

# Moving Average Reversion Strategy for On-Line Portfolio Selection<sup>☆</sup>

Bin Li<sup>‡</sup>, Steven C. H. Hoi<sup>†\*</sup>, Doyen Sahoo<sup>†</sup>, Zhi-Yong Liu<sup>‡</sup>

<sup>‡</sup>*Economics and Management School, Wuhan University, Wuhan, P. R. China 430072*

<sup>†</sup>*School of Information Systems, Singapore Management University, Singapore 178902*

<sup>‡</sup>*Institute of Automation, Chinese Academy of Sciences, Beijing, P. R. China 100080*

---

## Abstract

On-line portfolio selection, a fundamental problem in computational finance, has attracted increasing interests from artificial intelligence and machine learning communities in recent years. Empirical evidence shows that stock's high and low prices are temporary and stock price relatives are likely to follow the mean reversion phenomenon. While existing mean reversion strategies are shown to achieve good empirical performance on many real datasets, they often make the *single-period mean reversion* assumption, which is not always satisfied, leading to poor performance in certain real datasets. To overcome this limitation, this article proposes a *multiple-period mean reversion*, or so-called "Moving Average Reversion" (MAR), and a new on-line portfolio selection strategy named "On-Line Moving Average Reversion" (OLMAR), which exploits MAR via efficient and scalable online machine learning techniques. From our empirical results on real markets, we found that OLMAR can overcome the drawbacks of existing mean reversion algorithms and achieve significantly better results, especially on the datasets where existing mean reversion algorithms failed. In addition to its superior empirical performance, OLMAR also runs extremely fast, further supporting its practical applicability to a wide range of applications. Finally, we have made all the data sets and source codes of this work publicly available at our project website: <http://OLPS.stevenhoi.org/>.

**Keywords:** Portfolio Selection, On-line Learning, Mean Reversion, Moving Average Reversion

---

## 1. Introduction

Portfolio selection is a fundamental problem of computational finance extensively explored across several fields, ranging from traditional finance theory and quantitative finance, to machine learning and artificial intelligence [43]. It generally aims to achieve some targets in the long run by sequentially allocating wealth across a set of assets. Two major schools of principles and theories for portfolio selection include: (i) Mean-variance theory [48] that trades off between expected return (mean) and risk (variance) of a portfolio, which is suitable for single-period portfolio

---

<sup>☆</sup>The short version of this work [42] appeared on the 29th International Conference on Machine Learning (ICML2012).

<sup>\*</sup>Corresponding author: S.C.H. Hoi, e-mail: [chhoi@smu.edu.sg](mailto:chhoi@smu.edu.sg); other authors' E-mail addresses: [binli.whu@whu.edu.cn](mailto:binli.whu@whu.edu.cn) (B. Li) [doyen.2014@phdis.smu.edu.sg](mailto:doyen.2014@phdis.smu.edu.sg) (D. Sahoo), [zhiyong.liu@ia.ac.cn](mailto:zhiyong.liu@ia.ac.cn) (Z. Liu).

selection; and (ii) Kelly investment [41, 10, 23] that aims to maximize the expected log return of a portfolio and is naturally available to multiple-period portfolio selection. Due to the sequential nature of a real-world portfolio selection task, many recent on-line portfolio selection techniques often follow the second approach.

One important property exploited by many existing studies [8, 46, 45] is the *mean reversion* property, which assumes poor performing stocks will perform well in the subsequent periods and vice versa. Although some recently proposed mean reversion algorithms [46, 45] have achieved promising results on many real datasets, they might perform extremely poorly on certain datasets, such as the DJIA dataset [8]. Comparing with Borodin et al. [8], which exploits multi-period correlation, we found that the assumption of single-period prediction may attribute to the performance degradation. On the other hand, as illustrated in existing studies [46, 45], Borodin et al. [8] cannot fully exploit the potential of (multi-period) mean reversion for its heuristical nature.

To address the above drawbacks, we present a new approach for on-line portfolio selection, named “On-Line Moving Average Reversion” (OLMAR). The basic idea is to represent multi-period mean reversion as “Moving Average Reversion” (MAR), which explicitly predicts next price relatives using moving averages, and then learns portfolios via online learning techniques [38]. To the best of our knowledge, OLMAR is the first algorithm that exploits moving average in the setting of on-line portfolio selection. Though simple in nature, OLMAR has a reasonable updating strategy and has been empirically validated via a set of extensive experiments on real markets. The experimental results show that OLMAR not only can achieve better performance (in terms of *cumulative wealth*) than existing algorithms, and but also can avoid the significant performance degradation on certain datasets, such as the DJIA dataset [8, 46]. Finally, OLMAR runs highly efficiently, and thus is suitable for large-scale trading applications.

The rest of the paper is organized as follows. Section 2 formulates the on-line portfolio selection problem, and Section 3 reviews the background and analyzes related work. Section 4 presents the proposed OLMAR approach, and its effectiveness is validated by extensive empirical studies on real stock markets in Section 5. Section 6 discusses our empirical findings and some threats to validity. Section 7 summarizes the paper and provides directions for future work.

## 2. Problem Setting

Before we formulate the online portfolio selection problem, we first introduce some notations to be used throughout the article. Symbols in bold are vectors, e.g., we define  $a$  as a scalar, and  $\mathbf{b}$  and  $\mathbf{c}$  as vectors. One typical operation is the product or division between a scalar and a vector, which means each element of the vector will multiply or divide by the scalar, e.g.,  $[a\mathbf{b}]_i = a * b_i$  and  $[\frac{a}{\mathbf{b}}]_i = \frac{a}{b_i}$ . For the element-wise operations between two vectors, we denote by  $\frac{\mathbf{a}}{\mathbf{b}}$  the element-wise division and  $\mathbf{a} \otimes \mathbf{b}$  the element-wise product, which means  $[\frac{\mathbf{a}}{\mathbf{b}}]_i = a_i/b_i$  and  $[\mathbf{a} \otimes \mathbf{b}]_i = a_i * b_i$ , respectively. The dot product between two vectors is defined as  $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^T \mathbf{b} = \sum_{i=1}^n a_i * b_i$ .

Now let us consider an investment task over a financial market with  $m$  assets for  $n$  periods. On the  $t^{\text{th}}$  period, the asset prices are represented by a *close price vector*  $\mathbf{p}_t \in \mathbb{R}_+^m$ , and each element  $p_{t,i}$  represents the close price of asset

---

**Algorithm 1** On-line portfolio selection framework.

---

- 1: **Input:**  $\mathbf{x}_1^n$ : Historical market price relative sequence
  - 2: **Output:**  $S_n$ : Final cumulative wealth
  - 3: **Procedure:**
  - 4: Initialization:  $\mathbf{b}_1 = \frac{1}{m}\mathbf{1}$ ,  $S_0 = 1$ ;
  - 5: **for**  $t = 1, 2, \dots, n$  **do**
  - 6: Portfolio manager learns the portfolio  $\mathbf{b}_t$
  - 7: Market reveals the market price relative  $\mathbf{x}_t$
  - 8: Portfolio incurs period return  $s_t = \mathbf{b}_t^\top \mathbf{x}_t$  and updates cumulative return  $S_t = S_{t-1} \times (\mathbf{b}_t^\top \mathbf{x}_t)$
  - 9: Portfolio manager updates his/her on-line portfolio selection rules
  - 10: **end for**
- 

*i.* The price changes are represented by a *price relative vector*  $\mathbf{x}_t \in \mathbb{R}_+^m$ , and  $x_{t,i} = \frac{p_{t,i}}{p_{t-1,i}}$ . Thus, an investment in asset  $i$  on the  $t^{\text{th}}$  period increases by a factor of  $x_{t,i}$ . Let us denote  $\mathbf{x}_{t_1}^{t_2} = \{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \dots, \mathbf{x}_{t_2}\}$  as a sequence of price relative vectors ranging from period  $t_1$  to  $t_2$ . Therefore,  $\mathbf{x}_1^n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  represents the sequence of price relative vectors over the entire  $n$  periods.

An investment on the  $t^{\text{th}}$  period is specified by a *portfolio vector*  $\mathbf{b}_t = (b_{t,1}, \dots, b_{t,m})$ , where  $b_{t,i}$  represents the proportion of wealth invested in asset  $i$ . Typically, we assume the portfolio is self-financed and no margin/short sale is allowed, therefore each entry of a portfolio is non-negative and adds up to one, that is,  $\mathbf{b}_t \in \Delta_m$ , where  $\Delta_m = \{\mathbf{b}_t : \mathbf{b}_t \in \mathbb{R}_+^m, \sum_{i=1}^m b_{t,i} = 1\}$ . The investment procedure is represented by a *portfolio strategy*, that is,  $\mathbf{b}_1 = \frac{1}{m}\mathbf{1}$  and the following sequence of mappings  $\mathbf{b}_t : \mathbb{R}_+^{m(t-1)} \rightarrow \Delta_m, t = 2, 3, \dots$ , where  $\mathbf{b}_t = \mathbf{b}_t(\mathbf{x}_1^{t-1})$  is the  $t^{\text{th}}$  portfolio given past market sequence of  $\mathbf{x}_1^{t-1} = \{\mathbf{x}_1, \dots, \mathbf{x}_{t-1}\}$ . We denote by  $\mathbf{b}_1^n = \{\mathbf{b}_1, \dots, \mathbf{b}_n\}$  the strategy for  $n$  periods.

On the  $t^{\text{th}}$  period, a portfolio  $\mathbf{b}_t$  produces a *portfolio period return*  $s_t$ , that is, the wealth increases by a factor of  $s_t = \mathbf{b}_t^\top \mathbf{x}_t = \sum_{i=1}^m b_{t,i} x_{t,i}$ . Since we reinvest and adopt price relative, the portfolio wealth would multiplicatively grow. Thus, after  $n$  periods, a portfolio strategy  $\mathbf{b}_1^n$  produces a *portfolio cumulative wealth* of  $S_n$ , which increases the initial wealth by a factor of  $\prod_{t=1}^n \mathbf{b}_t^\top \mathbf{x}_t$ , that is,  $S_n(\mathbf{b}_1^n, \mathbf{x}_1^n) = S_0 \prod_{t=1}^n \mathbf{b}_t^\top \mathbf{x}_t$ , where  $S_0$  is set to \$1 for convenience.

Finally, we formally formulate the on-line portfolio selection procedure, and outline the algorithmic framework in Algorithm 1. In this task, a portfolio manager is a decision maker, whose goal is to produce a portfolio strategy  $\mathbf{b}_1^n$ , aiming to maximize the cumulative wealth  $S_n$ . He/she computes the portfolios sequentially. On each period  $t$ , the manager has access to the sequence of previous price relative vectors  $\mathbf{x}_1^{t-1}$ . Then, he/she computes a new portfolio  $\mathbf{b}_t$  for next price relative vector  $\mathbf{x}_t$ , where the decision criterion varies among different managers. The portfolio  $\mathbf{b}_t$  is scored based on portfolio period return  $s_t$ . This procedure is repeated until the end, and the portfolio strategy is finally scored according to portfolio cumulative wealth  $S_n$ .

It is important to note that we have made several general and common assumptions in the above model:

1. Transaction cost: no commission fees or taxes exists;
2. Market liquidity: one can buy and sell any desired amount, even fractional, at last closing price of any given trading period;
3. Market impact: any portfolio selection strategy shall not influence the market, or other stocks' prices.

Note that although these assumptions are commonly made in many literature studies, they are nontrivial in practice. We will further analyze and discuss their implications and effects in the empirical studies.

### 3. Background Review and Related Work

In this section, we survey the on-line portfolio selection strategy literature. We start by introducing three benchmarks, that is, market strategy, best stock strategy, and constant rebalanced portfolios strategy. After that, we sequentially introduce three main categories of existing approaches, that is, "Follow-the-Winner", "Follow-the-Loser", and "Pattern-Matching" based approaches. Finally, we analyze existing algorithms following the framework of Kelly's investment. Readers are also encouraged to read a more comprehensive survey in [43].

#### 3.1. Benchmarks

The simplest and commonly used baseline is the *Buy-And-Hold* (BAH) strategy, that is, one invests his capital among the market with an initial allocation of  $\mathbf{b}_1$  and holds the portfolio. The manager only buys assets at the beginning of the 1<sup>st</sup> period and does not rebalance in subsequent periods. BAH's cumulative wealth equals to weighted average of assets' returns, or  $S_n(\text{BAH}(\mathbf{b}_1)) = \mathbf{b}_1 \cdot (\otimes_{t=1}^n \mathbf{x}_t)$ . The BAH strategy with uniform initial portfolio  $\mathbf{b}_1 = (\frac{1}{m}, \dots, \frac{1}{m})$  is referred to as *uniform BAH* strategy, which is usually adopted as *market* strategy to produce market index. Another common benchmark is the *Best Stock* (Best) strategy, a special BAH strategy that invests all capital on the best stock in hindsight.

One challenging benchmark is the *Constant Rebalanced Portfolios* (CRP) strategy, which rebalances to a fixed portfolio  $\mathbf{b}$  every period. In particular, the portfolio strategy can be represented as  $\mathbf{b}_1^n = \{\mathbf{b}, \mathbf{b}, \dots\}$ , in which  $\mathbf{b}$  is a predefined portfolio. Thus, CRP's cumulative portfolio wealth can be calculated as  $S_n(\text{CRP}(\mathbf{b})) = \prod_{t=1}^n \mathbf{b}^\top \mathbf{x}_t$ . One special CRP is the best CRP (BCRP) strategy, which maximizes the cumulative wealth as,

$$S_n(\text{BCRP}) = \max_{\mathbf{b} \in \Delta_m} S_n(\text{CRP}(\mathbf{b})) = S_n(\text{CRP}(\mathbf{b}^*)).$$

Note that BCRP is only a hindsight strategy, which can only be calculated with complete market sequence. Cover and Thomas [16, Theorem 15.3.1] proved that BCRP is an optimal strategy in the i.i.d market and Cover [14] showed its benefits as a target, that is, BCRP is guaranteed to outperform the Best Stock, Value Line Index and Dow Jones Index.

