Kalman Filtering and Online Learning Algorithms for Portfolio Selection

Raphael Nkomo and Alain Kabundi

ERSA working paper 394

November 2013

Economic Research Southern Africa (ERSA) is a research programme funded by the National Treasury of South Africa.

Kalman Filtering and Online Learning Algorithms for Portfolio Selection

Raphael Nkomo* Alain Kabundi†

November, 2013

Abstract

This paper proposes a new online learning algorithms for portfolio selection based on alternative measure of price relative called the Cyclically Adjusted Price Relative (CAPR). The CAPR is derived from a simple state-space model of stock prices and we prove that the CAPR, unlike the standard raw price relative widely used in the machine literature, has well defined and desirable statistical properties that makes it better suited for nonparametric mean reversion strategies. We find that the statistical evidence of out-of-sample predictability of stock returns is stronger once stock price trends are adjusted for high persistence. To demonstrate the robustness of our approach we perform extensive historical simulations using previously untested real market datasets. On all the datasets considered, our proposed algorithms significantly outperform their comparative benchmark allocation techniques without any additional computational demand or modeling complexity.

JEL Classification Numbers: C3, E32

Keywords: Online Learning, Portfolio Selection, Kalman Filter, Price Relative.

1 Introduction

The conventional investment wisdom over the last several decades has been to buy-and-hold good-quality stocks for the long run with the hope that securities markets will rise over time. It has been generally believed that securities markets were extremely

*Corresponding Author: Department of Economics and Econometrics,University of Johannesburg, South Africa. E-mail: raphaeln@uj.ac.za

†Department of Economics and Econometrics, University of Johannesburg, Aucland Park, Johannesburg, South Africa. E-mail: akabundi@uj.ac.za.

efficient in reflecting information not only about individual stocks but also about the stock market as a whole. The accepted view was that when information arises, the news spreads very quickly and is incorporated into the prices of securities without any material delay. Thus, neither technical analysis, nor even fundamental analysis would enable an investor to achieve returns greater than those that could be obtained by holding a randomly selected portfolio of individual stocks, at least not without taking additional risk.

However, the amount of evidence showing the disadvantage that traditional, long-term, buy-and-hold investors currently face is staggering. After recent bear markets and the resulting poor performance delivered by most fund managers, there is renewed search for reliable active investment strategies that can outperform not only the market but also the best stock.

In recent years the growth of theoretically well grounded algorithms for online portfolio selection problems has been significant. These algorithms have demonstrated good finite sample properties with a performance that generally exceeds both the market and the best stock even after accounting for transaction costs. More specifically algorithms such as Universal Portfolio (Cover [1991]), Exponential Gradient (Hembold et al. [2003]) and Online Newton Step (Agarwal et al. [2006]) have demonstrated that the wealth achieved through sequential rebalancing strategy possesses explicit lower bounds given sufficiently long period of time. Although very elegant in their mathematical formulation these state of the art algorithms have displayed very disappointing performance in practical applications when compare to some alternative algorithms derived from simple heuristics like the Anticor algorithm of Borodin et al. (2006). More particularly empirical simulations using real market data sequences have shown that algorithms which has no theoretical guarantees have perform rather surprisingly well.

This paper presents a new heuristic approach to online portfolio selection within the context of algorithmic trading. We build on the existing Anticor (Borodin et al. (2006)) algorithm for portfolio selection but use ideas from signal processing and statistical learning to demonstrate the superiority of our new methodology. In a sense our algorithm extends the Anticor (AC) algorithm along many lines.

First we generalize the Anticor to account not only for price reversal but also for momentum that seem to coexist in the market place as demonstrated by an extensive body of behavioral finance literature (see Conrad and Kaul [2006]). This coexistence of both momentum and reversal means that our algorithm is capable of generating abnormal returns from these stock market peculiarities much better than the Borodin et. al (2006) benchmark Anticor algorithm that focuses only on mean reversals.

Second unlike most online learning algorithms that use the raw price relative as the sole input in the program trading, we propose an alternative measure of price relative that is more consistent with portfolio manager's best practice. We use a state-space model via the Kalman Filter algorithm to filter price-cycle oscillations out of the current share prices and compute the cyclically adjusted price relative (CAPR in short). The CAPR helps to de-noise the stock price data in other to account for the possibility of multi period mean reversion in stock prices as argued by Li et al. (2012). To our knowledge this is the first time that a research has combined ideas from signal processing with online learning algorithms to select portfolios in an optimal way. To demonstrate the usefulness of our methodology we evaluate our algorithm against some benchmark online portfolio allocation techniques using previously untested market datasets (including South Africa). Our algorithm substantially outperforms existing online stock selection techniques without much additional computational demand or modeling complexity.

The rest of the paper is organized as follows. In Section 2 the mathematical model is described, and related results are surveyed briefly. In Section 3, we introduce the concept of Cyclically Adjusted Price Relative (CAPR) for online portfolio selection problems. Section 4 presents our new online portfolio selection algorithms together with a pseudo code. Numerical results based on various data sets are described in Section 5 and section 6 draws some concluding remarks for our work.

2 Mathematical Model

The stock market model considered in this paper is the same as investigated by amongst others Gyorfi et al. [2007] and Algoet [1996]. Consider a market of m assets such that a market vector $x = (x^1, x^2, ..., x^m) \in R_m^+$ is the vector of m numbers representing price relatives for a given trading period. The j^{th} component $x^j \geq 0$ of x expresses the ratio of two consecutive closing prices of asset j, such that $x_{t,i} = \frac{p_{t,i}}{p_{t-1,i}}$ and each element $p_{t,i}$ represents the closing price of asset i on period t and $p_{t-1,i}$ is the closing price of stock i on period $t-1$. Thus, an investment in asset i on period t increases by a factor of $x_{t,i}$.. The investor distributes his capital at the beginning of each trading period according to a portfolio vector $b = (b^1, b^2, ..., b^m)$, where the j^{th} component b^j of b denotes the proportion of the investor's capital invested in asset j. Throughout the paper we assume that the portfolio vector b has non negative components which means that the investment strategy is self- financing and that both consumption of capital and short selling is not permitted. Mathematically this simply means that $\sum_{j=1}^{m} b^j = 1$ and $b^j \geq 0$.

Let S_0 denote the investor's initial capital and S_1 the investor's wealth at the end of the first trading period

$$S_1 = S_0 \sum_{j=1}^{n} b^j x^j = \langle b, x \rangle \tag{1}$$

where $\langle .. \rangle$ denotes the inner product.

The evolution of the market in time is represented by a sequence of market vectors x_1,x_2,...,$x_t \in R_m^+$ where the j^{th} component x_i^j of x_i denotes the amount obtained after investing a unit capital in the j^{th} asset on the i^{th} trading period. We denote by $b\left(x_1^{i-1}\right)$ the portfolio vector chosen by the investor on the t^{th} trading period, upon observing the past behavior of the market. b_1 is a constant uniform portfolio vector, usually $b_1 = \left(\frac{1}{m}, ..., \frac{1}{m}\right)$.

