Long-and Short-Term Forecasting for Portfolio Selection with Transaction Costs

Guy Uziel
Department of Computer Science
Technion - Israel Institute of Technology
Haifa, Israel

Ran El-Yaniv
Department of Computer Science
Technion - Israel Institute of Technology
Haifa, Israel

Abstract

In this paper we focus on the problem of online portfolio selection with transaction costs. We tackle this problem using a novel approach for combining the predictions of long-term experts with those of short-term experts so as to effectively reduce transaction costs. We prove that the new strategy maintains bounded regret relative to the performance of the best possible combination (switching times) of the long-and short-term experts. We empirically validate our approach on several standard benchmark datasets. These studies indicate that the proposed approach achieves state-of-the-art performance.

1 Introduction

Online portfolio selection (Cover, 1991) remains a challenging open problem in online vector prediction. In this problem the learner maintains an online allocation vector, called a *portfolio*, specifying the fraction of wealth to be invested in all the stocks in the market. At the start of each trading period (e.g., a day), the learner receives the current prices of the stocks and submits his next day's portfolio to his broker. Useful portfolio vectors should allocate wealth to stocks that will rise on the following day and avoid allocations to falling stocks. This prediction problem turns out to be quite challenging when applied in reality with true stock prices, but there are several known portfolio selection algorithms that exhibit interesting performance on historical data (Györfi et al., 2007; Li and Hoi, 2012). The problem becomes significantly harder when trying to model other elements, such as brokerage transactions costs, which exist in realistic applications.

Classical portfolio selection algorithms such as Universal Portfolios (UP) (Cover, 1991; Blum and Kalai, 1999), EG (Helmbold et al., 1998) and Online Newton-Steps (ONS) (Hazan and Seshadhri, 2009) have been studied in the context of online learning and were analyzed using regret analysis, where the comparison class was the set of constant rebalancing portfolios (CRPs). Such algorithms can resist very large (proportional) transaction costs. The reason for this tolerance is that CRP portfolios change very little from day to day. More aggressive algorithms that achieve significantly better empirical performance in a market without transaction costs, such as Anticor (Borodin et al., 2004), Olmar (Li and Hoi, 2012) and kernel-based algorithms (Györfi et al., 2006), are very sensitive even to small transaction costs.

The two main approaches for adjusting a given algorithm to deal with transaction costs are a regularization (e.g., Das et al., 2013) and dilution of the trading days. The latter approach was motivated by an observation made by Blum and Kalai (1999) that a dilution in the number of trading days leads to a significant reduction in transaction costs. In this paper, we focus on the dilution approach and try to achieve meaningful dilution by combining the predictions of long-term experts. While we can reduce transaction costs by ignoring (diluting) trading periods, such dilutions invalidate some short-term predictions (and trading decisions). The inclusion of larger horizon predictions can alleviate this problem. Therefore, our goal in this paper will be to combine the best of all worlds. We consider "diluted" experts that are defined with respect to diluted price sequences (e.g., only trade every d days). This dilution makes them robust to larger transaction costs. They, however, are required to generate longer-term predictions, which are harder to make (and potentially adversely affect performance). To overcome this longer term prediction challenge, we also employ a short-term "emergency"

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

expert that observes the entire sequence (every day), and has the mandate to undo any long-term decision (recommended by any of the long-term experts) when it faces extreme events that conflict with these long-term predictions.

We present an algorithm called Long-and Short-term Portfolio Ensemble (LSPE) and prove that it is competitive with the best switching strategy in hindsight over periods defined by applications of the emergency expert. We explain how to implement this algorithm in linear time complexity and then present an empirical study of our strategy implemented as an ensemble over instances of the well-known OLMAR online portfolio selection algorithm (Li and Hoi, 2012). These experiments indicate that LSPE consistently outperforms the known commission-aware strategies.

While here we focus only on the long-standing problem of handling transaction costs in portfolio selection, our proposed framework offers a novel approach to deal with the problem of combining long-and short-term experts, and can be utilized in other online learning tasks in which predictions in varying time-scales are required. Potential applications can be found, for example, in weather forecasting.