One desired theoretical result for an on-line portfolio selection algorithm is *universality* [14, 50]. An algorithm *Alg* is *universal* if the average (*external*) *regret* [54, 5] over **all** sequences of price relative vectors for  $n$  periods asymptotically approaches zero, that is,

$$\frac{1}{n} \text{regret}_n(\text{Alg}) = \frac{1}{n} (\log S_n(\text{BCRP}) - \log S_n(\text{Alg})) \longrightarrow 0, \text{ as } n \rightarrow \infty.$$

In other words, for an arbitrary sequences of price relatives, a universal algorithm asymptotically approaches the same exponential growth rate as the BCRP strategy.

### 3.2. Follow-the-Winner Approaches

The first approach, *Follow-the-Winner*, is characterized by increasing the relative weights of more successful experts/stocks. Instead of targeting market and the best stock, algorithms in this category often aim to approach the BCRP strategy.

Cover [14] proposed the *Universal Portfolios (UP)* strategy. Formally, its update scheme is historical performance weighted average of all possible constant rebalanced portfolios in the entire simplex domain,

$$\mathbf{b}_1 = \left( \frac{1}{m}, \dots, \frac{1}{m} \right), \quad \mathbf{b}_{t+1} = \frac{\int_{\Delta_m} \mathbf{b} S_t(\mathbf{b}) d\mathbf{b}}{\int_{\Delta_m} S_t(\mathbf{b}) d\mathbf{b}}.$$

In detail, UP's initial portfolio  $\mathbf{b}_1$  is uniform over the market, and its  $t + 1^{st}$  portfolio is historical performance weighted average of all CRP experts with  $\mathbf{b} \in \Delta_m$ . Thus, UP's final cumulative wealth is uniform weighted average of all CRP experts' wealth,  $S_n(UP) = \int_{\Delta_m} S_n(\mathbf{b}) d\mathbf{b}$ . Intuitively, Cover's UP operates similar to a *Fund of Fund (FOF)*, and its main idea is to buy and hold the parameterized CRP strategies over the simplex domain. The regret achieved by Cover's UP is  $O(m \log n)$ , and UP's time complexity is  $O(n^m)$ .

Helmbold et al. [37] proposed the *Exponential Gradient (EG)* strategy, which is based on the same algorithm for mixture estimation [36]. The strategy generally formulated as the following optimization problem,

$$\mathbf{b}_{t+1} = \arg \max_{\mathbf{b} \in \Delta_m} \eta \log \mathbf{b} \cdot \mathbf{x}_t - R(\mathbf{b}, \mathbf{b}_t),$$

where  $R(\mathbf{b}, \mathbf{b}_t)$  denotes a regularization term and  $\eta > 0$  denotes the learning rate. Straightforwardly, the optimization is to track last best stock and keep previous portfolio information via regularization. EG adopts relative entropy as its regularization term, that is,  $R(\mathbf{b}, \mathbf{b}_t) = \sum_{i=1}^m b_i \log \frac{b_i}{b_{t,i}}$ . EG's formulation is convex in  $\mathbf{b}$ , however, it is hard to solve since log is nonlinear. Thus, the authors took the log function's first-order Taylor expansion at  $\mathbf{b}_t$ , which is linear and straightforward to solve. With a stronger assumption of lower bound on the price relatives than Cover's UP, EG has a regret bound of  $O(\sqrt{n} \log m)$  and a running time of  $O(mn)$ . In terms of length of trading periods ( $n$ ), the regret is not as tight as Cover's UP<sup>1</sup>, however, its linear time substantially surpasses UP.

Gaivoronski and Stella [24] proposed *Successive Constant Rebalanced Portfolios (SCRP)* for stationary markets. For each period, SCRP directly adopts the BCRP to date<sup>2</sup>,

$$\mathbf{b}_{t+1} = \arg \max_{\mathbf{b} \in \Delta_m} \sum_{\tau=1}^t \log(\mathbf{b} \cdot \mathbf{x}_\tau).$$

The authors further solved the optimal portfolio  $\mathbf{b}_t^*$  via stochastic optimization [3], resulting in SCRP's portfolio updates [24, Algorithm 1]. Under stronger assumptions [24, Eq. (16) & (17)] than Cover's UP, the regret bound

<sup>1</sup>The EG approach has a better dependence on the number of assets,  $m$ .

<sup>2</sup>Ordentlich [49, Chapter 4.4] briefly mentioned a similar idea.

achieved by SCRP is  $O(m \log n)$ . Gaivoronski and Stella [25] further proposed *Adaptive Portfolio Selection* (APS), which can handle three portfolio selection tasks, including adaptive Markowitz portfolio, log-optimal constant rebalanced portfolio, and index tracking.

Agarwal et al. [1] formulated the *Online Newton Step* (ONS) as

$$\mathbf{b}_{t+1} = \arg \max_{\mathbf{b} \in \Delta_m} \sum_{\tau=1}^t \log(\mathbf{b} \cdot \mathbf{x}_\tau) - \frac{\beta}{2} R(\mathbf{b}),$$

where  $\beta$  denotes a trade-off parameter and  $R(\mathbf{b})$  is a regularization term on  $\mathbf{b}$ . Note that all historical information is exploited by the first term, thus the regularization term only involves next portfolio. It then solves the optimization problem with L2-norm regularization via online convex optimization [57, 35, 33, 34]. Similar to regular offline Newton method, ONS' basic idea is to replace log term via its second-order Taylor expansion at  $\mathbf{b}_t$ , and then solve for the closed form updates. ONS' regret bound is  $O(m \log n)$ , which is the same to Cover's UP. Note that ONS requires stronger assumptions on the price relatives than Cover's UP. Since it iteratively updates the first and second order information, it costs  $O(m^3)$  per period, which is irrelevant to the number of past periods.

### 3.3. Follow-the-Loser Approaches

The underlying assumption for the optimality of BCRP strategy is that asset returns are i.i.d. However, according to the related empirical financial studies [11, 13], real asset returns may not be i.i.d.<sup>3</sup> and they may be systematic or predictable. Although Cover [14] proved that BCRP is able to achieve maximum performance theoretically, BCRP didn't achieve outstanding performance in backtests (c.f., Table 6.), as empirical asset returns violate the i.i.d. assumption. While the Follow-the-Winner strategies asymptotically approach the growth rate of BCRP, they often perform worse than BCRP. On the contrary, strategies in this section, or the Follow-the-Loser approaches, assume that asset returns follow a property of *mean reversion* [6, 52, 47], which indicates that the well (poor)-performing assets will perform poor (well) in the subsequent periods. Rather than tracking the winners, the *Follow-the-Loser* approach is often characterized by transferring the wealth from winners to losers.

Borodin et al. [8] proposed a Follow-the-Loser strategy named *Anti Correlation* (Anticor). Rather than no distributional assumption like Cover's UP, Anticor assumes that market follows the mean reversion principle. To exploit the property, it statistically makes bets on the consistency of positive lagged cross-correlation and negative auto-correlation. Anticor calculates a cross-correlation matrix  $M_{cor}$  between two specific market windows. Following the mean reversion principle, Anticor transfers weights from winning assets to losing assets, and the corresponding amounts are adjusted by the cross-correlation matrix.

Li et al. [46] proposed *Passive Aggressive Mean Reversion* (PAMR) strategy, which exploits the mean reversion property with the Passive Aggressive (PA) online learning [53, 17]. The main idea of PAMR is to design a loss

---

<sup>3</sup>The related test is refer to the test of "Random Walk 1". For the test hypothesis and some existing evidences, please refer to Campbell et al. [11, Chapter 2]. There also exist a set of stylized facts or patterns of asset returns, please refer to Cont [13].

function of  $\ell_\epsilon(\mathbf{b}; \mathbf{x}_t)$  in order to reflect the mean reversion property, that is, if the expected return based on last price relative is larger than a pre-defined threshold, the loss will linearly increase; otherwise, the loss is zero. Based on the loss function, PAMR passively maintains last portfolio if the loss is zero, otherwise it aggressively approaches a new portfolio that can force the loss to zero. In summary, PAMR obtains the next portfolio via the following optimization,

$$\mathbf{b}_{t+1} = \arg \min_{\mathbf{b} \in \Delta_m} \frac{1}{2} \|\mathbf{b} - \mathbf{b}_t\|^2 \quad \text{s.t.} \quad \ell_\epsilon(\mathbf{b}; \mathbf{x}_t) = 0.$$

Solving the optimization problem, PAMR has a clean closed form update scheme,

$$\mathbf{b}_{t+1} = \mathbf{b}_t - \tau_t (\mathbf{x}_t - \bar{x}_t \mathbf{1}), \quad \tau_t = \max \left\{ 0, \frac{\mathbf{b}_t \cdot \mathbf{x}_t - \epsilon}{\|\mathbf{x}_t - \bar{x}_t \mathbf{1}\|^2} \right\}.$$

Since the authors ignored the non-negativity constraint of the portfolio in the derivation, they also added a simplex projection step [22]. The closed form update scheme clearly reflects mean reversion by transferring the wealth from poor performing assets to well performing assets.

Li et al. [45] proposed *Confidence Weighted Mean Reversion* (CWMR) algorithm to further exploit the second order portfolio information and the mean reversion trading idea via Confidence Weighted (CW) online learning [20, 19, 18, 21]. The basic idea of CWMR is to model the **portfolio vector** (not the price relative vector) as a multivariate Gaussian distribution with mean  $\mu \in \mathbb{R}^m$  and the diagonal covariance matrix  $\Sigma \in \mathbb{R}^{m \times m}$ , which has nonzero diagonal elements  $\sigma^2$  and zero for off-diagonal elements. While the mean represents the knowledge for the portfolio, the diagonal covariance matrix term stands for the confidence we have in the corresponding portfolio mean. Then CWMR sequentially updates the mean and covariance matrix of the Gaussian distribution. In particular, CWMR keeps the next distribution close to the last distribution in terms of Kullback-Leibler divergence if the probability that a portfolio return is lower than  $\epsilon$  is higher than a pre-defined threshold. In summary, CWMR formulates the optimization as follows:

$$(\mu_{t+1}, \Sigma_{t+1}) = \arg \min_{\mu \in \Delta_m, \Sigma} D_{\text{KL}}(\mathcal{N}(\mu, \Sigma) \parallel \mathcal{N}(\mu_t, \Sigma_t)) \quad \text{s.t.} \quad \Pr[\mu \cdot \mathbf{x}_t \leq \epsilon] \geq \theta.$$

The above optimization can be solved by turning it into convex optimization and deriving closed-form updates.

Similar to Anticor, both PAMR and CWMR's mean reversion nature makes it hard to obtain a meaningful theoretical regret bound. Empirical performance show that the two algorithms can outperform the state-of-the-art, including Anticor, which passively reverts to the mean. However, both PAMR and CWMR suffer from drawbacks in risk management since it suffers significant performance degradation if the underlying single period mean reversion fails to exist. Such drawback is clearly indicated by its performance in the DJIA dataset [7, 8, 46], and motivates the proposed new algorithms in this article. Finally, we note that an early short version of this work has been published in the conference proceedings of ICML [42], and a recent robust variant in this series was also published in IJCAI [39].

### 3.4. Pattern-Matching based Approaches

Besides the two categories of Follow-the-Winner/Loser, another type of strategies may utilize both winners and losers, which is based on pattern matching. Based on nonparametric prediction [29], this category consists of several

pattern-matching based investment strategies [27, 31, 30, 44, 28].

The Pattern-Matching based approaches [27] consists of two steps: (i) *sample selection* and (ii) *portfolio optimization*. The first sample selection step selects a subset  $C$  of trading days, which have similar historical price relatives and will be used to predict the next price relatives. After locating the similarity set, each sample price relative  $\mathbf{x}_i, i \in C$  is assigned with a probability  $P_i, i \in C$ . The portfolio optimization step learns an optimal portfolio based on the set:

$$\mathbf{b}_{t+1} = \arg \max_{\mathbf{b} \in \Delta_m} U(\mathbf{b}, C),$$

where  $U(\cdot)$  is a specified utility function. One particular utility function is the log utility, which is always the default function. In case of empty similarity set, a uniform portfolio is adopted.

### 3.4.1. Sample Selection Techniques

The general idea is to select similar samples from historical price relatives by comparing the preceding market windows of two price relatives. Let us assume that we are locating the price relatives that are similar to next price relative  $\mathbf{x}_{t+1}$ . The basic routine is to iterate all historic price relatives  $\mathbf{x}_i, i = w + 1, \dots, t$  and count  $\mathbf{x}_i$  as one similar price relative, if the preceding market window  $\mathbf{x}_{i-w}^{i-1}$  is similar to the latest market window  $\mathbf{x}_{t-w+1}^t$ . A set  $C$  is maintained to contain the indexes of similar price relatives. Note that market window is a  $w \times m$ -matrix and the similarity between two market windows is often calculated on the concatenated  $w \times m$ -vector.

Nonparametric *histogram-based* sample selection [29] pre-defines a set of discretized partitions, and partitions both market windows, and chooses the price relatives whose  $\mathbf{x}_{i-1}^{i-w}$  is in the same partition as  $\mathbf{x}_{t-w+1}^t$ . In particular, given a partition  $P = A_j, j = 1, 2, \dots, d$  of  $\mathbb{R}_+^m$  into  $d$  disjoint sets and a corresponding discretization function  $G(\mathbf{x}) = j$ , we can define the similarity set as,

$$C_H(\mathbf{x}_1^t, w) = \{w < i < t + 1 : G(\mathbf{x}_{t-w+1}^t) = G(\mathbf{x}_{i-w}^{i-1})\}.$$

Nonparametric *kernel-based* sample selection [27] identifies the similarity set by comparing two market windows via Euclidean distance, that is,

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

where  $c$  and  $\ell$  are the thresholds used to control the number of similar samples. Note that the authors adopted two threshold parameters for theoretical analysis. Nonparametric *nearest neighbor-based* sample selection [30] searches the price relatives whose preceding market windows are within the  $\ell$  nearest neighbor of the latest market window in terms of Euclidean distance, that is,

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

where  $\ell$  is a threshold parameter. *Correlation-driven* nonparametric sample selection [44] identifies the linear similarity among two market windows via correlation coefficient, that is,

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



Table 1: Pattern-Matching based approaches: sample selection and portfolio optimization.

