarly, we set parameters for GARCH(p, q) to the default values, i.e., (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. In practice, 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 criterion 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 criteria denotes the *strength* of the profitability: *average ratio of the daily wealth gained by the strategy over that achieved by the market strategy*. In practice, the first criteria denotes the chance how likely the prediction based strategy is able to produce profit better than the market strategy, and the second criteria denotes the ratio between the profit produced by the prediction strategy over that of the market. The higher the values of these criterion, the better performance the algorithm achieves on the sequential time series prediction tasks.

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)	48.71%	49.44%	52.82%	53.52%
	0.9999	1.0000	1.0018	1.0019
SP500	47.65%	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

Table VI: Comparison of the proposed CORN strategy against two time series prediction methods (ARMA & GARCH). For each dataset, the first row denotes the accuracy, and the second row is the strength of profitability.

Table VI shows the results, which clearly indicate CORN significantly beats the well-known time series prediction algorithms, i.e., ARMA & GARCH. For the first criteria, the prediction accuracy of CORN significantly surpasses those of the ARMA & GARCH. In particular, all accuracies produced by ARMA are below 50%, and those produced by the GARCH strategy are fluctuating around 50%, while all accuracies obtained by CORN are over 50%. This clearly shows that CORN performs significantly better than the traditional widely used ARMA & GARCH time series prediction techniques. For the second criteria, 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 where the accuracies are below 50%. On the other hand, the results of CORN are always above 1, significantly

surpassing the ARMA & 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 above, CORN has been empirically proved as 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 explained that the market has some hidden information which has not yet been explored by market traders. While at the same time, our method using statistical correlation can exploit such hidden information, leading to the amazing performance. Although we cannot provide what are 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 fiance engineering. In particular, one useful fact is that our promising result provides strong evidence to show that the market is inefficient, which has been explained in Section 1. Another useful knowledge is that the price does often move in trends and the price relative patterns could reappear in practice. Such knowledge provides evidences 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 portfolio in real-world finance applications.

6. CONCLUSION

This paper proposed a novel **COR**relation-driven **N**onparametric learning (CORN) strategy for portfolio selection, which effectively exploits the statistical correlation information hidden in the 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 in the market, and also consistently surpasses a variety of state-of-the-art algorithms. Moreover, previous research and our research shows that the proposed method can be easily extended to solve sequential time series prediction problem.

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're currently developing such risk-limiting strategies for CORN. Moreover, we're also looking at exploiting transaction volume information, which could be a 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.

Acknowledgement

REFERENCES

AGARWAL, A., HAZAN, E., KALE, S., AND SCHAPIRE, R. E. 2006. Algorithms for portfolio management based on the newton method. In *ICML*. ACM, New York, NY, USA, 9–16.

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

BIAU, G., BLEAKLEY, K., GYÖRFI, L., AND OTTUCSÁK, G. 2010. Nonparametric sequential prediction of time series. *Journal of Nonparametric Statistics 22,* 3 (4), 297–317.

BLUM, A. AND KALAI, A. 1999. Universal portfolios with and without transaction costs. *Machine Learning 35,* 3, 193–205.

BOLLERSLEV, T. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics 31,* 3 (April), 307–327.

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.

BOX, G. E. P., JENKINS, G. M., AND REINSEL, G. C. 1994. *Time Series Analysis: Forecasting & Control*, 3rd edition ed. Prentice Hall, Englewood Cliffs, NJ, USA.

COVER, T. M. 1991. Universal portfolios. *Mathematical Finance 1,* 1, 1–29.

COVER, T. M. AND GLUSS, D. H. 1986. Empirical bayes stock market portfolios. *Advances in applied mathematics 7,* 2, 170–181.

COVER, T. M. AND ORDENTLICH, E. 1996. Universal portfolios with side information. *IEEE Transactions on Information Theory 42,* 2, 348–363.

EDWARDS, R. D., MAGEE, J., AND BASSETTI, W. H. C. 2007. *Technical Analysis of Stock Trends*. CRC Press, Boca Raton, FL, USA.

ELTON, E. J., GRUBER, M. J., BROWN, S. J., AND GOETZMANN, W. N. 2003. *Modern Portfolio Theory and Investment Analysis*. J. Wiley & Sons, New York, NY, USA.

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

GRAHAM, B. AND DOOD, D. L. 1996. *Security Analyis*. Whittlesey House, New York, NY, USA.

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

GYÖRFI, L. AND SCHÄFER, D. 2003. Nonparametric prediction. In *Advances in Learning Theory: Methods, Models and Applications*, J. A. K. Suykens, G. Horváth, S. Basu, C. Micchelli, and J. Vandevalle, 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. *Statistics and Decisions 26,* 2, 145–157.

GYÖRFI, L., URBÁN, A., AND VAJDA, I. 2007. Kernel-based semi-log-optimal empirical portfolio selection strategies. *International Journal of Theoretical and Applied Finance 10,* 3, 505–516.

HAZAN, E. 2006. Efficient algorithms for online convex optimization and their applications. Ph.D. thesis, Princeton University.

HAZAN, E. AND SESHADHRI, C. 2009. Efficient learning algorithms for changing environments. In *ICML*. ACM, New York, NY, USA, 50.

HELMBOLD, D. P., SCHAPIRE, R. E., SINGER, Y., AND WARMUTH, M. K. 1998. On-line portfolio selection using multiplicative updates. *Mathematical Finance 8,* 4, 325–347.

KALAI, A. AND VEMPALA, S. 2002. Efficient algorithms for universal portfolios. *The Journal of Machine Learning Research 3*, 423–440.

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

MAGDON-ISMAIL, M. AND ATIYA, A. 2004. Maximum drawdown. *Risk Magazine 10*, 99–102.

MARKOWITZ, H. 1952. Portfolio selection. *The Journal of Finance 7,* 1, 77–91.

MOODY, J., WU, L., LIAO, Y., AND SAFFELL, M. 1998. Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting 17*, 441–471.

ORDENTLICH, E. AND COVER, T. M. 1996. On-line portfolio selection. In *COLT*. ACM, New York, NY, USA, 310–313.

RODGERS, J. L. AND NICEWANDER, A. W. 1988. Thirteen ways to look at the correlation coefficient. *The American Statistician 42,* 1, 59–66.

SHARPE, W. F. 1994. The sharpe ratio. *Journal of Portfolio Management 21,* 1, 49 – 58.

TIMMERMANN, A. AND GRANGER, C. W. J. 2004. Efficient market hypothesis and forecasting. *International Journal of Forecasting 20,* 1, 15–27.

TSAY, R. S. 2005. *Analysis of Financial Time Series*. Wiley, Hoboken, NJ, USA.