Starting with an initial wealth S_0, after n trading periods, the investment strategy B achieves the wealth

$$\mathbf{S}_n = \mathbf{S}_0 \prod_{i=1}^{n} \left\langle b_i\left(x_1^{i-1}\right), x_i \right\rangle \ = \ \mathbf{S}_0 \exp\left\{ \sum_{i=1}^{n} \log \left\langle b_i\left(x_1^{i-1}\right), x_i \right\rangle \right\} \tag{2}$$

$$\mathbf{S}_n = S_0 \exp\left\{ n W_n\left(B\right) \right\} \tag{3}$$

where $W_n\left(B\right)$ denotes the average growth rate and is given by

$$W_n\left(B\right) = \frac{1}{n} \sum_{i=1}^{n} \log \left\langle b_i\left(x_1^{i-1}\right), x_i \right\rangle . \tag{4}$$

This is essentially a log utility function whose expected value needs to be maximized given a suitable choice of nonnegative portfolio vectors $b_i\left(x_1^{i-1}\right)$. Therefore maximizing $S_n = S_n\left(B\right)$ is equivalent to maximizing the average growth rate whose expression is given by

$$W_n\left(B\right) : b_i^*\left(x_1^{i-1}\right) = \arg_b \max E\left\{ \log \left\langle b_i\left(x_1^{i-1}\right), x_i \right\rangle \mid x_1^{i-1} \right\} \tag{5}$$

This is a nonlinear convex optimization problem for which closed-form solutions are not easily available. The search for an acceptable solution has lead many researchers in the machine-learning community to suggest various deterministic or randomized rules that explicit determine a sequence of portfolios weights with the aim of maximizing the investor's wealth without prior knowledge of the statistical distribution of stock prices. The current research study proposes one such algorithm.

2.1 Benchmark Portfolio Selection Algorithms

The complexity of equation (5) makes it very hard to find closed form solutions for the portfolio selection problem unless some structure is imposed a priory on the evolution of the portfolio weights. This certainly explains why market practitioners have adopted rather simplistic but intuitive portfolio selection rules in an attempt to derive optimal portfolio weights. The most basic of portfolio selection algorithms is the so called Buy and Hold (BAH_b) which buys stocks using some portfolio b. This algorithm invests according to b on the first trading day, and then never re-invests any money after that. This results in a portfolio sequence given by

$$b_{i+1} = \frac{\langle x_i, b_i \rangle}{\sum_{i=1}^{m} b_i x_i}. \tag{6}$$

Of course there is an optimal Buy-and-Hold strategy, called BAH_{b^*} that can achieve maximal growth although in hindsight. The BAH_{b^*} is simply a portfolio that assigns a weight of 1 to the best stock, and a weight of 0 to all others. Mathematically this is expressed as

$$b^* = \underset{b(.)}{\arg\max}\, ret_x \left(BAH_b\right) \tag{7}$$

where $ret_x \left(BAH_b\right)$ is simply the returns achieved by the Buy-and-Hold given the sequence of price relative x. When b is set such that the total available investment is initially equally distributed amongst various assets $b = \left(\frac{1}{m}, \frac{1}{m}, ..., \frac{1}{m}\right)$, the BAH_b is referred to as the Uniform Buy-and-Hold or $UBAH_b$. The BAH_b strategy has at least two major advantages. First the Buy-and-Hold strategy does not incur any transaction costs after the initial trade allocation is made given that no trade is generated after that. The second advantage is that this strategy does not induce any market impact and other stock market frictions irrespective of the portfolio size. The major drawback of the BAH_b is the over reliance on the tendency of stock markets to grow over time. In the event of severe market correction as was the case in 2008 this strategy can suffer severe losses.

An alternative approach to the static buy-and-hold is to dynamically change the portfolio during the trading period. In this case the algorithm maintains a fixed portfolio weights throughout the entire trading period by actively re-investing every portfolio gains at the end of each trading day. One example of an active trading strategy is the Constant Rebalancing methodology which fixes a portfolio b and (re)invest each available capital according to b. We denote this constant rebalancing strategy by $CBAL_b$ and let $CBAL_{b^*}$ denote the optimal (in hindsight) $CBAL$ where b is given by

$$b^* = \arg\max_{b(.)} ret_x\left(CBAL_b\right) \quad (8)$$

A major benefit of the constant rebalancing strategy lies in its ability to take advantage of market fluctuations to achieve a return sometime significantly greater than that of BAH_{b^*} although this might come at the expense of much higher transactions cost. $CBAL_{b^*}$ is always at least as good as the best stock and BAH_{b^*} $\left(ret_x CBAL_{b^*} \geq ret_x BAH_{b^*}\right)$, and in some real market sequences a constant rebalancing strategy will take advantage of market fluctuations and significantly outperform the best stock. (see Borodin et al. [2006]).

2.2 Machine Learning Portfolio Selection Algorithms

Machine learning methods for stock selection encompass a variety of trading strategies for which the stock selection algorithm operates in rounds. On round t the algorithm receives a vector of price relatives $x_t \in R_+^m$ to which it applies its current prediction rule to produce a vector of portfolio weights b_t. At time $t+1$ after the initial prediction has been made the portfolio manager receives the new stock price returns x_{t+1} and incur a portfolio period returns equal to $b_t^\intercal x_{t+1}$ after which he updates the total wealth as $S_{t+1} = S_t\left(b_t^\intercal x_{t+1}\right)$. With the new wealth determined the portfolio manager updates his prediction rule and proceeds to the next round. Of course the goal of the portfolio manager is to maximize his total wealth in the long run without any prior knowledge of the statistical distribution of the stock prices. This typical problem specification has its origins in the perceptron algorithm of Rosenblatt [1958] and has now been studied extensively in the machine learning community with various degrees of success.

Several approaches to online portfolio selection have been proposed in the machine learning literature. Cover [1991] for example proposed Universal Portfolios (UP) strategy that weights all constant rebalanced portfolios, referred to as experts, by their empirical probability distribution generated from the performance of each expert. The regret achieved by Cover's UP is $O\left(m \log n\right)$, where m denotes the number of stocks and n denotes the number of trading days. However, the implementation cost of the UP algorithm is exponential in the number of stocks and thus restricts the number of assets used in real market experiments. Although Kalai and Vempala [2002] presented a time-efficient implementation of Cover's UP based on non-uniform random walks, the performance of the UP algorithm has not been satisfactory enough in historical simulations.

The Exponential Gradient strategy of Helmbold et al. [1996, 1997] for online portfolio selection proposes a portfolio selection algorithm using multiplicative updates. To achieve this, the EG strategy tries to maximize the expected logarithmic portfolio daily return (approximated using the last price relative), and minimize the deviation between next portfolio and last portfolio. One straightforward interpretation of the EG algorithm is that it tends to track the stock with the best performance in last period but keep the new portfolio close to the previous portfolio weights.