	Sample Selection Techniques			
Portfolio Optimization	Histogram	Kernel	Nearest Neighbor	Correlation
Log-optimal	$B^H: C_H + U_L$	$B^K: C_K + U_L$	$B^{NN}: C_N + U_L$	$CORN: C_C + U_L$
Semi log-optimal	—	$B^S: C_K + U_S$	—	—
Markowitz-type	—	$B^M: C_K + U_M$	—	—
GV-type	—	$B^{GV}: C_K + U_R$	—	—

where  $\rho$  is a pre-defined correlation coefficient threshold.

### 3.4.2. Portfolio Optimization Techniques

The second step of the Pattern-Matching based Approaches is to construct an optimal portfolio based on the similar set  $C$ . Two main approaches are the Kelly’s capital growth portfolio and Markowitz’s mean variance portfolio.

Györfi et al. [27] proposed to figure out a *log-optimal* (Kelly) portfolio based on similar price relatives, which clearly follows the Capital Growth Theory. Given a similarity set, the log-optimal utility function is defined as,

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

where  $P_i$  denotes the probability assigned to the similar price relative  $\mathbf{x}_i, i \in C(\mathbf{x}_1^t)$ . Györfi et al. [27] assumes a uniform probability among the similar samples, thus it is equivalent to the following utility function,

$$U_L(\mathbf{b}, C(\mathbf{x}_1^t)) = \sum_{i \in C(\mathbf{x}_1^t)} \log \mathbf{b} \cdot \mathbf{x}_i.$$

Besides the *log-optimal* portfolio optimization, other methods can also be adopted, such as *semi-log-optimal* portfolio [56, 31], *Markowitz-type* portfolio [51], and *GV-type* portfolio [32], which we do not detail here.

In any above procedure, if the similarity set is empty, we can choose either uniform portfolio or previous portfolio.

### 3.4.3. Combinations

In this section, we combine the first and second step and describe the algorithms using Pattern-Matching based approaches. Table 1 shows the existing combinations, where “—” indicates there is no algorithm for such combination.

One default utility function is the log-optimal function or the BCRP portfolio. Györfi and Schäfer [29] introduced the *nonparametric histogram-based log-optimal* investment strategy ( $B^H$ ), which combines the histogram-based sample selection and log-optimal utility function and proved its universality. Györfi et al. [27] presented *nonparametric kernel-based log-optimal* investment strategy ( $B^K$ ), which combines the kernel-based sample selection and log-optimal utility function and proved its universality. Györfi et al. [30] proposed *nonparametric nearest neighbor log-optimal* investment strategy ( $B^{NN}$ ), which combines the nearest neighbor sample selection and log-optimal

utility function and proved its universality. Li et al. [44] created *correlation-driven nonparametric* learning approach (CORN) by combining the correlation driven sample selection and log-optimal utility function and showed its superior empirical performance over previous three combinations. Besides the log-optimal utility function, several algorithms using different utility functions have been proposed. Györfi et al. [31] proposed *nonparametric kernel-based semi-log-optimal* investment strategy ( $B^S$ ) by combining the kernel-based sample selection and semi-log-optimal utility function to ease the computation of ( $B^K$ ). Ottucsák and Vajda [51] proposed *nonparametric kernel-based Markowitz-type* investment strategy ( $B^M$ ) by combining the kernel-based sample selection and Markowitz-type utility function to make trade-offs between the return (mean) and risk (variance) of expected portfolio return. Györfi and Vajda [32] proposed *nonparametric kernel-based GV-type* investment strategy ( $B^{GV}$ ) by combining the kernel-based sample selection and GV-type utility function to construct portfolios in case of transaction costs.

### 3.5. Analysis of Existing Work

Table 2: Summary of the existing optimization formulations and their underlying predictions.  $R(\cdot)$  denotes the regularization term, such as  $L_2$  norm.  $\text{Prob}(\cdot)$  denotes a probability function. PAMR/CWMMR's prediction is not a strict equivalence, which we do not prove. PAMR & CWMMR are formulated with **argmin**, reflecting the mean reversion idea.

Categories	Methods	Formulations ( $\mathbf{b}_{t+1}$ )	Pred. ( $\tilde{\mathbf{x}}_{t+1}^i$ )	Prob. ( $p_i$ )
In hindsight	BCRP	$\arg \max_{\mathbf{b} \in \Delta_m} \sum_{i=1}^n \frac{1}{n} \log \mathbf{b} \cdot \mathbf{x}_i$	$\mathbf{x}_i, i = 1, \dots, n$	$1/n$
1	SCRP	$\arg \max_{\mathbf{b} \in \Delta_m} \sum_{i=1}^t \frac{1}{t} \log \mathbf{b} \cdot \mathbf{x}_i$	$\mathbf{x}_i, i = 1, \dots, t$	$1/t$
	ONS	$\arg \max_{\mathbf{b} \in \Delta_m} \sum_{i=1}^t \frac{1}{t} \log \mathbf{b} \cdot \mathbf{x}_i - \lambda R(\mathbf{b})$	$\mathbf{x}_i, i = 1, \dots, t$	$1/t$
2	$B^K/B^{NN}/\text{CORN}$	$\arg \max_{\mathbf{b} \in \Delta_m} \sum_{i \in C_t} \frac{1}{ C_t } \log \mathbf{b} \cdot \mathbf{x}_i$	$\mathbf{x}_i, i \in C_t$	$1/ C_t $
3	EG	$\arg \max_{\mathbf{b} \in \Delta_m} \log \mathbf{b} \cdot \mathbf{x}_t - \lambda R(\mathbf{b}, \mathbf{b}_t)$	$\mathbf{x}_t$	1.00
	PAMR	$\arg \min_{\mathbf{b} \in \Delta_m} \mathbf{b} \cdot \mathbf{x}_t + \lambda R(\mathbf{b}, \mathbf{b}_t)$	$1/\mathbf{x}_t$	1.00
	CWMMR	$\arg \min_{\mathbf{b} \in \Delta_m} \text{Prob}(\mathbf{b} \cdot \mathbf{x}_t) + \lambda R(\mathbf{b}, \mathbf{b}_t)$	$1/\mathbf{x}_t$	1.00

Most existing on-line portfolio selection research grounds on the principle of Kelly investment [41, 10, 55, 23], that is, to maximize the expected *log* return of a portfolio. In particular, a portfolio manager first predicts  $\tilde{\mathbf{x}}_{t+1}$  in terms of  $k$  possible outcomes  $\tilde{\mathbf{x}}_{t+1}^1, \dots, \tilde{\mathbf{x}}_{t+1}^k$  associated with some distribution  $(p_1, \dots, p_k)$ . Note that each  $\tilde{\mathbf{x}}_{t+1}^i$  denotes one possible outcome, or a vector of individual price relative predictions. Then, the optimal portfolio is generated by maximizing the expected log return:  $\mathbf{b}_{t+1} = \arg \max_{\mathbf{b} \in \Delta_m} \sum_{i=1}^k p_i \log(\mathbf{b} \cdot \tilde{\mathbf{x}}_{t+1}^i)$ .

Based on different methods of predicting  $\tilde{\mathbf{x}}_{t+1}^i$  and  $p_i$ , most existing algorithms can be classified into three categories, excluding the BCRP strategy in hindsight. Table 2 summarizes their formulations and underlying predictions. Note that we have done some transformations but maintained their key ideas. We introduce them according to their empirical performance.

The first category, which consists of SCRCP and ONS, assumes the predictions consist of all historical price relatives

with uniform distribution<sup>4</sup>. That is, it assumes that on period  $t + 1$ , price relative vector may be  $\mathbf{x}_i$  with a probability of  $\frac{1}{t}$ ,  $i = 1, \dots, t$ . In other words, this category aims to model the next price relatives as the average of all historical price relatives. The algorithms in this category all have theoretical regret bound and are universal. However, their empirical results show that such assumption may not be precise enough to model the market.

The second category mainly consists of the pattern matching-based algorithms. It models next price relatives as the average of a sampled set of similar price relatives. Denoting the similar index set as  $C_t$ , it models the next price relative as  $\mathbf{x}_i, i \in C_t$  with a uniform probability of  $\frac{1}{|C_t|}$ . The algorithms in this category (except CORN) have been proved universal consistency, and their empirical results show that such assumption can fit the market well.

The last category, including EG, PAMR and CWMR, assumes a single prediction with 100% probability, and considers their former histories by regularizing the previous  $\mathbf{b}_{t-1}$ . In particular, EG assumes  $\tilde{\mathbf{x}}_{t+1}^1 = \mathbf{x}_t$  with  $p_1 = 100\%$ <sup>5</sup>, while PAMR and CWMR assume  $\tilde{\mathbf{x}}_{t+1}^1 = \frac{1}{\mathbf{x}_t}$  with  $p_1 = 100\%$ , which is in essence the mean reversion idea. Note that the formulations of PAMR and CWMR ignore log utility due to their single-value prediction and the consideration of convexity and computational issues<sup>6</sup>. Though all three algorithms assume that all information is fully reflected by  $\mathbf{x}_t$ , their performance diverges and supports that mean reversion may better fit the markets. On the one hand, even with a decent theoretical result, EG always performs far behind. On the other hand, though without theoretical guarantees, PAMR and CWMR always produce the best results in various real markets. However, PAMR and CWMR suffer from dramatic failures when such single-period mean reversion is not satisfied [46], which motivates our study.

## 4. On-Line Moving Average Reversion

### 4.1. Motivation

Empirical results [46, 45] show that mean reversion, which assumes poor stock will perform well in the subsequent periods, may better fit the markets. PAMR and CWMR can exploit the mean reversion property well and achieve excellent results on most datasets at the time, especially the NYSE benchmark dataset [14]. However, they rely on a simple assumption that the predicted price relative  $\tilde{\mathbf{x}}_{t+1}$  will be inversely proportional to last price relative  $\mathbf{x}_t$ . In particular, they implicitly assume that next price  $\tilde{\mathbf{p}}_{t+1}$  will revert to last price  $\mathbf{p}_{t-1}$  as follows,

$$\tilde{\mathbf{x}}_{t+1} = \frac{1}{\mathbf{x}_t} \implies \frac{\tilde{\mathbf{p}}_{t+1}}{\mathbf{p}_t} = \frac{\mathbf{p}_{t-1}}{\mathbf{p}_t} \implies \tilde{\mathbf{p}}_{t+1} = \mathbf{p}_{t-1}.$$

<sup>4</sup>This characterization is in the view of the authors. In the online learning community in general, “risk empirical risk minimization” is a good strategy for the “online convex optimization” problems. Such characterization also occurs on the below categories.

<sup>5</sup>But this does not mean that EG is forgetful. The former history is built into the vector of  $\mathbf{b}_{t-1}$  and the algorithm is designed to regularize from the previous solution.

<sup>6</sup>PAMR&CWMR’s formulations with log utility function make their optimization formulations non-convex and hard to solve. As done by Helmbold et al. [37], such formulation is often solved with linear approximation, which thus becomes the version without log utility. That is, as  $\log(\mathbf{b} \cdot \mathbf{x}_t) \approx \log(\mathbf{b}_t \cdot \mathbf{x}_t) + \frac{\mathbf{x}_t}{\mathbf{b}_t \cdot \mathbf{x}_t} \cdot (\mathbf{b} - \mathbf{b}_t)$ , minimizing  $\log(\mathbf{b} \cdot \mathbf{x}_t)$  is often solved by minimizing  $\mathbf{b} \cdot \mathbf{x}_t$ , which is convex and easy to solve. Please find the details in their respective studies. For the same reason, this article also ignores the log utility.

Though empirically effective on most datasets, PAMR and CWMR’s *single-period* assumption causes two potential problems. Firstly, both algorithms suffer from the frequently fluctuating raw prices, as they often contain a lot of noises. Second, their assumption of single-period mean reversion may not be satisfied in the real world. Even two consecutively declining price relatives, which are common, can fail both algorithms. One real example [46] is DJIA dataset [8], on which PAMR performs the worst among the state of the art. Thus, traders are more likely to predict prices using certain long-term values. Also on the DJIA dataset, Anticor, which exploits the *multi-period* statistical correlation, performs much better. However, as illustrated in Li et al. [46, 45], due to its heuristic nature, Anticor can not fully exploit the mean reversion property. The two problems caused by the single-period assumption and Anticor’s inability to fully exploit mean reversion call for a more powerful approach to effectively exploit mean reversion, especially in terms of multi-period.

Table 3: Illustration of growth of mean reversion strategies on toy markets. OLMAR adopts a simple moving average with window size  $k$ , and a threshold of 2.