2 Online Portfolio Selection

In Cover's classic portfolio selection setting (Cover, 1991), we are given a market with n stocks and consider an online game between an algorithm and an adversary played through T rounds (say, days). On each day t, the market is represented by a *market vector* $\mathbf{X}_t$ of relative prices, $\mathbf{X}_t \triangleq (x_1^t, x_2^t, ..., x_n^t)$, where for each $i = 1, \ldots, n$, $0 < c_1 \leq x_i^t \leq c_2$ for some constants c_1, c_2. The *relative price* of stock i is defined to be the ratio of its closing price on day t relative to its closing price on day $t-1$. We denote by $\mathbf{X}_0^{T-1} \triangleq \mathbf{X}_0, \ldots, \mathbf{X}_{T-1}$ the sequence of T market vectors starting at day 0. The algorithm's *portfolio* for day t is $\mathbf{b}_t \triangleq (b_1^t, b_2^t, \ldots, b_n^t)$, where $b_i^t \geq 0$ is the wealth allocation for stock i. We require that the portfolio satisfies $\sum_{i=1}^m b_i^t = 1$. Thus, $\mathbf{b}_t$ specifies the online player's wealth allocation for each of the n stocks on day t, and b_i^t is the fraction of total current wealth invested in stock i on that day. We denote by $\mathbf{B} \triangleq \mathbf{b}_0, \ldots, \mathbf{b}_{T-1}$ the sequence of T portfolios played by the algorithm for the entire game. The portfolio sequence where all $\mathbf{b}_i$ equal the same fixed portfolio is called a *constant rebalanced portfolio* (CRP). At the start of each trading day t, the algorithm chooses a portfolio $\mathbf{b}_t$. Thus, by the end of day t, the player's wealth is multiplied by $\langle \mathbf{b}_t, \mathbf{X}_t \rangle = \sum_{i=1}^n b_i^t x_i^t$, and assuming an initial wealth of \$1, the player's cumulative wealth by the end of the game is

$$R(\mathbf{B}, \mathbf{X}_0^{T-1}) \triangleq \prod_{t=0}^{T-1} \langle \mathbf{b}_t, \mathbf{X}_t \rangle . \qquad (1)$$

The goal in the online approach to portfolio learning is generating a sequence of portfolios $\mathbf{B}_{ALG}$, without the power of hindsight, which competes with the best fixed strategy A from reference class $\mathcal{A}$. We, therefore, define the regret w.r.t. reference class $\mathcal{A}$ as follows:

$$\mathbf{Regret}(\mathbf{B}_{ALG}, \mathcal{A}, T) \triangleq$$
$$\max_{A \in \mathcal{A}} \ln R(\mathbf{B}_A, \mathbf{X}_0^{T-1}) - \ln R(\mathbf{B}_{\mathbf{B}_{ALG}}, \mathbf{X}_0^{T-1}).$$

Accordingly, we aim to achieve sublinear regret w.r.t $\mathcal{A}$. In the case where this reference set is all the possible CRPs, the goal is to be competitive with the best-in-hindsight fixed portfolio, denoted $\mathbf{b}_*$, which is also known as the best constant rebalanced portfolio.[1] In this paper we are concerned with portfolio ensembles over trading strategies (experts). When dealing with portfolio ensembles, we thus choose a portfolio through our choice of trading algorithms (each of which selects its own portfolio).

2.1 Introducing Transaction Costs

We now turn to extend the vanilla portfolio selection model that abstracts away transaction costs. Transaction costs might include several elements in addition to the actual commissions paid to brokers, clearing firms and the exchange for participating in trades. These may include *slippage* (the price difference between the time we decide to buy/sell a security and the time the transaction is actually executed in the exchange) and *market impact costs*. The latter element is considered to be extremely challenging to model as it might be affected by factors such as the type of order used and the present liquidity in the limit order book. Modeling brokerage exchange commissions, however, follows a fixed and known schedule and, as a first approximation, it is common to apply a linear transaction cost model where each transaction incurs a cost proportional to its size (Blum and Kalai, 1999; Lobo et al., 2007).

We, therefore, focus on the following simple and common multiplicative (proportional) cost model, commonly used in the online portfolio selection literature (Borodin and El-Yaniv, 2005, Sec 14.5.4). In this model, commissions are specified via a fixed parameter, $0 < \gamma$, called the *commission rate* and for buying (or selling) \$$w$ worth of any stock, the player must pay

[1] Interestingly, BCRP is known to be better than any online strategy in a market governed by an i.i.d. process (Cover, 1991; Cover and Ordentlich, 1996).