Borodin et al. [2004] propose a non-universal but empirically robust portfolio strategy named Anti-Correlation (Anticor or simply AC). The Anticor strategy simply takes advantage of the statistical properties of mean-reverting stock prices where the underlying motivation is to bet on the consistency of positive lagged cross-correlation and negative autocorrelation. Although it does not provide any theoretical guarantee, empirical results showed that Anticor can outperform most existing state of the art strategies in real market historical data.

Györfi et al. [2006] introduced a framework of Nonparametric Kernel-based learning strategies for portfolio selection based on nonparametric prediction techniques of Györfi and Schäfer [2003]. Their algorithm first identifies a list of similar historical price relative sequences whose Euclidean distance with the recent market window is smaller than a threshold, and then optimizes the portfolio with respect to the list of observed realized returns following instances of similarity. Following the same line the Nonparametric Nearest Neighbor learning (NN) strategy proposed in Györfi et al. [2008] aims to search for the nearest neighbors in historical price relative sequences rather than search price relatives within a specified Euclidean ball.

Recently, Li et al. [2012, 2013] proposed the Confidence Weighted Mean Reversion (CWMR) and the Passive Aggressive Mean Reversion (PAMR) strategies. These algorithms actively exploit the mean reversion property and the second order information of a portfolio. Their algorithms have been empirically shown to be a robust trading strategy as they outperform many earlier state of the art algorithms in historical simulations.

3 CAPR in Online Portfolio Selection Algorithms

The generally accepted practice in published state of the art algorithms (see Borodin et al. [2004]; Li et al.[2012])) is to use the raw price relative as the main argument in the machine learning algorithm. Raw price relative is defined as the ratio of two consecutive closing prices $x_t = \frac{p_t}{p_{t-1}}$.at time t . However, raw price relatives and their logarithms are notoriously volatile time series and there is very little to believe that the mean reversion characteristics will be effective in the very next periods, if at all. While the academic research community has settled on raw price relative as the primary input argument, market practitioners have always recognized the need for smoothing stock prices before proceeding to any credible statistical analysis.

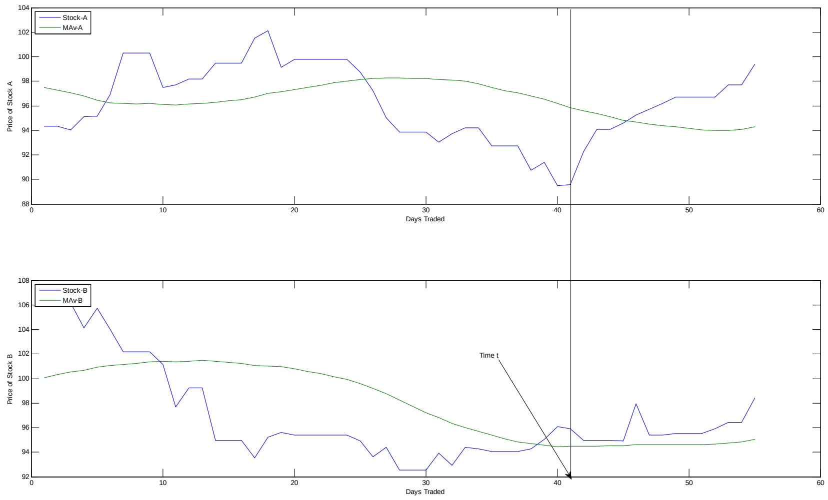
Many stochastic processes, including stock prices, have inherent noise that obscure the true underlying values. To be successful in today's market place, portfolio managers need to see through all the market noise that occurs on a daily basis and be able to identify the trend of prices. Classical approaches to this problem have been to apply a moving average, as in Li et al. (2012), or an exponential moving average to the time-series to obtain a smoothed one We believe that it is potentially risky to use simple moving averages on time-series because they most likely will change the statistical properties of the time-series under consideration.

Appropriate smoothing of price data can eliminate some of the market noise and allow the portfolio manager to focus on trading. In order to reduce the impact of these noises in smoothing stock prices we use the scalar Kalman Filter (KF). The KF is an optimal filter that provides us with clearer resolution and allow us to isolate the peak excursions when the stock price significantly departs from its "true" unobserved component. When the stock price significantly departs from it's trend price we anticipates that the move is unsustainable and takes the view that short-term price will reverse from these peaks. Therefore, critical to our analysis is not how much the price moves from one period to the next as assumed by comparable studies, but instead, how far apart is the stock price from its Kalman trend.

To illustrate why CAPR and raw price relative can lead to two very distinct conclusions we plot in Figure 1 two actively traded stock prices in the Johannesburg stock exchange together with their simple moving average. There are clearly periods where both stocks lie above, below or in opposite side of their respective moving averages. On the surface, it seems as though the higher the stock price is from its moving average, the more bullish the market is (and the lower it goes, the more bearish). In practice, however, the reverse is generally true. Extremely high readings are a warning that the market may soon reverse to the downside. High readings reveal that traders are far too optimistic. When this occurs, fresh new buyers are often few and far between. Meanwhile, very low readings signify the reverse; the bears are in the ascendancy and a bottom is near.

At time t for example (see Figure 1) stock A is below its moving average while stock B lies above its trend. At time $t + 1$, stock A rises by about 3% while stock B falls by 1%. Given this outperformance of stock A over B most online learning algorithms exclusively based on mean reversion would likely start transferring wealth from stock A to stock B depending on their speed of adjustment. However this is a very misleading interpretation of the dynamics of these stocks as stock A comes from a very over sold position and is simply "catching up". Our proposed algorithm acknowledges this fact and will instead recommend transferring more wealth from stock B to stock A. As a consequence the distance between a stock price and its moving average (or trend) is of critical importance in our algorithm.

Figure 1: Impact of smoothing on two stock prices



In this paper we use the Kalman Filter, instead of a simple moving average, as the appropriate filtering methodology since it is robust to noise measurements. Using the Kalman Filter helps us to filter out very volatile and cyclical components of stock prices and derive what we refer to as the cyclically adjusted price relative (CAPR). If p_t represents the stock price and p_t^k the filtered or unobserved "true" price (as defined below) at time t, the cyclically-adjusted price relative is expressed as

$$CAPR = \frac{p_t}{p_t^k}$$

In summary the CAPR ratio is used in this paper to judge how over sold or over bought the stock of a corporation is relative to its own unobserved true price. The further away the price relative is from its own trend the more attractive the stock will be for purchase or for sale in our model. Because our algorithm takes a bet only when a given stock price significantly deviates from its trend, this new measure of price relative is therefore expected to yield better mean reversion characteristics compared to traditional counterparts.

3.1 Some Statistical Properties of the CAPR

Let us assume that the stock price process is governed by a state-space representation where the measurement equation is given by $p_t = Mp_t^k + v_t$, where M is known and $v_t \sim N(0, R)$ with R known. This equation essentially describes the relationship between observed stock price p_t and the unobserved "true" stock price p_t^k. Let us also assume that the transition equation follows autoregressive AR(1) process given by the equation $p_t^k = \phi p_{t-1}^k + w_t$, in this equation $|\phi| < 1$ is assumed known and $w_t \sim \mathrm{N}(0, Q)$ with Q known.