Market Sequences	BCRP	PAMR	OLMAR
A : $(1, 2), (1, \frac{1}{2}), (1, 2), (1, \frac{1}{2}), \dots$	$(\frac{9}{8})^{n/2}$	$\frac{3}{2} \times 2^{\frac{n-1}{2}}$	$\frac{9}{8}$
B : $(1, 2), (1, 2), (1, \frac{1}{2}), (1, \frac{1}{2}), (1, 2), \dots$	$(\frac{9}{8})^{n/2}$	$\frac{3}{2}$	$\frac{9}{16} \times 2^{\frac{n-4}{2}}$
C : $(1, 2), (1, 2), (1, 2), (1, \frac{1}{2}), (1, \frac{1}{2}), (1, \frac{1}{2}), (1, 2), \dots$	$(\frac{9}{8})^{n/2}$	$\frac{3}{2} \times (\frac{1}{2})^{\frac{n-1}{6}}$	$\frac{9}{8} \times 2^{\frac{n-5}{6}}$
D : $\underbrace{(1, 2), \dots, (1, 2)}_{k=4}, \underbrace{(1, 1/2), \dots, (1, 1/2)}_{k=4}, (1, 2), \dots$	$(\frac{9}{8})^{n/2}$	$\frac{3}{2} \times (\frac{1}{2})^{\frac{n-1}{4}}$	$\frac{9}{4} \times 2^{\frac{n-6}{8}}$
E : $\underbrace{(1, 2), \dots, (1, 2)}_{k=5}, \underbrace{(1, 1/2), \dots, (1, 1/2)}_{k=5}, (1, 2), \dots$	$(\frac{9}{8})^{n/2}$	$\frac{3}{2} \times (\frac{1}{2})^{(n-1) \times \frac{3}{10}}$	$\frac{9}{2} \times 2^{\frac{n-7}{10}}$

Now let us see a classic example [15, 46] to illustrate the drawbacks of single-period mean reversion, as shown in Table 3. The toy market consists of cash and one volatile stock, which follows market sequence A. It is easy to prove that BCRP ( $\mathbf{b} = (\frac{1}{2}, \frac{1}{2})$ ) grows by a factor of  $(\frac{9}{8})^{n/2}$ , while PAMR & CWMR can grow by a better factor of  $\frac{3}{2} \times 2^{(n-1)/2}$ . Note that this virtual sequence essentially follows single-period mean reversion, which perfectly fits with PAMR and CWMR’s assumption. However, if market sequence does not satisfy such assumption, both PAMR and CWMR would fail badly. Let us extend the market sequence to a two-period reversion, that is, market sequence B. In such a market, BCRP can achieve the same growth as before. Contrarily, PAMR/CWMR can achieve a constant wealth of  $\frac{3}{2}$ , which means no growth! This shows that PAMR and CWMR actually will fail on such markets. More generally, if we further extend the market sequence to  $k$ -period mean reversion, BCRP can still achieve the same growth, while PAMR & CWMR will grow to  $\frac{3}{2} \times (\frac{1}{2})^{(n-1) \times (\frac{1}{2} - \frac{1}{k})}$ , which will definitely approach bankruptcy when  $k \geq 3$ .

To better exploit the (multi-period) mean reversion property, we proposed a new type of algorithm, named *On-Line Moving Average Reversion* (OLMAR), for on-line portfolio selection. The essential idea is to exploit multi-period moving average (mean) reversion via powerful online machine learning. Rather than  $\tilde{\mathbf{p}}_{t+1} = \mathbf{p}_{t-1}$ , OLMAR assumes

that next price will revert to a *Moving Average* (MA), that is,  $\tilde{\mathbf{p}}_{t+1} = MA_t$ , where  $MA_t$  denotes the *Moving Average* (MA) at the end of  $t^{th}$  period. In time series analysis, MA is typically used to smooth short-term price fluctuations and focuses on long-term trends. Thus it can solve the two drawbacks of existing mean reversion algorithms.

Without detailing the calculation <sup>7</sup>, we list the growth of OLMAR in different toy markets in Table 3. Clearly, OLMAR performs much better than PAMR/CWMR in the scenarios of multi-period mean reversion, while performs poor in single-period reversion. Our further empirical analysis in Section 5 shows that the markets are more likely to follow multi-period reversion.

#### 4.2. Formulations

In this article, we adopt two types of moving average. The first, so-called ‘‘Simple Moving Average’’ (SMA), truncates the historical prices via a window and calculates its arithmetical average,

$$SMA_t(w) = \frac{1}{w} \sum_{i=t-w+1}^t \mathbf{p}_i,$$

where  $w$  denotes the window size. Although we can enlarge the window size such that SMA can cover more historical price relatives, the empirical evaluations in Section 5 show that generally their performance drops as the window size increases.

To consider entire price relatives rather than a window, the second type, ‘‘Exponential Moving Average’’ (EMA), adopts all historical prices, and each price is exponentially weighted,

$$\begin{aligned} EMA_t(\alpha) &= \alpha \mathbf{p}_t + (1 - \alpha) EMA_{t-1}(\alpha) \\ &= \alpha \mathbf{p}_t + (1 - \alpha) \alpha \mathbf{p}_{t-1} + (1 - \alpha)^2 \alpha \mathbf{p}_{t-2} + \cdots + (1 - \alpha)^{t-1} \mathbf{p}_1, \end{aligned}$$

where  $\alpha \in (0, 1)$  denotes a decaying factor.

To this end, we can calculate the predicted price relative vector following the idea of so-called ‘‘Moving Average Reversion’’ (MAR). Based on the two types of moving average, we can infer two types of MAR,

##### **Moving Average Reversion: MAR-1**

$$\begin{aligned} \tilde{\mathbf{x}}_{t+1}(w) &= \frac{SMA_t(w)}{\mathbf{p}_t} = \frac{1}{w} \left( \frac{\mathbf{p}_t}{\mathbf{p}_t} + \frac{\mathbf{p}_{t-1}}{\mathbf{p}_t} + \cdots + \frac{\mathbf{p}_{t-w+1}}{\mathbf{p}_t} \right) \\ &= \frac{1}{w} \left( \mathbf{1} + \frac{1}{\mathbf{x}_t} + \cdots + \frac{1}{\otimes_{i=0}^{w-2} \mathbf{x}_{t-i}} \right), \end{aligned} \tag{1}$$

where  $w$  is the window size and  $\otimes$  denotes element-wise product.

##### **Moving Average Reversion: MAR-2**

$$\begin{aligned} \tilde{\mathbf{x}}_{t+1}(\alpha) &= \frac{EMA_t(\alpha)}{\mathbf{p}_t} = \frac{\alpha \mathbf{p}_t + (1 - \alpha) EMA_{t-1}(\alpha)}{\mathbf{p}_t} \\ &= \alpha \mathbf{1} + (1 - \alpha) \frac{EMA_{t-1}(\alpha)}{\mathbf{p}_{t-1}} \frac{\mathbf{p}_{t-1}}{\mathbf{p}_t} \\ &= \alpha \mathbf{1} + (1 - \alpha) \frac{\tilde{\mathbf{x}}_t}{\mathbf{x}_t}, \end{aligned} \tag{2}$$

<sup>7</sup>We calculate OLMAR’s growth using Algorithm 3, which will finally stabilize.

where  $\alpha \in (0, 1)$  denotes the decaying factor and the operations are all element-wise.

Based on the expected price relative vector in Eq. (1) and Eq. (2), OLMAR further adopts the idea of an effective online learning algorithm, that is, Passive Aggressive (PA) [17] learning, to exploit the moving average reversion. Generally proposed for classification, PA passively keeps the previous solution if the classification is correct, while aggressively approaches a new solution if the classification is incorrect. After formulating the proposed OLMAR optimization, we solve its closed form update and design the proposed algorithm.

The proposed formulation, OLMAR, is to exploit moving average reversion via online learning techniques [38]. The basic idea is to maximize the expected return  $\mathbf{b} \cdot \tilde{\mathbf{x}}_{t+1}$ , and keep last portfolio information via a regularization term, using the online passive-aggressive learning method [17]. Thus, we follow the similar idea of PAMR [46] and formulate an optimization as,

**Optimization Problem:** OLMAR

$$\mathbf{b}_{t+1} = \arg \min_{\mathbf{b} \in \Delta_m} \frac{1}{2} \|\mathbf{b} - \mathbf{b}_t\|^2 \quad \text{s. t.} \quad \mathbf{b} \cdot \tilde{\mathbf{x}}_{t+1} \geq \epsilon.$$

Note that we adopt expected return instead of expected log return<sup>8</sup>. According to Helmbold et al. [37], to solve the optimization with expected log return, one can adopt the first-order Taylor expansion, which is essentially linear.

The above formulation explicitly reflects the basic idea of the proposed OLMAR. On the one hand, if its constraint is satisfied, that is, the expected return is higher than a threshold, then the resulting portfolio equals to previous portfolio. On the other hand, if the constraint is not satisfied, then the formulation will figure out a new portfolio such that the expected return is higher than the threshold, while the new portfolio is not far from previous one.

Note that although OLMAR's formulation looks similar to that of PAMR, they have some essential differences. In particular, PAMR's key constraint is  $\mathbf{b} \cdot \mathbf{x}_t \leq \epsilon$ , while OLMAR's key constraint is  $\mathbf{b} \cdot \tilde{\mathbf{x}}_t \geq \epsilon$ . First, PAMR uses raw price relative ( $\mathbf{x}_t$ ), while OLMAR adopts the price relative calculated by following the moving average reversion principle ( $\tilde{\mathbf{x}}_t$ ). Second, their inequality signs (" $\leq$ " for PAMR, and " $\geq$ " for OLMAR) are different. The different signs will make their behaviors different, which distinguish the two algorithms. After making certain transformations, PAMR may be written in a similar form, as shown in Table 2. However, their prediction functions are different, that is, OLMAR adopts multi-period mean reversion, while PAMR exploits single-period mean reversion.

### 4.3. Algorithm and Analysis

The above formulations are convex and thus can be solved by convex optimization techniques [9]. We now derive the OLMAR solutions in closed form and show its solution in Proposition 1.

**Proposition 1.** *The solution of OLMAR without considering the non-negativity constraint is*

$$\mathbf{b}_{t+1} = \mathbf{b}_t + \lambda_{t+1} (\tilde{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1}),$$

---

<sup>8</sup>With log function, the empirical performance does not make too much difference.

---

**Algorithm 2** Portfolio Selection with OLMAR.

---

1: **Input:**  $\epsilon > 1$ : Reversion threshold;  $w \geq 2$ : Window size;  $\alpha \in (0, 1)$ : Decaying factor;  $\mathbf{x}_1^n$ : Market sequence;

2: **Output:**  $S_n$ : Cumulative wealth after  $n^{th}$  periods

3: **Procedure:**

4: Initialization:  $\mathbf{b}_1 = \frac{1}{m} \mathbf{1}$ ,  $S_0 = 1$ ;

5: **for**  $t = 1, 2, \dots, n$  **do**

6:   Receive stock price relatives:  $\mathbf{x}_t$

7:   Calculate daily return and cumulative return:  $S_t = S_{t-1} \times (\mathbf{b}_t \cdot \mathbf{x}_t)$

8:   Predict next price relative vector:

$$\tilde{\mathbf{x}}_{t+1} = \begin{cases} \frac{1}{w} \left( \mathbf{1} + \frac{1}{\mathbf{x}_t} + \dots + \frac{1}{\otimes_{i=0}^{w-2} \mathbf{x}_{t-i}} \right) & \text{MAR-1} \\ \alpha + (1 - \alpha) \frac{\tilde{\mathbf{x}}_t}{\mathbf{x}_t} & \text{MAR-2} \end{cases}$$

9:   Update the portfolio:

$$\mathbf{b}_{t+1} = \text{OLMAR}(\epsilon, \tilde{\mathbf{x}}_{t+1}, \mathbf{b}_t)$$

10: **end for**

---

where  $\bar{x}_{t+1} = \frac{1}{m} (\mathbf{1} \cdot \tilde{\mathbf{x}}_{t+1})$  denotes the average predicted price relative and  $\lambda_{t+1}$  is the Lagrangian multiplier calculated as,

$$\lambda_{t+1} = \max \left\{ 0, \frac{\epsilon - \mathbf{b}_t \cdot \tilde{\mathbf{x}}_{t+1}}{\|\tilde{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1}\|^2} \right\}.$$

Note that in the derivations we do not consider the non-negativity constraint following Helmbold et al. [37]. Thus, it is possible that the resulting portfolio goes out the portfolio simplex domain. To maintain a proper portfolio, we finally project the portfolio calculated according to the Propositions to the simplex domain.

To this end, we can design the proposed algorithms based on the propositions. The on-line portfolio selection procedure is illustrated in Algorithm 2, and the proposed OLMAR procedures are illustrated in Algorithm 3.

In the above algorithms, there are several important parameters, including window size  $w$ , reversion threshold  $\epsilon$ , and decaying factor  $\alpha$ . Our empirical observations of the parameter sensitivity evaluation in Section 5.6 show that the final performance could be sensitive to the parameter  $w$ . To alleviate the parameter sensitivity issue, we propose to explore the online expert strategy for combining the OLMAR experts using different parameters in order to smooth the final performance of the trading system. In particular, we propose to adopt the Buy and Hold (BAH) combination strategy [8, 27, 46], that is, for each period, we treat an individual OLMAR model with a specified parameter  $w \geq 3$  as an expert and combine multiple experts' portfolios weighted by their historical trading performance. We denote the expert algorithm as  $\text{BAH}_W(\text{OLMAR})$  with a parameter  $W$  denoting the maximum window size, that is,  $\text{BAH}_W(\text{OLMAR})$  combines  $W - 2$  individual OLMAR experts with  $w = 3, \dots, W$ .

*Remark on the OLMAR optimization.* We note that it is generally difficult to directly solve the OLMAR optimiza-

---

**Algorithm 3** OLMAR( $\epsilon, C, \tilde{\mathbf{x}}_{t+1}, \mathbf{b}_t$ ).

---

1: **Input:**  $\epsilon > 1$ : Reversion threshold;  $\tilde{\mathbf{x}}_{t+1}$ : Predicted price relatives;  $\mathbf{b}_t$ : Current portfolio;

2: **Output:**  $\mathbf{b}_{t+1}$ : Next portfolio;

3: **Procedure:**

4: Calculate the predicted market return:

$$\bar{x}_{t+1} = \frac{\mathbf{1}^\top \tilde{\mathbf{x}}_{t+1}}{m}$$

5: Calculate the Lagrangian multipliers:

$$\lambda_{t+1} = \max \left\{ 0, \frac{\epsilon - \mathbf{b}_t \cdot \tilde{\mathbf{x}}_{t+1}}{\|\tilde{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1}\|^2} \right\}$$

6: Update the portfolio:

$$\mathbf{b}_{t+1} = \mathbf{b}_t + \lambda_{t+1} (\tilde{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1})$$

7: Normalize the portfolio:

$$\mathbf{b}_{t+1} = \arg \min_{\mathbf{b} \in \Delta_m} \|\mathbf{b} - \mathbf{b}_{t+1}\|^2$$


---

tion, as there is no closed form solution. To solve it efficiently, we decompose the optimization task into two-step optimizations:

$$\mathbf{b}_{t+\frac{1}{2}} = \arg \min \frac{1}{2} \|\mathbf{b} - \mathbf{b}_t\|^2 \quad \text{s. t.} \quad \mathbf{b} \cdot \tilde{\mathbf{x}}_{t+1} \geq \epsilon, \mathbf{b} \cdot \mathbf{1} = 1 \quad (3)$$

$$\mathbf{b}_{t+1} = \arg \min_{\mathbf{b} \in \Delta_m} \left\| \mathbf{b} - \mathbf{b}_{t+\frac{1}{2}} \right\|^2 \quad (4)$$

Helmbold et al. [37] also used a similar approach to solve the optimization of EG. Following Helmbold et al. [37], Eq. (3) ignores portfolio's non-negativity constraint ( $\mathbf{b} \succeq 0$ ), and thus allows shorting, which is common in the portfolio studies [26]. Proposition 1 gives the solution to this optimization. Moreover, Eq. (4) projects the solution into the simplex domain to satisfy the constraint in Section 2.