a commission of $\$\frac{\gamma}{2}w$. Thus, the transaction cost incurred when the player rebalances a portfolio $\mathbf{b}$ to portfolio $\mathbf{b}'$ is $\frac{\gamma}{2}||\mathbf{b}-\mathbf{b}'||_1$. In the present transaction cost model, we assume that commissions are *self-refinanced*, which means that the player pays them immediately after performing the daily transactions. Thus, on day t, after rebalancing to portfolio $\mathbf{b}_t$, the market vector $\mathbf{X}_t$ is revealed and portfolio $\mathbf{b}_t$ becomes

$$\hat{\mathbf{b}}_t \triangleq \frac{1}{\langle \mathbf{b}_t, \mathbf{X}_t \rangle}(b_1x_1, b_2x_2, \ldots, b_nx_n). \tag{2}$$

Accordingly, the commission incurred to rebalance to the next day's portfolio, $\mathbf{b}_{t+1}$, is

$$\frac{\gamma}{2}||\mathbf{b}_{t+1} - \hat{\mathbf{b}}_t||_1, \tag{3}$$

which is paid from the current wealth, $\langle \mathbf{b}_t, \mathbf{X}_t \rangle$. Altogether, the cumulative wealth of a player paying commission at rate γ is

$$R^{\gamma}(\mathbf{B}, \mathbf{X}_0^{T-1}) = \prod_{t=0}^{T-1} \left(\langle \mathbf{b}_t, \mathbf{X}_t \rangle \left[1 - \frac{\gamma}{2}||\mathbf{b}_{t+1} - \hat{\mathbf{b}}_t||_1 \right] \right).$$

3 Related Work and Contributions

The study of portfolio optimization with transaction costs within mainstream finance is a huge topic, beyond our scope. Typically, such studies design solutions under specific distributions (Davis and Norman, 1990; Konno and Wijayanayake, 2001; Lobo et al., 2007). In the brief survey below we only refer to related works emerging from the online learning research initiated by Cover (1991).

Blum and Kalai (1999), as far as we know, were the first to study commissions in online portfolio selection and they presented an elegant regret analysis for Cover and Ordentlich's universal portfolios (UP) algorithm (Cover and Ordentlich, 1996), which pays proportional commissions.

The regularization approach for incorporating transaction costs was initiated in the papers of Das et al. (2013, 2014). Their approach, which is called Online Lazy Updates (OLU), proposes dealing with transaction costs by penalizing costly rebalancing using ℓ_1 regularization, which serves as a proxy for the true proportional transaction cost incurred by the update as given by Equation (3).

Uziel and El-Yaniv (2016) presented a method called Commission Avoidant Portfolio Ensemble (CAPE), which allows one to combine several commission-oblivious algorithms. Their regularization is in the form of a static expert that forces the algorithm to allocate more wealth to the previous portfolio. Li et al. (2017) presented the Transaction Costs Optimization (TCO) mechanism that allows transaction cost regularization to be added in the cases where the base algorithm has an explicit market vector prediction for the next period. Although TCO has no theoretical guarantee, its approach makes several algorithms more resilient to transaction costs and exhibit nice results on several datasets.

The Semi-Constant Rebalanced Portfolios (SCRP) algorithm, which operates by diluting the number of trading days for commission reduction, was studied by Kozat and Singer (2008, 2009) based on an observation made by Blum and Kalai (1999). The algorithms presented there track a fixed and a priori defined CRP. Huang et al. (2015) extended the above approach and proposed two strategies, SUP and SUP-q. These algorithms follow the best (global) CRP (SUP) and best horizon (q) CRP (SUP-q) instead of following a specific (given) CRP as SCRP does. The SUP algorithms are shown by Huang et al. (2015) to outperform SCRP on many random projections (over two stocks) of the NYSE and SP500 datasets.

Despite the fact that the SCRP-based methods discussed above are extremely resilient to transaction costs, they has several drawbacks. First, their empirical performance is only marginally better than a commission-oblivious BCRP tracking algorithm (e.g., UP). This observation is backed up by the results shown in Table 2. Another serious drawback is that the portfolio updates generated by these methods are too conservative and they are unable to track and utilize market dynamics such as mean reversion, which is utilized by the state-of-the-art algorithms (Borodin et al., 2004; Li et al., 2012; Huang et al., 2013; Li and Hoi, 2014). The third problematic issue follows from empirical observations that even the best CRP *computed in hindsight* is not a strong contender relative to other known algorithms such as several mean-reversion methods. This observation has been reported in numerous empirical studies including those by Borodin et al. (2004); Li et al. (2012); Li and Hoi (2012). Consequently, if high (empirical) performance is as an objective, it does not make sense to consider algorithms that try to track BCRP.