The ratio $\frac{p_t}{p_t^k} = M + \frac{v_t}{p_t^k}$, represents our measure of the cyclically adjusted price relative (CAPR) centered around the known coefficient M. CAPR is dominated by the behavior of the ratio $\frac{v_t}{p_t^k}$ whose statistical properties could be defined in the following way .

From the specification of the transition equation it is easy to demonstrate that the mean $E\left(p_t^k\right) = 0$ and the variance $Var\left(p_t^k\right) = \frac{Q}{1-\phi^2}$ therefore $p_t \sim N\left(0, \frac{Q}{1-\phi^2}\right)$

In Appendix A we prove (using the Taylor expansions around $g(.)$) that given two random variables R and S where S has support $[0, \infty[$ the function $G = g(R, S) = \frac{R}{S}$, has the following approximates for $E(G)$ and $Var(G)$.

$$E\left(\frac{R}{S}\right) = \frac{ER}{ES} - \frac{Cov(R, S)}{E^2 S} + \frac{Var(S)\, ER}{E^3 S} \tag{9}$$

$$Var\left(\frac{R}{S}\right) \approx \frac{E^2 R}{E^2 S}\left[\frac{Var(R)}{E^2 R} - 2\frac{Cov(R, S)}{ER\ ES} + \frac{Var(S)}{E^2 S}\right] \tag{10}$$

We therefore provide the following expression for the mean and variance of the ratio $\frac{p_t}{p_t^k}$

$$E\left(\frac{p_t}{p_t^k}\right) = M \tag{11}$$

and

$$Var\left(\frac{p_t}{p_t^k}\right) = aVar(v_t) + bVar(p_t^k),\ where\ a = \frac{E^2 v_t}{E^2 p_t^k E^2 v_t}, b = \frac{E^2 v_t}{E^2 p_t^k E^2 p_t^k} \tag{12}$$

The ratio of the observed stock price to its unobserved component (Kalman trend) is such that

$$\frac{p_t}{p_t^k} \sim N\left(M, aR + b\frac{Q}{1-\phi^2}\right) \tag{13}$$

This last expression demonstrates that the statistical properties of the cyclically adjusted price relative $\frac{p_t}{p_t^k}$ are well established and that in a sense assures better mean reversion characteristics. This finding is the main motivation why we would expect historical simulations that use the new price relative measure to outperform the base line models that rely on raw price relative as used as default in most empirical analyses. In the rest of the paper, price relative will therefore need to be understood as observed prices relative to the Kalman trend prices and not the ratio of two consecutive closing prices.

3.2 The Scalar Kalman Filter Algorithm

The Kalman Filter is a recursive algorithm that produces estimates of a time series of unobservable variables (along with parameter estimates for the theoretical model that generates the data) using a related but observable time series of variables. The estimates of the unobservable variables are updated at each time step based on the revelation of new observable data. The Kalman filter uses the current observation to predict the next period's value of unobservable and then uses the realization next period to update that forecast. The linear Kalman filter is optimal, i.e. minimum Mean Squared Error estimator if the observed variable and the noise are jointly Gaussian. Suppose $p_1, p_2, ..., p_t$ is the observed values of the stock prices for a given firm at time $1, 2, ..., t$. We assume that p_t depends on an unobservable quantity p_t^k, known as the state of nature or the true stock price.

The Kalman Filter recursive estimation algorithm works as follows. At time t_0 the process start with an initial estimate p_0^{k*} for p_t^k which has a mean of μ_0 and a variance

$$P = E\left(p_0^k - p_0^{k*}\right)^2$$

At time t_1 and before any measurement is taken (or before any stock price is revealed) the state price is given by

$$\overline{p}_1^k = \phi p_0^{k*}$$

its variance is given by

$$\overline{P} = E\left(p_1^k - \overline{p}_1^k\right)^2 = E\left[\phi p_0^k + \mu_0 - \phi p_0^{k*}\right]^2 = \phi^2 P + Q,$$

and the transition equation is given by

$$\overline{p}_1 = M\overline{p}_1^k$$

Still at time t_1 and after the measurement p_1 becomes available

$$p_1^{k*} = \overline{p}_1^k + K\left[p_1 - \overline{p}_1\right] = \overline{p}_1^k + K\left[p_1 - M\overline{p}_1^k\right]$$

where K is the kalman gain. After p_1 becomes the variance of the measurement needs to be updated in the following way.

$$P = \left[p_1^k - p_1^{k*}\right]^2 = \left[p_1 - \overline{p}_1^k - K\left[p_1 - M\overline{p}_1^k\right]\right]^2$$

$$P = \left[p_1^k - \overline{p}_1^k - K\left[Mp_1^k + w_1 - M\overline{p}_1^k\right]\right]^2 = \overline{P}\left(1 - KM\right)^2 + RK^2$$

The value of the kalman gain (K) that minimizes the variance is

$$K = M\overline{P}\left(\overline{P}M^2 + R\right)^{-1}$$

4 CAPR and Online Learning Algorithms for Portfolio Selection

In this section we demonstrate the merit of our methodology using a generalized version of the Anticor algorithm of Borodin et al [2004].

4.1 The Anitcor (AC) algorithm Revisited

In its original form, the Anticor (AC) algorithm of Borodin et al. [2006] provides results on historical stock prices that show that some an algorithm derived from simple heuristics can significantly outperform those that provide theoretical guarantees Unlike competing state of the art algorithms that are derived from sound mathematical and statistical learning theory, the AC algorithm is derived from simple heuristics. The algorithm evaluates changes in stocks' performance by dividing the historical sequence of past returns series into equal-sized periods called windows, each with a length of w days where w is an adjustable parameter. According to the AC algorithm the wealth is transferred from recently high-performing experts to anti-correlated low-performing experts. Specifically, whenever the algorithm detects that stock i outperformed stock j during the last window, but $i's$ performance in the last window is anti-correlated to $j's$ performance in the second-to-last window ($\mu_2(i) \geq \mu_2(j)$ and $Mcorr(i,j) > 0$), it transfers wealth from stock i to stock j and calculate new portfolio weights. Borodin et al. [2006] show historical simulation results or some real market datasets that demonstrate that the AC algorithm is indeed very robust in those datasets based solely on the mean reversion principal.

Despite this impressive empirical performance, an extensive body of behavioral finance literature has documented that price reversals is hardly the only feature at play in equity markets. It has been argued that price momentum and reversals tend to coexist in world stock markets in short term. In a comprehensive investigation, Conrad and Kaul [2006] find both momentum and contrarian profits in the U.S. market, depending on the time horizon investigated. Balvers and Wu [1998] also demonstrate that mean reversion and momentum can simultaneously occur on the same set of assets in 18 developed countries. This coexistence of both momentum and reversal means that an exclusive focus on mean reversion is likely to generate suboptimal results. It is therefore possible that in the presence of both price reversal and price continuation the original AC algorithm will fail to perform optimally. To correct this shortcoming we provide an important modification to the AC algorithm that can deal with both momentum as well as reversal in the following way.