*Remark on the Update Strategy.* The update of OLMAR is straightforward, that is,

$$\mathbf{b}_{t+1} = \mathbf{b}_t + \lambda_{t+1} (\tilde{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1}).$$

Basically, the above update divides the assets into two subgroups based on the prediction average  $\bar{x}_{t+1}$ . For the assets with predicted values higher than the average, OLMAR increases their proportions; for the others, OLMAR decreases their proportions. The transferred proportions are related to the surprise of the predictions over their average and the positive Lagrangian multiplier,  $\lambda$ . This is consistent with normal portfolio selection procedure, that is, to transfer the wealth to the assets with better prospects.

*Remark on the Time and Space Complexity.* The time complexity of the proposed OLMAR algorithm is  $\mathcal{O}(mn)$ , which is linear with respect to the number of assets  $m$  and the total number of trading periods  $n$ . The space complexity is also linear, namely,  $\mathcal{O}(m)$ . Note that the normalization step can be done efficiently by adopting some fast



implementation that runs in linear time [22]. Overall, the entire algorithm is easy to implement and efficient to run. To the best of our knowledge, OLMAR’s linear running time is no worse than any existing algorithms.

## 5. Experiments

In this section, we will present an extensive set of empirical studies, including our experimental testbed, experimental protocols, comparison schemes, experimental results and detailed empirical analysis. All the source code and datasets for this work can be found in our website: <http://OLPS.stevenhoi.org/OLMAR/>.

### 5.1. Experimental Testbed on Real Data

In our study, we focus on historical daily prices in stock markets<sup>9</sup>, which are easy to obtain from public domains (such as Yahoo Finance and Google Finance<sup>10</sup>), and thus publicly available to other researchers. Data from other types of markets, such as high frequency intra-day quotes and Forex markets, are either too expensive or hard to obtain and process, and thus may reduce the experimental reproducibility. Summarized in Table 4, we employ six real and diverse datasets from several financial markets<sup>11</sup>.

Dataset	Market	Region	Time Frame	# Periods	# Assets
NYSE (O)	Stock	US	Jul. 3 <sup>rd</sup> 1962 - Dec. 31 <sup>st</sup> 1984	5651	36
NYSE (N)	Stock	US	Jan. 1 <sup>st</sup> 1985 - Jun. 30 <sup>th</sup> 2010	6431	23
TSE	Stock	CA	Jan. 4 <sup>th</sup> 1994 - Dec. 31 <sup>st</sup> 1998	1259	88
SP500	Stock	US	Jan. 2 <sup>nd</sup> 1998 - Jan. 31 <sup>st</sup> 2003	1276	25
MSCI	Index	Global	Apr. 1 <sup>st</sup> 2006 - Mar. 31 <sup>st</sup> 2010	1043	24
DJIA	Stock	US	Jan. 14 <sup>th</sup> 2001 - Jan. 14 <sup>th</sup> 2003	507	30

Table 4: Summary of the six datasets from real markets.

The first dataset, “NYSE (O)”, is one “standard” dataset pioneered by Cover [14] and followed by others [37, 8, 1, 27, 30]. This dataset contains 5651 daily price relatives of 36 stocks<sup>12</sup> in New York Stock Exchange (NYSE) for a 22-year period from Jul. 3<sup>rd</sup> 1962 to Dec. 31<sup>st</sup> 1984.

The second dataset is an extended version of the NYSE dataset. For consistency, we collected the latest data in New York Stock Exchange (NYSE) from Jan. 1<sup>st</sup> 1985 to Jun. 30<sup>th</sup> 2010, which consists of 6431 trading days.

<sup>9</sup>We did evaluate certain algorithms using high frequency data and weekly data, as in Li et al. [45].

<sup>10</sup>Yahoo Finance: <http://finance.yahoo.com>; Google Finance: <http://www.google.com/finance>.

<sup>11</sup>All related codes and the datasets, including their compositions, are available on <http://www.cais.ntu.edu.sg/~chhoi/olps>. Borodin et al. [8]’s datasets (NYSE (O), TSE, SP500, DJIA) are available on <http://www.cs.technion.ac.il/~rani/portfolios/>.

<sup>12</sup>According to Helmbold et al. [37], the dataset was originally collected by Hal Stern. The stocks are mainly large cap stocks in NYSE, however, we do not know the criteria of choosing these stocks.

We denote this new dataset as “NYSE (N)”<sup>13</sup>. Note that the new dataset consists of 23 stocks rather than previous 36 stocks owing to amalgamations and bankruptcies. All self-collected price relatives are adjusted for splits and dividends, which is consistent with the previous “NYSE (O)” dataset.

The third dataset “TSE” is collected by Borodin et al. [8], which consists of 88 stocks from Toronto Stock Exchange (TSE) containing price relatives of 1259 trading days, ranging from Jan. 4<sup>th</sup> 1994 to Dec. 31<sup>st</sup> 1998. The fourth dataset “SP500” is collected by Borodin et al. [8], which consists of 25 stocks with the largest market capitalizations in the 500 SP500 components. It ranges from Jan. 2<sup>nd</sup> 1998 to Jan. 31<sup>st</sup> 2003, containing 1276 trading days.

The fifth dataset is “MSCI”, which is a collection of global equity indices that constitute the MSCI World Index<sup>14</sup>. It contains 24 indices that represent the equity markets of 24 countries around the world, and consists of a total of 1043 trading days, ranging from Apr. 1<sup>st</sup> 2006 to Mar. 31<sup>st</sup> 2010. The final dataset is the “DJIA” dataset [8], which consists of 30 Dow Jones composite stocks. DJIA contains 507 trading days, ranging from Jan. 14<sup>th</sup> 2001 to Jan. 14<sup>th</sup> 2003.

Unlike previous studies, the above testbed covers much longer trading periods from 1962 to 2010 and much more diversified markets, which enables us to examine the behaviors of the proposed strategies under different events and crises. For example, it covers several well-known events in the stock markets, such as dot-com bubble from 1995 to 2000 and subprime mortgage crisis from 2007 to 2009. The five stock datasets are mainly chosen to test the capability of the proposed algorithms on regional stock markets, while the index dataset aims to test their capability on global indices, which may be potentially applicable to “Fund on Fund” (FOF)<sup>15</sup>. As a remark, although we numerically test the proposed algorithms on stock and ETF markets, we note that the proposed strategy could be generally applied to any types of financial markets.

## 5.2. Experimental Setup and Metrics

In our experiments, we implement the proposed OLMAR, and its BAH version BAH(OLMAR). For OLMAR, we denote algorithms by their moving average calculation (MAR-1/2). For example, OLMAR-2 denotes the algorithm with MAR2 prediction. In all cases, we empirically set the parameters, that is,  $\epsilon = 10$  and  $w = 5$ , which provide a consistent results for OLMAR in all cases. For BAH(OLMAR), we set their maximum window size  $W = 30$ , resulting in 28 OLMAR experts with  $w = 3$  to 30. In addition to the empirical selection of parameters, we also evaluate the parameter scalability of the two possible parameters in Section 5.6. Finally, we also refer the best performance (in hindsight) among the underlying experts of the BAH version, named MAX(OLMAR).

We adopt the most common metric, *cumulative wealth*, to primarily compare different trading strategies. In addition to the cumulative wealth, we also adopt *annualized Sharpe Ratio* (SR) to compare the performance of different

---

<sup>13</sup>The dataset before 2007 was collected by Gábor Gelencsér (<http://www.cs.bme.hu/~oti/portfolio>), we collected the remaining data from 2007 to 2010 via Yahoo Finance.

<sup>14</sup>The constituents of MSCI World Index are available on MSCI Barra (<http://www.msribarra.com>), accessed on 28 May 2010.

<sup>15</sup>Note that not every index is tradable through Exchange Traded Funds (ETFs).

trading algorithms. In general, higher values of the cumulative wealth and annualized Sharpe Ratio indicate better algorithms. Besides, we also adopt *Maximum Drawdown* (MDD) and *Calmar Ratio* (CR) for analyzing a strategy's downside risk. The lower the MDD values, the less the strategy's (downside) risk is. The higher the CR values, the better the strategy's (downside) risk-adjusted return is. We summarize them in Table 5.

Criteria	Performance Metrics	
Absolute return	Cumulative wealth ( $S_n$ )	Annualized percentage yield
Risk	Annualized standard deviation	Maximum drawdown
Risk-adjusted return	Annualized sharpe ratio (SR)	Calmar ratio (CR)

Table 5: Summary of the performance metrics used in the evaluations.

Besides, we also introduce two practical issues in on-line portfolio selection, that is, margin buying and transaction costs, in Section 5.2.1, and further evaluate them in Section 5.7.

### 5.2.1. Practical Issues

While our model in Section 2 is concise and not complicated to understand, it omits some practical issues in the industry. We shall now relax some constraints to address these issues.

In reality, an important and unavoidable issue is *transaction cost*, which includes the commission fees and taxes imposed by brokers and governments, during the rebalance activities<sup>16</sup>. Note that the transaction cost is imposed by markets, and a portfolio's behavior can not change the properties of transaction costs, such as commission rates or tax rates. To handle the issue, the first way, which is commonly adopted by existing strategies, is that a portfolio selection model doesn't take transaction cost into account, and the second way is to directly integrate the costs in the model [32]. In this article, we take the first way and adopt the *proportional transaction cost* model [4]. To be specific, rebalancing a portfolio incurs transaction costs on every buy and sell operation, based upon a transaction cost rate of  $\gamma \in (0, 1)$ . At the beginning of period  $t$ , the portfolio manager rebalances his/her wealth to the new portfolio  $\mathbf{b}_t$ , from last close price adjusted portfolio  $\hat{\mathbf{b}}_{t-1}$ , each component of which is calculated as  $\hat{b}_{t-1,i} = \frac{b_{t-1,i} * x_{t-1,i}}{\mathbf{b}_{t-1}^T \mathbf{x}_{t-1}}$ . Such rebalance incurs a transaction cost of  $\frac{\gamma}{2} \times \sum_{i=1}^m |b_{t,i} - \hat{b}_{t-1,i}|$ , where the initial portfolio is set to  $(0, \dots, 0)$ . Thus, the cumulative wealth after  $n$  periods can be expressed as,

$$S_n^\gamma = S_0 \prod_{t=1}^n \left[ (\mathbf{b}_t \cdot \mathbf{x}_t) \times \left( 1 - \frac{\gamma}{2} \times \sum_{i=1}^m |b_{t,i} - \hat{b}_{t-1,i}| \right) \right].$$

Another practical issue is *margin buying*, which allows the portfolio managers to buy securities with cash borrowed from securities brokers, using their own equity positions as collateral. Following existing studies [14, 37, 1], we relax this constraint and evaluate it empirically. We assume the margin setting to be 50% down and 50% loan<sup>17</sup>, at an annual

<sup>16</sup>Besides commission and taxes, some other factors, such as bid-ask spreads, also implicitly incur transaction costs to a portfolio.

<sup>17</sup>That is, if one has \$ 100 stock (down or collateral), he/she can at most borrow \$ 100 cash (loan).

interest rate of 6% (equivalently, the corresponding daily interest rate of borrowing,  $c$ , is set to 0.000238). With such a setting, a new asset named “margin component” is generated for each asset and its price relative for period  $t$  equals  $2 * x_{t,i} - 1 - c$ . As a result, the total number of assets becomes  $2m$ , when margin buying is allowed. Adding such “margin component”, we can magnify both the potential profit or loss on the  $i^{th}$  asset.

### 5.3. Comparison Approaches

We compare the proposed algorithms with a number of benchmarks and representative strategies introduced in Section 3. Below we summarize a list of compared algorithms, all of which provide extensive empirical evaluations in their respective studies. All parameters are set following their original studies<sup>18</sup>. Focusing on empirical studies, we ignore certain algorithms that focus on theoretical analysis and lack thorough empirical evolutions.

1. Market: Market strategy, that is, uniform Buy-And-Hold (BAH) strategy;
2. Best-Stock: Best stock in the market, which is a strategy in hindsight;
3. BCRP: Best Constant Rebalanced Portfolios strategy in hindsight;
4. UP: Cover’s Universal Portfolios implemented according to Kalai and Vempala [40], where the parameters are set as  $\delta_0 = 0.004$ ,  $\delta = 0.005$ ,  $m = 100$ , and  $S = 500$ ;
5. EG: Exponential Gradient algorithm with the best learning rate  $\eta = 0.05$  as suggested by Helmbold et al. [37];
6. ONS: Online Newton Step with the parameters suggested by Agarwal et al. [1], that is,  $\eta = 0$ ,  $\beta = 1$ ,  $\gamma = \frac{1}{8}$ ;
7. Anticor:  $BAH_{30}(\text{Anticor}(\text{Anticor}))$  as a variant of Anticor to smooth the performance, which achieves the best performance among the three solutions proposed by Borodin et al. [8];
8.  $B^K$ : Nonparametric kernel-based moving window strategy with  $W = 5$ ,  $L = 10$  and threshold  $c = 1.0$ , which has the best empirical performance according to Györfi et al. [27];
9.  $B^{NN}$ : Nonparametric nearest neighbor-based strategy with parameters  $W = 5$ ,  $L = 10$  and  $p_\ell = 0.02 + 0.5 \frac{\ell-1}{L-1}$  as the authors suggested [30];
10. CORN: Correlation-driven nonparametric learning approaches [44] with parameters  $W = 5$ ,  $P = 1$ ,  $\rho = 0.1$ ;
11. PAMR: Passive aggressive mean reversion [46] with parameters  $\epsilon = 0.5$ ;
12. CWMR: Confidence weighted mean reversion [45] algorithm (exact version) with  $\epsilon = 0.5$ .