The goal in this work is exploit the useful observation made by Blum and Kalai (1999), that a dilution in the number of trading days results in a superior performance, while maintaining the flexibility to track state-of-the-art algorithms and not just the CRP class, even at the cost of reducing resiliency to some levels of transaction costs. We tackle this task using a novel mechanism for integrating and exploiting any kind of long- and short-term experts. Here we note that long-term prediction has been discussed in several papers before (Weinberger and Ordentlich, 2002; Burnaev et al., 2017). The main focus of those papers, however, was to deal with the delayed feedback of those experts since at *each* time instant, the player has to choose a prediction for d days. In other words, the feedback for the prediction made at time t is received only at time $t+d$. Our problem formulation is different, and in our setting we are not concerned with delayed feedback but rather with the ability of the learner to commit to his portfolio choice. This kind of problem arises in settings where long-term predictions are achievable only at some periods, and are impossible to make in others.

4 Long- and Short-Term Predictions

The framework We now describe the trading game in the presence of long- and short-term experts. At the start of the first trading day, $t = 0$, the learner must choose a long-term portfolio for day $d > 0$ from the set of l long-term portfolios proposed by its l long-term experts $\mathbf{b}^L_{1,0}, \ldots, \mathbf{b}^L_{l,0}$ (where $\mathbf{b}^L_{i,j}$ denotes the ith long-term expert portfolio introduced on day j). As will be discussed below, the learner chooses one of these long-term portfolios randomly. The chosen long-term portfolio is executed immediately and remains in effect until day $t = d$. The long-term experts thus receive feedback for their proposals only d days later. At the end of the first day, $t = 0$, the resulting market vector $\mathbf{X}_0$ is revealed to the learner, which has already committed to a long-term portfolio. At the start of the next day, $t = 1$, however, the learner can decide to update his present portfolio for day d, based on the short-term "emergency" expert $\mathbf{b}^S_1$. We denote the day t short-term portfolio of this expert by $\mathbf{b}^S_t$. Of course, any portfolio update (including an "emergency" update) incurs transaction costs and, therefore, the player should choose such emergency updates only in cases when the market moves aggressively against him.

In general, on day t, when $t \mod d = 0$, the player must choose a long-term portfolio from $\mathbf{b}^L_{1,t}, \ldots, \mathbf{b}^L_{l,t}$, and on any other day t, where $t \mod d \neq 0$, the player can choose to update to the short portfolio $\mathbf{b}^S_t$, a portfolio for the next $t \mod d = 0$ period. The learner's challenge is thus to balance between the need to use the short-term experts whose predictions might be more accurate and the need to save transaction costs, which is promoted by using long-term experts.

We define a *transition path* $\mathbf{S}_{k,T}$ as a vector $(t_1, \ldots, t_k)$ of transition day indices where the learner performed short-term portfolio updates. Thus, it contains k short-term updates in a game of length T. The set of all transitions with k short-term updates in a game of length T is denoted by $\mathcal{S}_{k,T}$ and the set of all the possible transitions of length T is denoted by $\mathcal{S}_{\cdot,T}$. For a given $\mathbf{S}_{k,T} \in \mathcal{S}_{k,T}$, we denote by $\mathbf{b}^t_{\mathbf{S}_{k,T}}$ the portfolio used by this transition path at time t.

A transition path $\mathbf{S}_{k,T}$ divides the entire trading period into $k+1$ contiguous segments called *investing periods*:

$$(\mathbf{X}_{t_0}, \ldots, \mathbf{X}_{t_1}), (\mathbf{X}_{t_1}, \ldots, \mathbf{X}_{t_2}), \ldots, (\mathbf{X}_{t_k}, \ldots, \mathbf{X}_{t_{k+1}}).$$

where we use the convention of $t_0 = 0$ and $t_{k+1} = T-1$. For each investment period $(\mathbf{X}_{t_i}, \ldots, \mathbf{X}_{t_{i+1}})$, we denote by $R^\gamma(\mathbf{B}_L, \mathbf{X}^{t_{i+1}}_{t_i})$ the wealth achieved by the *best* long-term expert for that specific investing period (we might, of course, have a different best long-term expert for each investing period). Using the above notation, we can express the maximal wealth achieved by a specific transition path $\mathbf{S}_{k,T}$ by:

$$\hat{R}^\gamma(\mathbf{B}_{\mathbf{S}_{k,T}}, \mathbf{X}^{T-1}_0) \triangleq \Pi^k_{i=0} R^\gamma(\mathbf{B}_L, \mathbf{X}^{t_{i+1}}_{t_i}). \qquad (4)$$

Our player's objective is to achieve a bounded regret for any transition path[2]. In other words, we seek a strategy for the player whose portfolios will satisfy the following for some $q < 1$:

$$\mathbf{Regret}(\mathbf{B}_A, \mathcal{S}_{k,T}, T) \leq O(T^q). \qquad (5)$$

For convenience, we summarize the notations introduced in this section in Table 1.