As in Borodin et al. [2006], for a window length w, we consider LX_1 and LX_2 as two $w \times n$ matrices over two consecutive time windows which we compute as follows

$LX_1 = (\log(X_{t-2w+1}), ..., \log(X_{t-w}))^T$ and $LX_2 = (\log(X_{t-w+1}), ..., \log(X_t))^T$

The j^{th} column of LX_k is denoted by $LX_k(j)$ and simply tracks the performance of stock j in window k where $k = 1, 2$. Let $\mu_k(j)$ be the mean of $LX_k(j)$ and $\sigma_k(j)$ be the corresponding standard deviation. The cross-covariance matrix between the columns vectors of LX_k is defined as follows $Mcov(i,j) = \frac{1}{w-1}[LX_1(i) - \mu_1(i)]^T [LX_2(j) - \mu_2(j)]$ and the corresponding cross-correlation matrix is given by

$$MCorr(i,j) = \begin{cases} \frac{MCov(i,j)}{\sigma_1(i)\sigma_2(j)}, & \sigma_1(i), \sigma_2(j) \neq 0 \\ 0 & otherwise \end{cases}$$

The reversion to mean strategy of Borodin et al. [2006] states that if $\mu_2(i) \geq \mu_2(j)$ and $Mcorr(i,j) > 0$ the proportion of wealth to be moved from stock i to stock j is defined as

$$claim_{i \to j} = Mcorr(i,j) + \max(-MCorr(i,i), 0) + \max(-MCorr(j,j), 0)$$

To account for the possibility of short term momentum we expand the benchmark AC algorithm as follows. If $\mu_2(i) \geq \mu_2(j)$ and $Mcorr(i,j) \leq 0$ the proportion of wealth to be moved from stock i to stock j is defined as

$$claim_{i \to j} = -Mcorr(i,j) + \max(MCorr(i,i), 0) + \max(MCorr(j,j), 0)$$

Therefore whenever our algorithm detects that stock i outperformed stock j during the last window, but $i's$ performance in the last window is not anti-correlated to $j's$ performance in the second-to-last window ($\mu_2(i) \geq \mu_2(j)$ and $Mcorr(i,j) \leq 0$), it transfers wealth from stock j to stock i (the model is adding into the holding of stock i) and calculate new portfolio weights. The simple logic here is that there will be price continuation in the direction of the outperforming stock.

From both the reversal to the mean and the price continuation conditions we calculate the transfers of stock i to stock j as

$transfer_{i\to j} = b_{t-1}\left(i\right) \frac{claim_{i\to j}}{\sum_j claim_{i\to j}}$

Using these transfer values, the portfolio is defined to be

$b_t\left(i\right) = b_{t-1}\left(i\right) + \sum\limits_{i\neq j}\left(transfer_{j\to i} - transfer_{i\to j}\right)$

and we call the resulting algorithm the Momentum-Anticor or ACM. In Table 1 below, we show a pseudo code implementation of the proposed K-ACM algorithm where the input price relative is the CAPR derived from the scalar Kalman Filter procedure.

Table 1: $K - ACM\left(w, \phi, M, P\right)$	
w: window size	
ϕ: autoregressive coefficient of the state equation	
M: coefficient in the measurement equation	
b_0: initial portfolio weights $b_0 = \left(\frac{1}{m}, ..., \frac{1}{m}\right)$	
Initialize the Kalman Filter parameters $\left(Q, R, Z, V\right)$	
for $t = 1, 2, ...$	
1	Estimate the true price Z_t using the following procedure
1.1	$\overline{Z} = \phi\widehat{Z}$
1.2	$\overline{V} = V\phi^2 + Q$
1.3	$K = M\overline{V}\left[\overline{V}M^2 + R\right]^{-1}$
1.4	$\widehat{Z} = \overline{Z} + K\left[P_t - M\overline{Z}\right]$
1.5	$V = \overline{V}\left[1 - KM\right]^2 + RK^2$
2	$CAPR = x_t = \frac{P_t}{\widehat{Z}_t}$
3	Return the current portfolio b_t if $t < 2w$
4	compute $LX_1 = \left(\log\left(X_{t-2w+1}\right), ..., \log\left(X_{t-w}\right)\right)^T$
5	compute $LX_2 = \left(\log\left(X_{t-w+1}\right), ..., \log\left(X_t\right)\right)^T$
6	compute $\mu_1 = average\left(LX_1\right)$ and $\mu_2 = average\left(LX_2\right)$
7	compute $Mcov\left(i,j\right) = \frac{1}{w-1}\left[LX_1\left(i\right) - \mu_1\left(i\right)\right]^T\left[LX_2\left(j\right) - \mu_2\left(j\right)\right]$
8	compute $MCorr\left(i,j\right) = \begin{cases} \frac{MCov(i,j)}{\sigma_1(i)\sigma_2(j)}, & \sigma_1\left(i\right), \sigma_2\left(j\right) \neq 0 \\ 0 & otherwise \end{cases}$
9	Initialize $claim_{i\to j} = 0$
10	if $\mu_2\left(i\right) \geq \mu_2\left(j\right)$ and $Mcorr\left(i,j\right) > 0$
11	$claim_{i\to j} = Mcorr\left(i,j\right) + \max\left(-MCorr\left(i,i\right), 0\right) + \max\left(-MCorr\left(j,j\right), 0\right)$
12	else if $\mu_2\left(i\right) \geq \mu_2\left(j\right)$ and $Mcorr\left(i,j\right) \leq 0$
13	$claim_{i\to j} = -Mcorr\left(i,j\right) + \max\left(MCorr\left(i,i\right), 0\right) + \max\left(MCorr\left(j,j\right), 0\right)$
14	$transfer_{i\to j} = b_{t-1}\left(i\right)\frac{claim_{i\to j}}{\sum_j claim_{i\to j}}$
15	$b_t\left(i\right) = b_{t-1}\left(i\right) + \sum\limits_{i\neq j}\left(transfer_{j\to i} - transfer_{i\to j}\right)$
end	

The pseudo code presented in Table 1 is clearly a generalization of the AC algorithm. Because it accounts for both price reversals and price momentum while accepting the CAPR as the main input we expect our K-ACM to perform significantly better than the original AC algorithm. Further to that Table 1 also allows us to derive new algorithms specification that could form the basis for valid investment strategies. For example if the portfolio manager uses the original AC Algorithm together with the CAPR we refer to this algorithm as the Kalman Anticor (K-AC) algorithm. Similarly if a portfolio manager uses the raw price relative as main argument together with our proposed generalized AC Algorithm we refer to this version as the Momentum Anticor or ACM.