### 5.4. Experimental Results — Cumulative Wealth

Table 6 illustrates the main results of this study, that is, the cumulative wealth on the six datasets achieved by various approaches. The results clearly show that OLMAR achieves the best performance among all competitors. On the well-known benchmark NYSE (O) dataset, OLMAR significantly outperforms the state of the art; similar observations also appear on the successive NYSE (N) and other datasets. In addition, though most existing algorithms except Anticor perform bad on the DJIA dataset, OLMAR achieves the best performance, which validates the motivating

---

<sup>18</sup>We can tune their parameters for better performance, but it is beyond the scope of this article.

Table 6: Cumulative wealth achieved by various trading strategies on the six datasets. The top two results (excluding the best experts at the bottom, which is in hindsight) on each dataset are highlighted in **bold**.

Methods	NYSE (O)	NYSE (N)	DJIA	TSE	SP500	MSCI
Market	14.50	18.06	0.76	1.61	1.34	0.91
Best-stock	54.14	83.51	1.19	6.28	3.78	1.50
BCRP	250.60	120.32	1.24	6.78	4.07	1.51
UP	26.68	31.49	0.81	1.60	1.62	0.92
EG	27.09	31.00	0.81	1.59	1.63	0.93
ONS	109.19	21.59	1.53	1.62	3.34	0.86
$B^K$	1.08E+09	4.64E+03	0.68	1.62	2.24	2.64
$B^{NN}$	3.35E+11	6.80E+04	0.88	2.27	3.07	13.47
CORN	1.48E+13	5.37E+05	0.84	3.56	6.35	<b>26.10</b>
Anticor	2.41E+08	6.21E+06	<b>2.29</b>	39.36	5.89	3.22
PAMR	5.14E+15	1.25E+06	0.68	264.86	5.09	15.23
CWMR	6.49E+15	1.41E+06	0.68	332.62	5.90	17.28
OLMAR-S	<b>3.68E+16</b>	<b>2.54E+08</b>	2.12	<b>424.80</b>	5.83	16.39
OLMAR-E	<b>1.09E+18</b>	<b>5.10E+08</b>	1.20	<b>678.44</b>	<b>8.63</b>	<b>21.21</b>
BAH(OLMAR-S)	2.27E+16	1.41E+08	<b>2.38</b>	172.11	6.05	12.68
BAH(OLMAR-E)	5.62E+17	7.36E+08	1.31	732.81	<b>8.57</b>	20.01
MAX(OLMAR)	1.62E+17	3.95E+08	3.30	1.18E+03	20.91	25.49
MAX(OLMAR)	2.19E+18	2.84E+09	3.00	2.60E+03	14.63	27.05

Table 7: Statistical  $t$ -test of the performance achieved by OLMAR on the stock datasets. MER denotes “Mean Excess Return”.

Stat. Attr.	NYSE (O)	NYSE (N)	DJIA	TSE	SP500	MSCI
Size	5651	6431	507	1259	1276	1043
MER (OLMAR)	0.0074	0.0036	0.0020	0.0061	0.0019	0.0030
MER (Market)	0.0005	0.0005	-0.0004	0.0004	0.0003	0.0000
Winning Ratio	56.91%	53.13%	52.47%	55.12%	51.49%	58.39%
$\alpha$	0.0068	0.0030	0.0025	0.0056	0.0015	0.0030
$\beta$	1.2965	1.1768	1.2627	1.5320	1.2854	1.1763
$t$ -statistics	15.2405	7.3704	2.1271	3.4583	1.9423	5.8387
$p$ -value	0.0000	0.0000	0.0169	0.0003	0.0262	0.0000

idea of the proposed algorithm. Besides, the proposed OLMAR-2, which exploits exponential moving average, can further improve the performance, as it achieves the best performance on most datasets. Moreover, the MAXs in hindsight show that it is feasible to achieve better performance via effective expert combination. Finally, we list the  $t$ -test statistics [26] in Table 7, which validate that the achieved results are not due to luck, whose chances are at most 1.69% on the DJIA dataset, and at most 2.62% on the SP500 dataset. In a word, the proposed OLMAR can outperform the state of the art on the widely used datasets with a high confidence.

### 5.5. Experimental Results — Risk-Adjusted Return

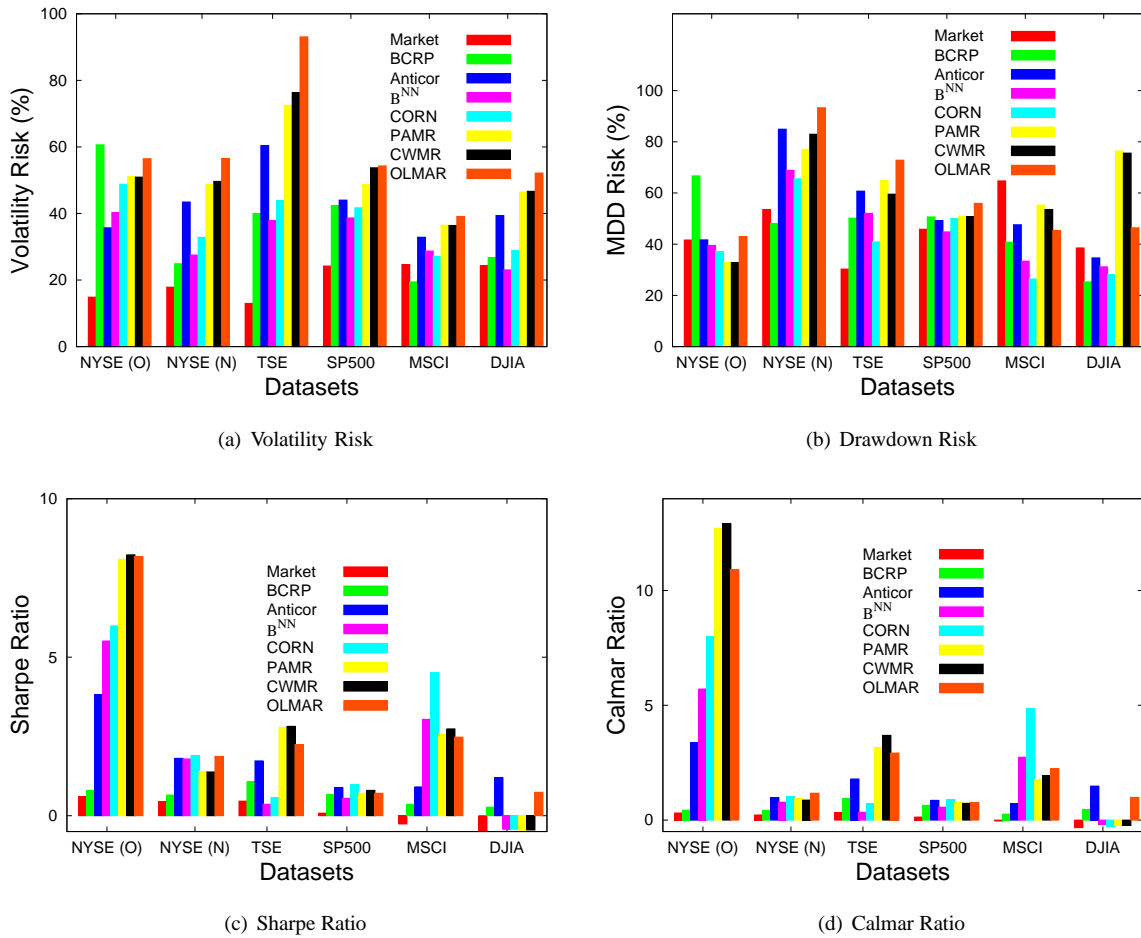


Figure 1: Risk and risk-adjusted performance of various strategies on the six different datasets. In each diagram, the rightmost bar represents the results of our proposed strategies.

We now evaluate the volatility risk and drawdown risk, and the risk-adjusted return in terms of annualized Sharpe ratio and Calmar ratio. Figure 1 shows the evaluation results on the six datasets. In addition to the proposed algorithms, we also plot two benchmarks (Market and BCRP) and five state-of-the-art algorithms (Anticor,  $B^{NN}$ , CORN, PAMR and CWMR).

In the preceding results on cumulative wealth, we find that the proposed OLMAR achieves the highest cumulative return on most datasets. However, high return is associated with high risk, as no real financial instruments can guarantee high return without high risk. The volatility risk in Figure 1(a) shows that the proposed method almost achieves the highest risk in terms of volatility risk on most datasets. In addition, the drawdown risk in Figure 1(b) also shows that the proposed method also achieves high drawdown risk in most datasets. These results are not difficult to understand since high return is often associated with high risk.

To further evaluate the return and risk, we examine the risk-adjusted return in terms of annualized Sharpe ratio and Calmar ratio. Despite the high risk, OLMAR yields a highly competitive result of risk-adjusted return. The results in Figure 1(c) and Figure 1(d) clearly show that OLMAR is always among the tops in most cases, including DJIA dataset. These encouraging results show that the proposed method is able to reach a good trade-off between return and risk, even though we do not explicitly consider risk in our method formulation.

### 5.6. Parameter Sensitivity

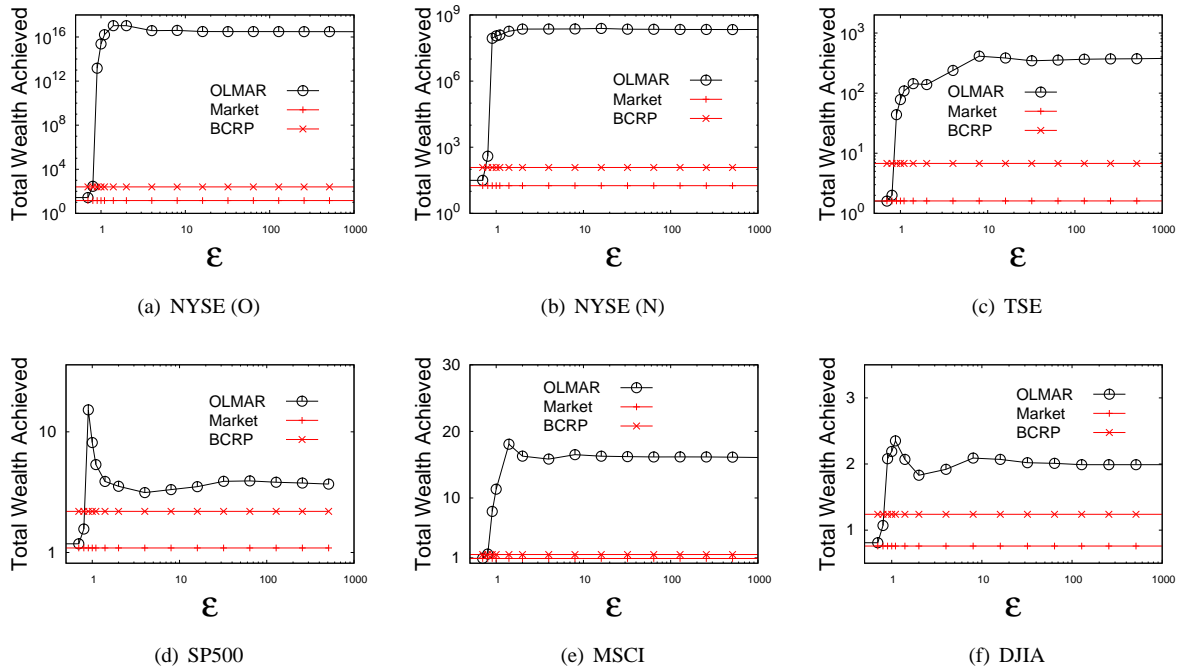


Figure 2: Parameter sensitivity of OLMAR-1 with respect to  $\epsilon$  with fixed  $w$  ( $w = 5$ ).

Now we evaluate OLMAR's sensitivity to its parameters, that is,  $\epsilon$  for both OLMAR-1 and OLMAR-2, and  $w$  and  $\alpha$  for OLMAR-1 and OLMAR-2, respectively. Figure 2 shows OLMAR-1's sensitivity of  $\epsilon$  with fixed  $w = 5$  and Figure 3 shows its sensitivity of  $w$  with fixed  $\epsilon = 10$ . Since OLMAR-2 and OLMAR-1 have the similar figures for  $\epsilon$ , we only list its effect on OLMAR-1. Figure 4 shows OLMAR-2's sensitivity of  $\alpha$  with fixed  $\epsilon = 10$ .