Table 1: Notations

$\hat{\mathbf{b}}$	Portfolio $\mathbf{b}$ after prices are revealed
$\mathbf{b}^L_{i,j}$	The i-th long-term expert's portfolio on day j
$\mathbf{b}^S_j$	The short-term expert's portfolio on day j
$\mathbf{S}_{k,T}$	The transition path as a vector $(t_1, \ldots, t_k)$ of transition day indices
$\mathbf{b}^t_{\mathbf{S}_{k,T}}$	The portfolio used by a transition path $\mathbf{S}_{k,T}$ at time t
$\mathcal{S}_{k,T}$	The set of all transitions with a total of k short-term updates
$\mathcal{S}_{\cdot,T}$	The set of all the possible transitions of length T

The algorithm We now describe our player's strategy, which we call LSPE. The strategy is a combination of two online experts learning algorithms (similar to Gyorgy et al. (2012)). The first, which is termed the *outer* algorithm, combines all possible transition paths $\mathcal{S}_{\cdot,T}$ (see Equation (6)) and, for each transition path $\mathbf{S}_{k,T}$, we apply a second, *inner* expert algorithm designed to ensure that the long-term portfolios used

[2] We can expect such a regret bound to have a linear dependency on the number of investment periods (Arora et al., 2012).

Algorithm 1 LSPE

Parameters: $d > 0, \eta > 0$, $\hat{\mathbf{b}}_{-1} = 0$, $\mathbf{b}_0 = (\frac{1}{m}, ..., \frac{1}{m})$, prior $w^0_{\mathbf{S}_{k,T}} = \mathbb{P}(\mathbf{S}_{k,T})$ for any $\mathbf{S}_{k,T}$.
for $t = 0 \dots T$ **do**
Play $\mathbf{b}_t$ and gain daily wealth $\langle \mathbf{b}_t, \mathbf{X}_t \rangle \left(1 - \frac{\gamma}{2} ||\mathbf{b}_t - \hat{\mathbf{b}}_{t-1}||_1\right)$
Update the transition paths weights $w^{t+1}_{\mathbf{S}_{k,T}}$, $\forall \mathbf{S}_{k,T} \in \mathcal{S}_{\cdot,T}$:

$$w^{t+1}_{\mathbf{S}_{k,T}} = \frac{w^t_{\mathbf{S}_{k,T}} \left\langle \mathbf{b}^t_{\mathbf{S}_{k,T}}, \mathbf{X}_t \right\rangle \left(1 - \frac{\gamma}{2} ||\mathbf{b}^t_{\mathbf{S}_{k,T}} - \hat{\mathbf{b}}^{t-1}_{\mathbf{S}_{k,T}}||_1\right)}{\sum_{\mathcal{S}_{\cdot,T}} w^t_{\mathbf{S}_{j,T}} \left\langle \mathbf{b}^t_{\mathbf{S}_{j,T}}, \mathbf{X}_t \right\rangle \left(1 - \frac{\gamma}{2} ||\mathbf{b}^t_{\mathbf{S}_{j,T}} - \hat{\mathbf{b}}^{t-1}_{\mathbf{S}_{j,T}}||_1\right)} \quad (6)$$

Update the transition paths portfolios $\mathbf{b}^{t+1}_{\mathbf{S}_{k,T}}$ $\forall \mathbf{S}_{k,T} \in \mathcal{S}_{\cdot,T}$:
If $mod(t, d) = 0$: $\mathbf{b}^{t+1}_{\mathbf{S}_{k,T}} = SD_{\mathbf{S}_{k,T}}(\eta, t)$
else:
If $t + 1 \in \mathbf{S}_{k,T}$: $\mathbf{b}^{t+1}_{\mathbf{S}_{k,T}} = \mathbf{b}^S_{t+1}$ **else:** $\mathbf{b}^{t+1}_{\mathbf{S}_{k,T}} = \hat{\mathbf{b}}^t_{\mathbf{S}_{k,T}}$
Aggregate the suggested portfolios: $\mathbf{b}_{t+1} = \sum_{\mathbf{S}_{k,n}} w^{t+1}_{\mathbf{S}_{k,T}} \mathbf{b}^{t+1}_{\mathbf{S}_{k,T}}$
end for

during the investing period will be competitive with the portfolios of the best long-term expert in hindsight.