4.2 Combining Portfolio of Experts

All our new Anticor based online portfolio selection algorithms requires an important fine tuning parameter, namely the window length w. This means that for all practical purposes the portfolio manager will need to set w to some level ex ante. In most instances the performance of the algorithm could fluctuate significantly depending on the set window size. Because it is practically impossible to know ex ante what window period will generate better trading performance our approach is simply to select a number of window periods and allow them to compete. In this paper we achieve this by choosing a wider range of parameters, w, where for each window we obtain a set of historical results, called experts, that are dependent on that particular window size. As in Gyorfi et al. [2006] we form a mixture of all experts using a positive probability distribution q_w on the set of all window lengths w of positive integers. The investment strategy simply weights these experts H^w according to their past performances and the q^w such that after the t^{th} trading period the investor wealth becomes

$$S_t = \sum_w q_w S_t \left(H^w\right) \tag{14}$$

where $S_t \left(H^w\right)$ is the capital accumulated after t^{th} trading period using the expert H^w with initial capital $S_0 = 1$. We then form our final portfolio by weighting all expert portfolio using the following

$$b\left(X_1^{t-1}\right) = \frac{\sum_w q_w S_{t-1}\left(H^w\right) h^w\left(x_1^{t-1}\right)}{\sum_w q_w S_{t-1}\left(H^w\right)} \tag{15}$$

There are of course many alternative ways one could choose to combine expert portfolios given by a range of window choices. Borodin et al. (2006) proposed using the Anticor algorithm to combine portfolio experts given an appropriate choice of the window size. However we decided against such a procedure as this will again be dependent on the choice of a new optimal window size that needs to be selected by the portfolio manager. The methodology adopted here has therefore the major advantage that no further parameter tuning is required.

4.3 Trading Costs and Market Impacts

Because our algorithm is likely to generate a large amount of daily transactions and that transaction are not costless, we need to be concerned about the amount of commissions this portfolio will incur in actual market trading. Although these costs have been significantly reduced in recent years due to technological advances and improved market liquidity, transaction costs remain an issue to be carefully analyzed if algorithms are to be trusted for real market applications. In this study we work on the assumption that there are charges on all transactions equal to a fixed percentage of the amount transacted. We adopt the proportional transaction cost model following Blum and Kalai [1999] and Borodin et al. [2004], that is, rebalancing the portfolio in any given day incurs transaction costs for both buy and sells orders.

At the beginning of the t^{th} trading day, the portfolio manager rebalances the portfolio from the previous closing price adjusted portfolio b_{t-1} to a new portfolio b_t. Specifically, we consider a transaction cost rate $c \in (0, 1)$, so the transaction cost will be charged according to

$$\frac{c}{2}\sum_{k} \left| b_{(t,k)} - b_{(t-1,k)} \right| \tag{16}$$

Thus, with transaction cost rate the total wealth achieve by the strategy becomes

$$S_T^c = S_0 \prod_{t=1}^{T} \left[(b_t x_t) \left(1 - \frac{c}{2}\sum_{k} \left| b_{(t,k)} - b_{(t-1,k)} \right| \right) \right] \tag{17}$$

In this paper, transaction costs are therefore taken into account and we assume a round-trip trading cost per trade of 10 basis points, to incorporate an estimate of price slippage and other costs as a single market friction coefficient. All our Anticor-based algorithms also assume that all portfolio adjustments are implemented using the quoted prices and that all transactions are implemented simultaneously using these prices. This is of course an oversimplification of what really happens in actual trading where a time delay is needed between updating of portfolio weights and actual trading. Although recent technological advances have made computerized systems a natural candidate for fast order execution there is still no guarantee that they will be all implemented instantly unless the portfolio manager is happy to cross the bid-ask spread at every round of trading. This trading friction will necessarily generate discrepancies between the model returns and those realized in actual trading.

5 Empirical results

This section presents numerical results obtained by applying the above algorithms (K-ACM, ACM, K-AC and the AC) to six financial market data sets. The back-testing experiments consists in running the signals through historical data, with the estimation of parameters, signal evaluations and portfolio re-balancing performed daily. However before testing the algorithms with data from real financial markets we hilight some assumptions implied in our model that are not found in real markets. As in Gyorfi et al. (2008) we assume that assets are available in any desired quantities at any given price. We also assumed that all trades are done at the closing price of that day and all the wealth achieved in the last period is fully invested in the next one, without any extra investment allowed.

5.1 Data Description

Our empirical experiments perform numerical evaluations on six real datasets by comparing the performance of our proposed algorithms with some benchmark methods. The first four datasets summarized in table 1 are obtained from Bloomberg and cover the period of January 2000 to October 2013 (see Table 1). Bloomberg stock prices are adjusted for stock split splits and dividends have been included in this simulations. To our knowledge this datasets has never been tested and possess the advantage that it covers the subprime crisis period of 2007 to 2009. This datasets covers about 4508 trading days that exclude weekends.

The first datasets is the TOP40 Index which consists of the largest 42 stocks by market capitalization listed in the Johannesburg Stock Exchange in South Africa. This datasets contains price relatives of 4508 trading days, ranging from January 2nd 2000 to October 2013 and is by far the most liquid index available to South African investor. The second datasets, FTSE100 Index, is a share index of the 100 companies listed on the London Stock Exchange with the highest market capitalization. It is one of the most widely used stock indices and is seen as a gauge of business prosperity in the United Kingdom. The third datasets is that of the Toronto Stock Exchange referred to as the S&P/TSE 60 or simply the TSE60 Index. This index represents the stock market index of 60 large companies listed on the Toronto Stock Exchange and it currently exposes the investors to about 10 industry sectors. The fourth datasets is derived from the NAS-

DAQ stock exchange. The NASDAQ100 is a stock market index of 100 of the largest non-financial companies listed on the NASDAQ. It is a modified capitalization-weighted index. The companies' weights in the index are based on their market capitalizations, with certain rules capping the influence of the largest components. It does not contain financial companies, and could include companies incorporated outside the United States.

Table 2: Data Description

Data Set	Region	Time Frame	Trade Days	Stocks
TOP40	SA	JAN-2000 - OCT-2013	4508	42
FTSE100	UK	JAN-2000 - OCT-2013	4508	100
TSE60	CANADA	JAN-2000 - OCT-2013	4508	60
NASDAQ100	US	JAN-2000 - OCT-2013	4508	100
NYSE(O)	US	JUL-1962 - DEC-1984	5651	36
NYSE(N)	US	JAN-1985 - JUN-2009	6179	23

The last two datasets, NYSE (O) and NYSE (N), are from the New York Stock Exchange in the US. This NYSE datasets have been widely analyzed in many previous studies [Cover [1991]; Helmbold et al. [1998]; Borodin et al. [2004]; Agarwal et al. [2006]; Gyorfi et al. [2006; 2008]] and recently by Li et al [2012]. Although not very useful for practical applications this datasets provides a benchmark against which one can compare historical simulation and testing of empirical data across many previously published state of the art online portfolio selection strategies. The NYSE (O) for example contains 5651 daily price relatives of 36 stocks in the New York Stock Exchange (NYSE) for a 22-year period from July 3rd 1962 to December 31st 1984. The NYSE(N) datasets is the one used by Li et al. (2012) that covers the period from January 1st 1985 to June 30th 2009 and contains 6179 trading days and 23 stocks.