From Figure 2, we can observe that in general the cumulative wealth sharply increases if  $\epsilon$  approaches 1 and flattens if  $\epsilon$  crosses a threshold. From Figure 3, we can see that as  $w$  increases, the performance initially increases,

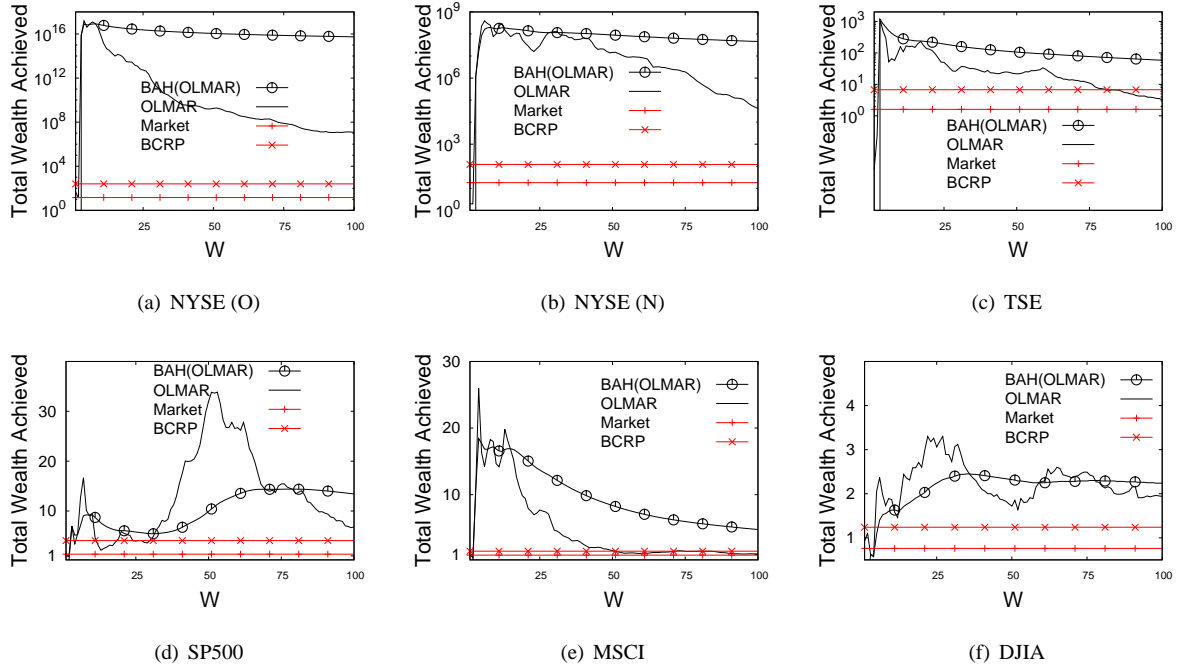


Figure 3: Parameter sensitivity of OLMAR-1 with respect to  $w$  with fixed  $\epsilon$  ( $\epsilon = 10$ ).

spikes at a data-dependant value, and then decreases. Its performance with most choices of  $\epsilon$  and  $w$  have much better performance than market and BCRP. To smooth the volatility of its performance, Figure 3 also shows its Buy and Hold versions [42] by combining a set of OLMAR experts with varying  $w$ s. As we can see that the BAH version provides a much more smoother cumulative return than its underlying experts. Note that on DJIA, OLMAR performs much better than PAMR/CWMMR as  $\epsilon$  varies, which validates the motivating multi-period mean reversion. From Figure 4, we can find that on most datasets,  $\alpha$  provides significant high cumulative wealth in a wide range of values, except the two extreme endpoints, that is, 0 and 1. If  $\alpha = 1$ , then all expected price relatives are always 1 and OLMAR-2 outputs  $\mathbf{b}_{t+1} = \mathbf{b}_t$ , which is initialized to uniform portfolio. If  $\alpha = 0$ , then its expected price relative vector equals  $\tilde{\mathbf{x}}_{t+1} = \frac{1}{\bigotimes_{i=1}^t \mathbf{x}_i}$ . Such price relatives inversely relate to asset's all historical price relatives, and produce bad results.

As a summary, OLMARs' performance is insensitive to  $\epsilon$  and thus it is easy to choose appropriate values of  $\epsilon$ . On the other hand, OLMARs' performance is sensitive to  $w$ . However, as shown in the figure, we can combine multiple OLMARs with different  $w < W$  such that the combined version is insensitive to  $w$ , and then we can easily choose a proper value of  $W$ .

### 5.7. Practical Performance under Transaction Cost and Margin Buying

To evaluate the practical applicability, we firstly evaluate the performance of the proposed algorithms with respect to proportional transaction cost [8]. Figure 5 illustrates the cumulative wealth achieved by OLMAR with increasing transaction cost rate  $\gamma$ , and also the results obtained by six representative algorithms (two benchmarks and four state of the art). On the one hand, the results clearly show that OLMAR can withstand reasonable transaction costs, as it often



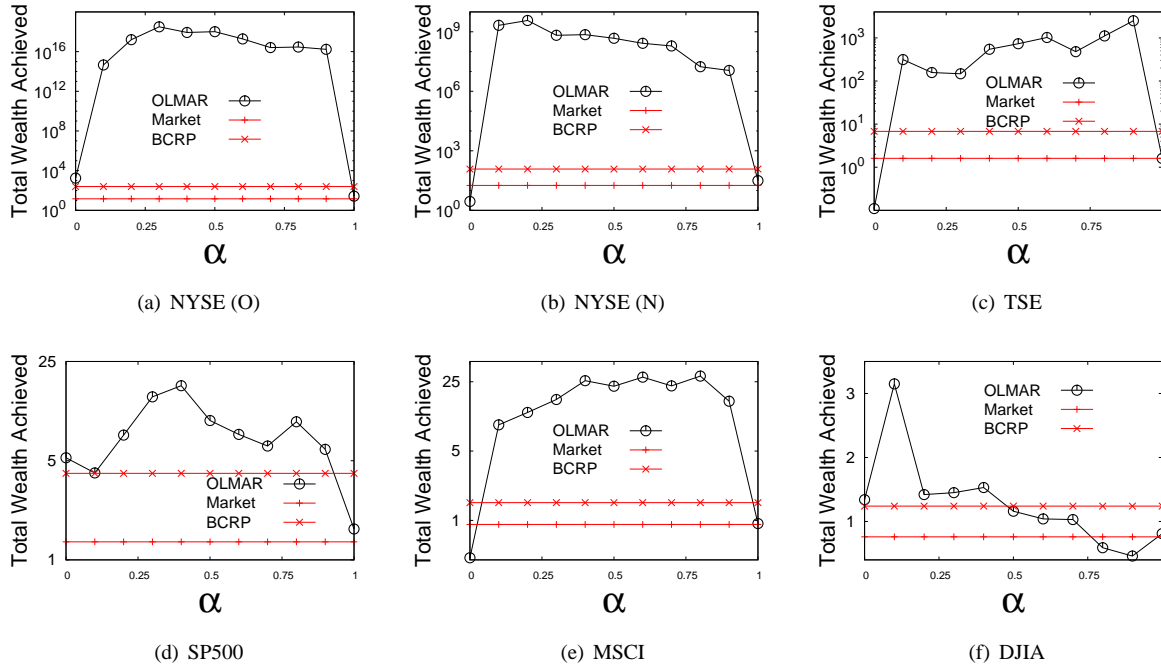


Figure 4: Parameter sensitivity of OLMAR-2 with respect to  $\alpha$  with fixed  $\epsilon$  ( $\epsilon = 10$ ).

has high break-even rates with respect to the market. On the other hand, OLMAR can outperform the benchmarks and state of the art, under various transaction cost rates. In a word, OLMAR performs excellent when trading is not frictionless, supporting its practical applicability.

Second, margin buying is another practical concern for a real-world portfolio selection task. Theoretically, each algorithm still has the opportunity to trade without using margin, so it shouldn't perform particularly worse with an increase in the available set of actions. But, in online learning context, increasing the set of hypotheses can hurt regret (and hence performance), though only slightly. To evaluate the impact of margin buying, we adopt the model described in Section 5.2.1, and present the cumulative wealth achieved by the competing approaches with/without margin buying in Table 8. The results clear show that if margin buying is allowed, the profitability of the proposed algorithms on most datasets increases. Similar to the results without margin buying, the proposed OLMAR algorithm often achieves the best results with margin buying. In summary, the proposed strategy can be extended to handle margin buying issue and benefit from it, and thus are practical applicable.

### 5.8. Evaluation of Computational Time Cost

Finally, we evaluate the computational time cost of the proposed algorithm. Table 9 shows their running time across six different data sets. Note that we only list the computational times of methods with comparable performance. As theoretically analyzed in Section 4.3, OLMAR enjoys linear computational time complexity. Empirically, as shown in the table, OLMAR algorithm takes the least time cost on all datasets. Note that with daily frequency, competitors'

Table 8: Cumulative wealth achieved by various strategies on the six datasets without and with margin loans (ML).

Algorithms	NYSE (O)		NYSE (N)		TSE	
	No ML	with ML	No ML	with ML	No ML	with ML
Market	14.50	15.75	18.06	17.68	1.61	1.71
Best-stock	54.14	54.14	83.51	173.18	6.28	10.53
BCRP	250.6	3755.09	120.32	893.63	6.78	21.23
UP	27.41	62.99	31.49	57.03	1.60	1.69
EG	27.09	63.28	31.00	55.55	1.59	1.68
ONS	109.19	517.21	21.59	228.37	1.62	0.88
Anticor	2.41E+08	1.05E+15	6.21E+06	5.41E+09	39.36	18.69
B <sup>K</sup>	1.08E+09	6.29E+15	4.64E+03	3.72E+06	1.62	1.53
B <sup>NN</sup>	3.35E+11	3.17E+20	6.80E+04	5.58E+07	2.27	2.17
CORN	1.48E+13	6.59E+25	5.37E+05	7.31E+07	3.56	5.00
PAMR	5.14+15	5.57E+25	1.25E+06	1.12E+09	264.86	720.42
CWMR	6.49E+15	6.59E+25	1.41E+06	7.31E+07	332.62	172.36
OLMAR	3.68E+16	5.67E+30	2.54E+08	1.73E+12	424.80	31.63
Algorithms	SP500		MSCI		DJIA	
	No ML	with ML	No ML	with ML	No ML	with ML
Market	1.34	1.03	0.91	0.69	0.76	0.59
Best-stock	3.78	3.78	1.50	1.50	1.19	1.19
BCRP	4.07	6.48	1.51	1.54	1.24	1.24
UP	1.62	1.75	0.92	0.71	0.81	0.66
EG	1.63	1.70	0.93	0.72	0.81	0.65
ONS	3.34	7.76	0.86	0.33	1.53	2.21
Anticor	5.89	10.73	3.22	3.40	2.29	2.89
B <sup>K</sup>	2.24	1.88	2.64	6.56	0.68	0.56
B <sup>NN</sup>	3.07	3.29	14.47	150.49	0.88	0.67
CORN	6.35	14.59	26.10	835.08	0.84	0.55
PAMR	5.09	15.91	15.23	68.83	0.68	0.84
CWMR	5.90	23.50	17.28	76.29	0.68	0.88
OLMAR	5.83	5.60	16.39	57.79	2.12	1.46

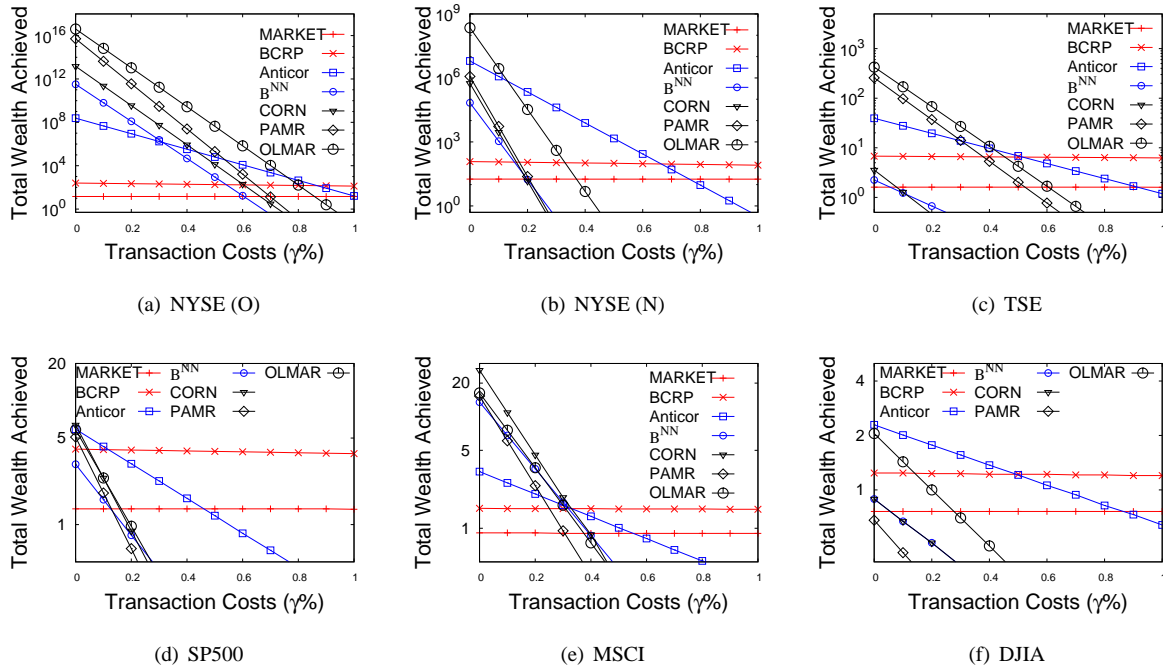


Figure 5: Scalability of the proposed strategies with respect to transaction cost rate ( $\gamma$ ).

average times are acceptable. However, their times are not acceptable in the scenario of high frequency trading [2]. Such time efficiency supports OLMAR’s large-scale real applications.

## 6. Discussions and Threats to Validity

Without theoretical guarantee, the empirical assumptions in Section 2 are worth inspecting, as done by all existing heuristic algorithms [8, 44, 46, 45]. As evaluated in Section 5.7, OLMAR can withstand reasonable rates of transaction cost. Note that the transaction cost rate in real market is low<sup>19</sup>, and there exists various methods to reduce the rate for institutional investors. All the datasets are composed of the largest index composite stocks, which have the highest market liquidity. To limit market impact when portfolio is too big, one solution is to scale-down the portfolio, as done by several quantitative funds. Here, we emphasize again that even in such a “perfect market”, no algorithm has ever claimed such good performance.

Backtests in historical markets may suffer from “data-snooping bias” issues, one of which is the dataset selection issue. Note that we selected all datasets based on previous studies without consideration to the proposed approaches. However, even we are cautious about the dataset selection issue, it may still appear in the experiments, especially for the datasets with relatively long history, that is, NYSE (O) and NYSE (N). Although they provide fair comparison

<sup>19</sup>For example, without considering taxes and bid-ask, Interactive Broker ([www.interactivebrokers.com](http://www.interactivebrokers.com)) charges 0.005\$ per share. Since the average price of Dow Jones Composites is around 50\$ (Accessed on June 2011), the transaction cost rate is about 0.01%.