LSPE gets, as an input, a prior on the transition paths, $\mathbb{P}(\mathbf{S}_{k,T}) \triangleq \mathbb{P}_{KT}(k, T - k)$, where $\mathbb{P}_{KT}(\cdot)$ is the Krichevsky–Trofimov (KT) weighting (Krichevsky and Trofimov, 1981), which assigns $\int_0^1 (1-\theta)^a \theta^b / \pi \sqrt{\theta(1-\theta)} d\theta$, for a binary sequence of length $a + b$ with a ones and b zeros. Notice that each transition path $\mathbf{S}_{k,T}$, can be seen as a binary sequence in which each transition represents a one and the lack of a transition represents a zero, forming a binary sequence of length $T - 1$.

The outer algorithm implements a simple online weighting algorithm (Cesa-Bianchi and Lugosi, 2006). This algorithm assigns a prior probability for each transition path $\mathbf{S}_{k,T}$ and updates the weights for each path according to their performance.

The inner algorithm needs to be chosen more carefully since the calculation of the transaction costs depends on the previous portfolio chosed by LSPE. Therefore, to track an arbitrary expert's achieved wealth, we need to follow her portfolios precisely and not just to use a weighted average portfolio over all the existing experts. Moreover, we will rarely need to change our chosen expert[3]. Thus, we use the Shrinking Dartboard (SD) algorithm of Geulen et al. (2010). We chose this algorithm because it reserves a probability mass that remains with the previous expert.[4]. Many common expert tracking algorithms, such as Follow the Regularized Leader (FRL) (Cesa-Bianchi and Lugosi, 2006) and Exponentiated Gradient (EG) (Helmbold et al., 1998), do not satisfy this requirement and cannot achieve sublinear-regret in a market with transaction costs. We thus denote by $SD_{\mathbf{S}_{k,T}}(\eta, t)$ the portfolio suggeted by the SD instance of $\mathbf{S}_{k,T}$ at time t, applied with learning rate η.

Theoretical guarantee We now state and prove the performance guarantee for our strategy. Hereafter, for convenience, we assume w.l.o.g. that the log-cumulative wealth after d days is bounded by 1 (otherwise, we can scale the loss function).

Theorem 1. *Let $\{\mathbf{X}_t\}_{t=0}^{T-1}$ be an arbitrary sequence of price relative vectors. Then, for any k, T the expected regret of LSPE for any $\mathcal{S}_{k,T}$ satisfies*

$$\textbf{\textit{Regret}}(\mathbf{B}_{LSPE}, \mathcal{S}_{k,T}, T) \leq$$
$$\frac{3k+1}{2} \ln(T) + (k+1) \log l + 4\sqrt{\frac{(k+1)T \log(l)}{d}} + O(k)$$

Proof. The proof of this theorem has three parts. In the first part, we analyze the regret of each transition path. In the second part, we show that the aggregation of all transition paths yields a portfolio whose expected wealth is as high as any individual transition path. Finally, the third part analyzes the time complexity of the aggregation scheme.

Regret bound for a fixed $\mathbf{S}_{k,T}$ Fixing k and T, we analyze the wealth of a transition path $\mathbf{S}_{k,T} = (t_1, \dots, t_k)$, denoted by $R^\gamma(\mathbf{S}_{k,T}, \mathbf{X}_0^{T-1})$. Recall that the switching times divide the market sequence into $k + 1$ time segments, where, at each segment, we apply the SD algorithm on the set of the long-term experts. Therefore, using its guarantees on the segment starting

[3]For a further details on this point, we refer the reader to papers dealing with memory loss setting (Arora et al., 2012; Anava et al., 2013)

[4]Several other algorithms also satisfy this property (Kalai and Vempala, 2005; Anava et al., 2013), and they could be used as well.

at t_i and ending at t_{i+1}, we obtain

$$\mathbb{E}\left(\ln R^\gamma(\mathbf{B}_{SD}, \mathbf{X}_{t_i}^{t_{i+1}})\right) + \log(l) + 4\sqrt{\frac{(t_{i+1}-t_i)\log(l)}{d}} \geq \ln R^\gamma(\mathbf{B}_L, \mathbf{X}_{t_i}^{t_{i+1}}). \quad (7)$$