The diverse markets and datasets in our test bed 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. While the first four previously untested datasets are used to test the performance of our methodology on up to date stock markets data the last two datasets are used to empirically evaluate our models on earlier datasets as these have been used extensively in similar studies.

5.2 Paper Trade and Historical Simulation Results

The first experiment evaluates the total wealth achieved by different learning-to trade algorithms including a 10 basis points transaction cost. Table 2 summarizes the total wealth achieved by various algorithms on the six market datasets. The market index is calculated as equal weighted constantly rebalanced portfolio on all stock available for that particular market.

Several observations can be drawn from the results. First of all, we find that all learning-to-trade algorithms can beat both the best stock and the market index on all the datasets. Second, we observe that our proposed algorithms K-ACM performs much better than its respective benchmarks AC and ACM on all datasets and for all period of time as shown in Figure 2.

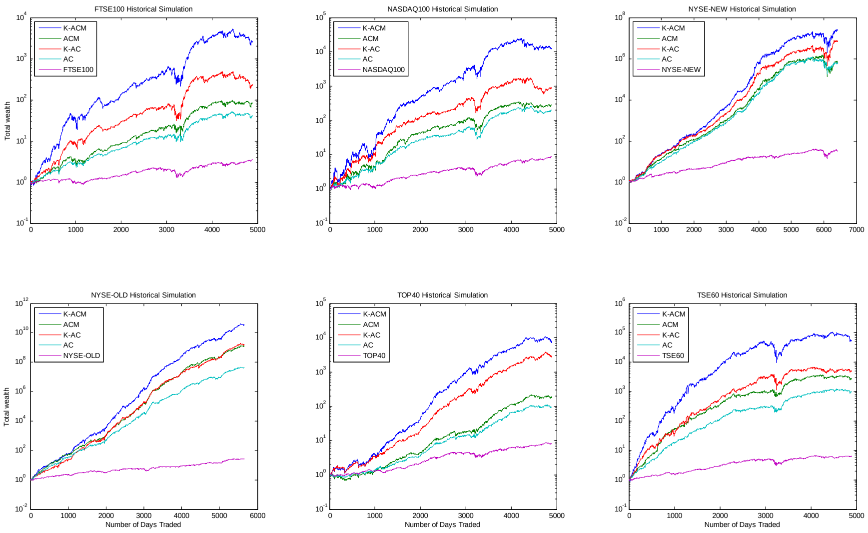
Table 3: Total Wealth achieved by selected algorithms

	FTSE	NASDAQ	NYSE(N)	NYSE(O)	TOP40	TSE60
SIZE	4507	4507	4507	5651	6431	4507
COUNTRY	UK	USA	USA	USA	RSA	CANADA
BEST STOCK	4.1E+01	2.7E+02	8.5E+01	5.3E+01	5.3E+01	2.7E+01
UBAH	3.3E+00	1.0E+01	1.8E+01	1.4E+01	8.3E+00	4.7E+00
UBCRP	3.6E+00	9.1E+00	3.2E+01	2.7E+01	8.3E+00	6.4E+00
BCRP*	4.4E+01	6.6E+02	1.2E+02	2.5E+02	7.3E+01	5.5E+01
K-ACM	2.6E+03	1.3E+04	2.3E+07	3.6E+10	7.8E+03	5.6E+04
ACM	8.4E+01	2.9E+02	6.4E+05	1.3E+09	1.9E+02	2.7E+03
K-AC	2.4E+02	9.1E+02	6.5E+06	1.7E+09	2.9E+03	5.1E+03
AC	4.1E+01	2.0E+02	5.6E+05	4.2E+07	9.6E+01	1.0E+03
MARKET	3.6E+00	9.1E+00	3.2E+01	2.7E+01	8.3E+00	6.4E+00

For example, on the NYSE (N) datasets after trading for 22 years, the total wealth achieved by the AC strategy and the ACM strategy impressively increases from \$1 to almost \$560K and \$640K, respectively, which are much higher than the best stock that achieves \$84.82 and the market index that achieves a mere \$31.82. For the same datasets the performance of the K-AC and K-ACM are even more impressive. In the case of the K-AC for example a 1\$ rises to \$6.5-million after 6431 days of trade while the K-ACM rises to an impressive \$23.0-million during the same time frame. Our propose methodology therefore achieves a growth in wealth that is more than 41 times that achieves by its benchmark algorithm. This dominance of our proposed algorithms relative to their benchmarks is also evident in all datasets and time frames as shown in Table 2. Both

K-AC and K-ACM achieve considerably better results than the market index, the best stock in the market, as well as all their benchmark state-of-the-art strategies. Besides the preceding results, we are also interested in examining how the total wealth achieved by various strategies change over different trading periods. Figure 2 shows the changes of total wealth achieved by the various strategies on the 6 datasets. From the figure, we first observe that our proposed algorithms consistently outperform their benchmark algorithms over all trading periods. It is also interesting to note that, although the market drops sharply due to the financial downturn in 2008, the proposed K-AC and K-ACM algorithms are still able to achieve encouraging returns in some datasets, which is especially more impressive in the later part of the South Africa Top40 datasets. All these impressive results reiterate the efficacy and robustness of the proposed learning-to-trade algorithm.

Figure 2: Performance of Various Online Learning Algorithms



The superior investment performance of our proposed methodology is unsurprising to us as it results from a better and more systematic exploitation of both mean reversion and momentum features of stock prices over short period of time. While existing state of the art algorithms are based on raw price relative as input our algorithm is rather based on the CAPR that has been shown to have not only desirable statistical properties but also seems perfectly in tune with portfolio managers actual trading practices. We do not only hope that stock prices will mean revert in subsequent period, but we only trade at those instances where the stock prices has the highest likelihood to reverse in the short term. Our methodology searches for instances where the stock prices are sufficiently distant from their unobserved true price before initiating a trade. This search for maximum excursion means that the probability of mean reversion is likely to be much higher in our case than it is for comparable state of the art benchmark algorithms.

6 Conclusion

We have presented a new online approach to portfolio selection which builds on existing state of the art portfolio selection algorithms. Our algorithm combines powerful online portfolio selection algorithms with ideas from signal processing and statistics to produce portfolios that substantially outperform their benchmark equivalent on real market datasets. Historical simulations have demonstrated that experiments with both K-ACM, the K-AC and the ACM brought to the fore the exceptional empirical performance improvement that a suitable heuristic can achieve over theoretically well-grounded approaches. De-trending price series and using the ratio of stock prices to their Kalman Filter trend improves the performance of the algorithm quite spectacularly in some cases. This impressive performance using data that are widely available casts some doubts on the market efficiency hypothesis at least on its weakest form. Only in the presence of weakly inefficient markets can these algorithms give the very good performance that we show in the examples. As argued by Gyorfi et al. 2006 this superior performance on datasets available to all investors may partially be explained by the fact that the dependence structures of the markets revealed by the proposed investment strategies are quite complex and, even though all information we use is publicly available, the way this information can be exploited remains hidden from most traders.