Algorithms	NYSE (O)	NYSE (N)	TSE	SP500	MSCI	DJIA
Anticor	2.57E+03	1.93E+03	2.15E+03	387	306	175
B <sup>K</sup>	7.89E+04	5.78E+04	6.35E+03	1.95E+03	2.60E+03	802
B <sup>NN</sup>	4.93E+04	3.39E+04	1.32E+03	2.91E+03	2.55E+03	1.28E+03
CORN	8.78E+03	1.03E+04	1.59E+03	563	444	172
PAMR	8	7	2	1.1	1.0	0.3
CWMR	12	11	3	1.4	1.3	0.5
OLMAR	4	3	0.7	0.6	0.5	0.3

Table 9: Computational time cost (in seconds) on the six real datasets.

for different algorithms, they are worse than others in terms of survival bias. In a word, even though the empirical results on these datasets clearly show the effectiveness of the proposed algorithms, one can not make claims without noticing the deficiencies of these datasets. Another common bias is asset selection issue. We try to avoid the asset selection bias via arbitrarily choosing some representative stocks in their respective markets, which usually have large capitalization and high liquidity and thus reduce the the market impact caused by any proposed portfolio strategy.

## 7. Conclusions

This paper investigated the on-line portfolio selection problem and presented a novel strategy named “On-Line Moving Average Reversion” (OLMAR), which exploits “Moving Average Reversion” using on-line machine learning algorithms. The proposed approach is able to overcome the limitations of the existing state-of-the-art mean reversion strategies mainly due to the single-period assumption, and achieve fairly encouraging results on several data sets from real-world financial markets. Last but not least, the proposed algorithm runs extremely fast and is thus suitable for large-scale real-world financial applications. Despite achieving encouraging empirical results, several open issues are worth further exploration in future work. First of all, it will be interesting to explore theoretical analysis of the mean reversion property and investigate if it is possible to bound the worst-case performance of OLMAR. Besides, it may be possible to anatomize the portfolio’s expected return to better understand the major sources of profit. Although pure trading strategies have been commonly anatomized<sup>20</sup> in the finance fields [12] for a long time, on-line portfolio selection algorithms have seldom been anatomized in a systematic way, which certainly is worth further investigations.

## Appendix A: Proof of Proposition 1

*Proof.* Since introducing nonnegative constraint of the simplex constraint causes much difficulty [37], at first we do not consider it and finally project on the simplex domain.

<sup>20</sup>In finance, the term “anatomy” means investigating the sources of a strategy’s profit.

The Lagrangian of optimization problem OLMAR is

$$\mathcal{L}(\mathbf{b}, \lambda, \eta) = \frac{1}{2} \|\mathbf{b} - \mathbf{b}_t\|^2 + \lambda(\epsilon - \mathbf{b} \cdot \tilde{\mathbf{x}}_{t+1}) + \eta(\mathbf{b} \cdot \mathbf{1} - 1),$$

where  $\lambda \geq 0$  and  $\eta$  are the Lagrangian multipliers. Taking the gradient with respect to  $\mathbf{b}$  and setting it to zero, we get,

$$0 = \frac{\partial \mathcal{L}}{\partial \mathbf{b}} = (\mathbf{b} - \mathbf{b}_t) - \lambda \tilde{\mathbf{x}}_{t+1} + \eta \mathbf{1} \implies \mathbf{b} = \mathbf{b}_t + \lambda \tilde{\mathbf{x}}_{t+1} - \eta \mathbf{1},$$

Multiplying both sides by  $\mathbf{1}^\top$ , we get,

$$1 = 1 + \lambda \tilde{\mathbf{x}}_{t+1} \cdot \mathbf{1} - \eta m \implies \eta = \lambda \bar{x}_{t+1},$$

where  $\bar{x}_{t+1}$  denotes the average predicted price relative (market). Plugging the above equation to the update of  $\mathbf{b}$ , we get the update of  $\mathbf{b}$ ,

$$\mathbf{b} = \mathbf{b}_t + \lambda(\tilde{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1}),$$

To solve the Lagrangian multiplier, let us plug above equation to the Lagrangian,

$$\mathcal{L}(\lambda) = \lambda(\epsilon - \mathbf{b}_t \cdot \tilde{\mathbf{x}}_{t+1}) - \frac{1}{2} \lambda^2 \|\tilde{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1}\|^2$$

Taking derivative with respect to  $\lambda$  and setting it to zero, we get

$$0 = \frac{\partial \mathcal{L}}{\partial \lambda} = (\epsilon - \mathbf{b}_t \cdot \tilde{\mathbf{x}}_{t+1}) - \lambda \|\tilde{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1}\|^2 \implies \lambda = \frac{\epsilon - \mathbf{b}_t \cdot \tilde{\mathbf{x}}_{t+1}}{\|\tilde{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1}\|^2}.$$

Further projecting the  $\lambda$  to  $[0, +\infty)$ , we get ,

$$\lambda = \max \left\{ 0, \frac{\epsilon - \mathbf{b}_t \cdot \tilde{\mathbf{x}}_{t+1}}{\|\tilde{\mathbf{x}}_{t+1} - \bar{x}_{t+1} \mathbf{1}\|^2} \right\}.$$

□

## Acknowledgments

We want to thank the associate editor and anonymous reviewers for their helpful comments and suggestions. This work was in part supported by the National Science Foundation of China (71401128) and “the Fundamental Research Funds for the Central Universities”, and the tier-1 research grant of Singapore Management University.

## References

- [1] Agarwal, A., Hazan, E., Kale, S., Schapire, R. E., 2006. Algorithms for portfolio management based on the newton method. In: Proceedings of International Conference on Machine Learning.
- [2] Aldridge, I., 2010. High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems. Hoboken, NJ : Wiley.

- [3] Birge, J. R., Louveaux, F., 1997. Introduction to Stochastic Programming. New York : Springer.
- [4] Blum, A., Kalai, A., 1999. Universal portfolios with and without transaction costs. *Machine Learning* 35 (3), 193–205.
- [5] Blum, A., Mansour, Y., 2007. From external to internal regret. *Journal of Machine Learning Research* 8, 1307–1324.
- [6] Bondt, W. F. M. D., Thaler, R., 1985. Does the stock market overreact? *The Journal of Finance* 40 (3), 793–805.
- [7] Borodin, A., El-Yaniv, R., Gogan, V., 2003. Can we learn to beat the best stock. In: *Proceedings of the Annual Conference on Neural Information Processing Systems*.
- [8] Borodin, A., El-Yaniv, R., Gogan, V., 2004. Can we learn to beat the best stock. *Journal of Artificial Intelligence Research* 21, 579–594.
- [9] Boyd, S., Vandenberghe, L., 2004. *Convex Optimization*. New York : Cambridge University Press.
- [10] Breiman, L., 1961. Optimal gambling systems for favorable games. *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability* 1, 65–78.
- [11] Campbell, J. Y., Lo, A. W., Mackinlay, A. C., 1996. *The Econometrics of Financial Markets*. Princeton, NJ : Princeton University Press.
- [12] Conrad, J., Kaul, G., 1998. An anatomy of trading strategies. *Review of Financial Studies* 11 (3), 489–519.
- [13] Cont, R., 2001. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance* 1 (2), 223–236.
- [14] Cover, T. M., 1991. Universal portfolios. *Mathematical Finance* 1 (1), 1–29.
- [15] Cover, T. M., Gluss, D. H., 1986. Empirical bayes stock market portfolios. *Advances in Applied Mathematics* 7 (2), 170–181.
- [16] Cover, T. M., Thomas, J. A., 1991. *Elements of information theory*. New York : Wiley.
- [17] Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y., 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research* 7, 551–585.
- [18] Crammer, K., Dredze, M., Kulesza, A., 2009. Multi-class confidence weighted algorithms. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.
- [19] Crammer, K., Dredze, M., Pereira, F., 2008. Exact convex confidence-weighted learning. In: *Proceedings of annual conference on Neural Information Processing Systems*.
- [20] Dredze, M., Crammer, K., Pereira, F., 2008. Confidence-weighted linear classification. In: *Proceedings of International Conference on Machine Learning*.
- [21] Dredze, M., Kulesza, A., Crammer, K., 2010. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning* 79 (1-2), 123–149.
- [22] Duchi, J., Shalev-Shwartz, S., Singer, Y., Chandra, T., 2008. Efficient projections onto the  $l_1$ -ball for learning in high dimensions. In: *Proceedings of International Conference on Machine Learning*.
- [23] Finkelstein, M., Whitley, R., 1981. Optimal strategies for repeated games. *Advances in Applied Probability* 13 (2), 415–428.
- [24] Gaivoronski, A. A., Stella, F., 2000. Stochastic nonstationary optimization for finding universal portfolios. *Annals of Operations Research* 100, 165–188.
- [25] Gaivoronski, A. A., Stella, F., 2003. On-line portfolio selection using stochastic programming. *Journal of Economic Dynamics and Control* 27 (6), 1013–1043.
- [26] Grinold, R., Kahn, R., 1999. *Active Portfolio Management: A Quantitative Approach for Producing Superior Returns and Controlling Risk*. New York : McGraw-Hill, New York.
- [27] Györfi, L., Lugosi, G., Udina, F., 2006. Nonparametric kernel-based sequential investment strategies. *Mathematical Finance* 16 (2), 337–357.
- [28] Györfi, L., Ottucsák, G., Walk, H., 2012. *Machine Learning for Financial Engineering*. Singapore : World Scientific.
- [29] Györfi, L., Schäfer, D., 2003. Nonparametric prediction. In: *Advances in Learning Theory: Methods, Models and Applications*. Netherlands : IOS Press.
- [30] Györfi, L., Udina, F., Walk, H., 2008. Nonparametric nearest neighbor based empirical portfolio selection strategies. *Statistics and Decisions* 26 (2), 145–157.
- [31] Györfi, L., Urbán, A., Vajda, I., 2007. Kernel-based semi-log-optimal empirical portfolio selection strategies. *International Journal of Theoretical and Applied Finance* 10 (3), 505–516.

- [32] Györfi, L., Vajda, I., 2008. Growth optimal investment with transaction costs. In: Proceedings of the International Conference on Algorithmic Learning Theory.
- [33] Hazan, E., 2006. Efficient algorithms for online convex optimization and their applications. Ph.D. thesis, Princeton University.
- [34] Hazan, E., Agarwal, A., Kale, S., 2007. Logarithmic regret algorithms for online convex optimization. *Machine Learning* 69 (2-3), 169–192.
- [35] Hazan, E., Kalai, A., Kale, S., Agarwal, A., 2006. Logarithmic regret algorithms for online convex optimization. In: Proceedings of the Annual Conference on Learning Theory.
- [36] Helmbold, D. P., Schapire, 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.
- [37] Helmbold, D. P., Schapire, R. E., Singer, Y., Warmuth, M. K., 1998. On-line portfolio selection using multiplicative updates. *Mathematical Finance* 8 (4), 325–347.
- [38] Hoi, S. C. H., Wang, J., Zhao, P., 2014. Libol: A library for online learning algorithms. *Journal of Machine Learning Research* 15, 495–499. URL <http://LIBOL.stevenhoi.org/>
- [39] Huang, D., Zhou, J., Li, B., Hoi, S. C. H., Zhou, S., 2013. Robust median reversion strategy for on-line portfolio selection. In: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence. AAAI Press, pp. 2006–2012.
- [40] Kalai, A., Vempala, S., 2002. Efficient algorithms for universal portfolios. *Journal of Machine Learning Research* 3, 423–440.
- [41] Kelly, J. J., 1956. A new interpretation of information rate. *Bell Systems Technical Journal* 35, 917–926.
- [42] Li, B., Hoi, S. C. H., 2012. On-line portfolio selection with moving average reversion. In: Proceedings of the International Conference on Machine Learning.
- [43] Li, B., Hoi, S. C. H., 2014. Online portfolio selection: A survey. *ACM Computing Surveys* 36, 35:1–35:36.
- [44] Li, B., Hoi, S. C. H., Gopalkrishnan, V., 2011. Corn: Correlation-driven nonparametric learning approach for portfolio selection. *ACM Transactions on Intelligent Systems and Technology* 2 (3), 21:1–21:29.
- [45] Li, B., Hoi, S. C. H., Zhao, P., Gopalkrishnan, V., Mar. 2013. Confidence weighted mean reversion strategy for on-line portfolio selection. In: *ACM Transactions on Knowledge Discovery from Data*. Vol. 7. pp. 4:1–4:38.
- [46] Li, B., Zhao, P., Hoi, S. C. H., Gopalkrishnan, V., 2012. Pamr: Passive aggressive mean reversion strategy for portfolio selection. *Machine Learning* 87 (2), 221–258.
- [47] Lo, A. W., MacKinlay, A. C., 1990. When are contrarian profits due to stock market overreaction? *Review of Financial Studies* 3 (2), 175–205.
- [48] Markowitz, H., 1952. Portfolio selection. *The Journal of Finance* 7 (1), 77–91.
- [49] Ordentlich, E., 1996. Universal investment and universal data compression. Ph.D. thesis, Stanford University.
- [50] Ordentlich, E., 2010. *Universal Portfolios*. John Wiley & Sons, Ltd.
- [51] Ottucsák, G., Vajda, I., 2007. An asymptotic analysis of the mean-variance portfolio selection. *Statistics and Decisions* 25, 63–88.
- [52] Poterba, J. M., Summers, L. H., 1988. Mean reversion in stock prices : Evidence and implications. *Journal of Financial Economics* 22 (1), 27–59.
- [53] Shalev-Shwartz, S., Crammer, K., Dekel, O., Singer, Y., 2003. Online passive-aggressive algorithms. In: Proceedings of the Annual Conference on Neural Information Processing Systems.
- [54] Stoltz, G., Lugosi, G., 2005. Internal regret in on-line portfolio selection. *Machine Learning* 59 (1-2), 125–159.
- [55] Thorp, E. O., 1971. Portfolio choice and the Kelly criterion. In: Business and Economics Section of the American Statistical Association.
- [56] Vajda, I., 2006. Analysis of semi-log-optimal investment strategies. In: Proceedings of Prague Stochastic.
- [57] Zinkevich, M., 2003. Online convex programming and generalized infinitesimal gradient ascent. In: Proceedings of the International Conference on Machine Learning.