By restarting the SD algorithm at the start of each investment period, we gain two things. The first is the switching regret of Hazan and Seshadhri (2009), and the second gain is that it helps us reduce the time-complexity of LSPE, as will be discussed in the last part of the proof. Using Equation (7) over all the time segments, we get

$$\mathbb{E}\left(\ln R^\gamma(\mathbf{B}_{\mathbf{S}_{k,T}}, \mathbf{X}_0^{T-1})\right) + (k+1)\log l + 4\sum_{i=0}^{k}\sqrt{\frac{(t_{i+1}-t_i)\log(l)}{d}} \geq \sum_{i=0}^{k}\ln R^\gamma(\mathbf{B}_L, \mathbf{X}_{t_i}^{t_{i+1}}) = \ln \hat{R}^\gamma(\mathbf{B}_{\mathbf{S}_{k,T}}, \mathbf{X}_0^{T-1}).$$

The concavity of the root square function results in the following ineqiality,

$$\mathbb{E}\left(\ln R^\gamma(\mathbf{B}_{\mathbf{S}_{k,T}}, \mathbf{X}_0^{T-1})\right) + (k+1)\log l + 4\sqrt{\frac{(k+1)T\log(l)}{d}} \geq \ln \hat{R}^\gamma(\mathbf{B}_{\mathbf{S}_{k,T}}, \mathbf{X}_0^{T-1}). \quad (8)$$

Regret bound for the aggregated portfolio Since our portfolio is the aggregation of all the transition paths,

$$R^\gamma(\mathbf{B}_{\text{LSPE}}, \mathbf{X}_0^{T-1}) = \sum_{\mathbf{S}_{k,T}\in\mathcal{S}_{\cdot,T}} \mathbb{P}(\mathbf{S}_{k,T}) R^\gamma(\mathbf{B}_{\mathbf{S}_{k,T}}, \mathbf{X}_0^{T-1}),$$

and since all non-zero weights should sum up to 1, $\sum_{\mathbf{S}_{k,T}\in\mathcal{S}_{\cdot,T}} \mathbb{P}(\mathbf{S}_{k,T}) = 1$. Further, the combined wealth is as large as the wealth of any other portfolio in the mixture. We thus obtain the following relation,

$$\ln\left(R^\gamma(\mathbf{B}_{\text{LSPE}}, \mathbf{X}_0^{T-1})\right) \geq \ln\left(\mathbb{P}(\mathbf{S}_{k,T})\right) + \ln\left(R^\gamma(\mathbf{B}_{\mathbf{S}_{k,T}}, \mathbf{X}_0^{T-1})\right).$$

By taking the expectation and combining the above equation and Equation (8), we get that the expected regret satisfies

$$\mathbf{Regret}(\mathbf{B}_{\text{LSPE}}, \mathcal{S}_{k,T}, T) \leq \max_{\mathbf{S}_{k,T}} -\ln\left(\mathbb{P}(\mathbf{S}_{k,T})\right) + (k+1)\log l + 4\sqrt{\frac{(k+1)T\log(l)}{d}}. \quad (9)$$

To finalize the regret proof we need to bound the prior for each transition path. This can be done by assigning a prior, which is generated using the Krichevsky–Trofimov (KT) weighting. Given a binary sequence of length $a+b$ with a ones and b zeros, the KT weight assigned to this binary sequence is calculated by $\mathbb{P}_{KT}(a,b) = \int_0^1 (1-\theta)^a\theta^b/\pi\sqrt{\theta(1-\theta)}d\theta$. Each transition path $\mathbf{S}_{k,T}$, can be seen as a binary sequence in which each transition represents a one and the lack of a transition represents a zero, forming a binary sequence of length $T-1$. We get that $\sum_{\mathbf{S}_{k,T}\in\mathcal{S}_{\cdot,T}} \mathbb{P}(\mathbf{S}_{k,T}) = 1$ (Willems, 1996), and the following bound holds:

$$-\ln\left(\mathbb{P}(\mathbf{S}_{k,T})\right) \leq \frac{3k+1}{2}\ln(T) + O(k).$$

Combining the above with Equation (9), we get the desired result.