Appendix: Statistical Properties of Ratio of Random Variables

Consider two random variables R and S where S has support $[0,\infty[$. Let $G = g(R,S) = \frac{R}{S}$, our goal is to find an approximation for $E(G)$ and $Var(G)$ using the Taylor expansion around $g(.)$.

For any function $f(x,y)$ the bivariate first order Taylor expansion about θ is

$$f(x,y) = f(\theta) + f_x^{'}(\theta)(x-\theta_x) + f_y^{'}(\theta)(y-\theta_y) + remainder$$

Let $\theta = (EX, EY)$, the simplest approximation for $E(f(X,Y))$ is therefore

$$E(f(x,y)) = f(\theta) + f_x^{'}(\theta)(0) + f_y^{'}(\theta)(0) + O\left(n^{-1}\right) \approx f(EX, EY)$$

The second order Taylor expansion is

$$\begin{aligned} f(x,y) &= f(\theta) + f_x^{'}(\theta)(x-\theta_x) + f_y^{'}(\theta)(y-\theta_y) \\ &+ \left\{f_{xx}^{''}\left(\theta(x-\theta_x)^2 + 2f_{xy}^{''}(x-\theta_x)(y-\theta_y) + f_{yy}^{''}(\theta)(y-\theta_y)^2\right)\right\} + remainder \end{aligned}$$

So a better approximation is given by

$$E(f(X,Y)) = f(\theta) + \left\{f_{xx}^{''}(\theta)Var(X) + 2f_{x\mathrm{y}}^{''}(\theta)Cov(X,Y) + f_{yy}^{''}(\theta)Var(Y)\right\} + remainder$$

For

$$g = \frac{R}{S}, g_{RR}^{''} = 0, g_{RS}^{''} = -S^{-2}, g_{SS}^{''} = \frac{2R}{S^3}$$

Then $E\left(\frac{R}{S}\right)$ is approximately

$$E\left(\frac{R}{S}\right) \approxeq E(g(R,S)) \simeq \frac{ER}{ES} - \frac{Cov(R,S)}{E^2S} + \frac{Var(S)\,ER}{E^3S}$$

Using the first order Taylor expansion, the variance is

$$\begin{aligned} Var(f(X,Y)) &= E\left\{[f(X,Y) - E(f(X,Y))]^2\right\} \\ &\simeq E\left\{[f(X,Y) - f(EX,EY)]^2\right\} \\ &= E\left\{\left[f_x^{'}(\theta)(X-\theta_x) - f_y^{'}(\theta)(Y-\theta_y)\right]^2\right\} + O(n^{-1}) \\ &\simeq f_x^{'}(\theta)^2 Var(X) + 2f_x^{'}(\theta)f_y^{'}(\theta)Cov(X,Y) + f_y^{'}(\theta)^2 Var(Y) \end{aligned}$$

Since

$$g_R^{'} = S^{-1}, g_S^{'} = -\frac{R}{S^2}$$

$$
\begin{aligned}
Var\left(\frac{R}{S}\right) &\simeq \frac{1}{E^2S}Var\left(R\right) - 2\frac{ER}{E^3R}Cov\left(R,S\right) + \frac{E^2R}{E^4S}Var\left(S\right) \\
&= \frac{E^2R}{E^2S}\left[\frac{Var[R]}{E^2R} - 2\frac{Cov(R,S)}{ER\ ES} + \frac{Var(S)}{E^2S}\right]
\end{aligned}
$$

References

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

[2] ALGOET, P. H. AND COVER, T. M. 1988. Asymptotic optimality and asymptotic equipartition properties of logoptimum investment. The Ann. Probab. 16, 2, 876–898.

[3] BALVERS, R. J, WU, Y., 2006, "Momentum and mean reversion across national equity markets", Journal Empirical Finance, Vol.13, No. 1, pp.24-48.

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

[5] 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.

[6] BREIMAN, L, 1961. Optimal gambling systems for favorable games. Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability, 1:65–78.

[7] CONRAD, J. KAUL, G. 1998, "An anatomy of trading strategies". Review of Financial studies, Vol. 11, pp. 489-519.

[8] COVER, T. M. 1991. Universal portfolios. Mathematical Finance 1, 1, 1–29

[9] DAS, P. AND BANERJEE, A. 2011. Meta optimization and its application to portfolio selection. In Proceedings of International Conference on Knowledge Discovery and Data Mining.

[10] GYORFI, L. AND SCH¨A FER, D. 2003. Nonparametric prediction. In Advances in Learning Theory: Methods, Models and Applications, J. A. K. Suykens, G. Horv´ath, S. Basu, C. Micchelli, and J. Vandevalle, Eds. IOS Press, Amsterdam, Netherlands, 339–354.

[11] GYO RFI, L., LUGOSI, G., AND UDINA, F. 2006. Nonparametric kernel-based sequential investment strategies.Mathematical Finance 16, 2, 337 – 357.

[12] GYO RFI, L., UDINA, F., AND WALK, H. 2008. Nonparametric nearest neighbor based empirical portfolio selection strategies. Statistics and Decisions 26, 2, 145–157.

[13] HELMBOLD, D. P., SHAPIRE, R. E., SINGER, Y., & WARMUTH, M. K. (1996). On-line portfolio selection using multiplicative updates. In Proceedings of the international conference on machine learning (pp. 243–251).

[14] HELMBOLD, D. P., SHAPIRE, R. E., SINGER, Y., & WARMUTH, M. K. (1997). A comparison of new and old algorithms for a mixture estimation problem. Machine Learning, 27(1), 97–119.

[15] HELMBOLD, D. P., SHAPIRE, R. E., SINGER, Y., & WARMUTH, M. K. (1998) On-line portfolio selection using multiplicative updates. Mathematical Finance, 8(4):325–347

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

[17] KANG, J, LIU, M. H., NI, S. X., 2002, "Contrarian and momentum strategies in the China stock market: 1993–2000", Pacific-Basin Finance Journal, Vol.10, pp.243–265.

[18] LI, B., HOI, S. C., ZHAO, P., AND GOPALKRISHNAN, V. 2013. Confidence weighted mean reversion strategy for online portfolio selection. In ACM Transactions on Knowledge Discovery from Data. Vol. 7. 4:1–4:38.

[19] LI, B., ZHAO, P., HOI, S., AND GOPALKRISHNAN, V. 2012. Pamr: Passive aggressive mean reversion strategy for portfolio selection. Machine Learning 87, 2, 221–258.

[20] LI, B. AND HOI, S. C. H. 2012. On-line portfolio selection with moving average reversion. In Proceedings of the International Conference on Machine Learning.

[21] ROSENBLATT, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. Psychological Review, 65, 386–407.

[22] SINGER, Y. 1997. Switching portfolios. International Journal of Neural Systems, 8(4):488–495,.