□

Computational complexity A naive implementation of LSPE is computationally prohibitive as it requires the maintenance and updating of the wealth associated with the 2^T different possible transition paths and, in addition, applying the SD algorithms the same number of times. Accordingly, we show an implementation whose time-complexity is linear in the number of the trading days t. We will use the weighting algorithm introduced by Willems (1996) for universal loss-less source coding. The first thing to notice is that $\mathbb{P}_{KT}(a,b)$ can be calculated recursively using the following formulas:

$$\mathbb{P}_{KT}(a+1,b) = \frac{a+0.5}{a+b+1}\mathbb{P}_{KT}(a,b),$$
$$\mathbb{P}_{KT}(a,b+1) = \frac{b+0.5}{a+b+1}\mathbb{P}_{KT}(a,b). \quad (10)$$

For convenience, we define $\hat{t}$ to be the number of times, up to time t, that the learner was given the opportunity to switch her portfolio. We next define $\mathbb{P}(\mathbf{S}_{k,T})$, using the following formula,

$$\mathbb{P}(\mathbf{S}_{k,T}) \triangleq \left(\Pi_{i=0}^{k-1}\mathbb{P}_{KT}(\hat{t}_{i+1}-\hat{t}_i+1,1)\right)\mathbb{P}_{KT}(\hat{t}_{k+1}-\hat{t}_k,0). \quad (11)$$

In Equation (11), we first assign a weight to the first transition at time t_1, which ends the first segment, and repeat this for all segments. Since, in the last segment there is no transition, we have $\mathbb{P}_{KT}(\hat{t}_{k+1}-\hat{t}_k,0)$.

The second thing to notice is that at each time instance t we maintain exactly $\hat{t}$ portfolios and that transition paths that last used the short-term expert at the same time share the same portfolio. Note that this follows also due to the restart the SD algorithm every time a transition path uses the short term expert. We, therefore, define $R^\gamma(\mathbf{B}_{\mathcal{S}_{\cdot,t}}, \mathbf{X}_0^{t-1}, s)$ to be the wealth achieved by LSPE after t trading days from investing in all the transition paths with the last switching time s and use $\hat{R}^\gamma(\mathbf{B}_{\mathcal{S}_{\cdot,t}}, \mathbf{X}_0^{t-1}, s)$ to denote the (unnormalized) weight before revealing $\mathbf{X}_{T-1}$. Note that this forms a disjoint partition of the wealth of LSPE: $R^\gamma(\mathbf{B}_{\text{LSPE}}, \mathbf{X}_0^{t-1}) = \sum_{s=1}^{\hat{t}} R^\gamma(\mathbf{B}_{\mathbf{B}_{\mathcal{S}_{\cdot,t}}}, \mathbf{X}_0^{t-1}, s)$.

It remains to show that $\hat{R}^\gamma(\mathbf{B}_{\mathcal{S}_{\cdot,t}}, \mathbf{X}_0^{t-1}, s)$ can be calculated sequentially in time-complexity linear in t. This will ensure that the aggregated portfolio of LSPE could be calculated in the same time complexity.

For $1 \leq s < \hat{t}$, the update of $\hat{R}^\gamma(\mathbf{B}_{\mathcal{S}_{\cdot,t-1}}, \mathbf{X}_0^{t-1}, s)$ simply consists of all the previous transition paths with s as the last rebalancing time, multiplied by the K-T weight of not switching at time $\hat{t}$. This weight can be calculated easily using Equations (11) and (4) since only the last term of Equations (11) should be modified. Thus, we get the following update,

$$\hat{R}^\gamma(\mathbf{B}_{\mathcal{S}_{\cdot,t}}, \mathbf{X}_0^{t-1}, s) = \frac{\hat{t}-1-s+0.5}{\hat{t}-1-s+1} R^\gamma(\mathbf{B}_{\mathcal{S}_{\cdot,t-1}}, \mathbf{X}_0^{t-2}, s).$$

The calculation in cases where $s = \hat{t}$ consists of the summation of all the possible paths multiplied by the K-T weight of switching at time $\hat{t+1}$. The relevant weights can be calculated as before. We, therefore, get the following formula:

$$\hat{R}^\gamma(\mathbf{B}_{\mathcal{S}_{\cdot,t}}, \mathbf{X}_0^{t-1}, \hat{t}) = \sum_{i=1}^{\hat{t}-1} \frac{0.5}{\hat{t}-1-i+1} R^\gamma(\mathbf{B}_{\mathcal{S}_{\cdot,t-1}}, \mathbf{X}_0^{t-2}, i).$$

This concludes the description of our implementation. The diagram in Figure 1 demonstrates the updating scheme described above.

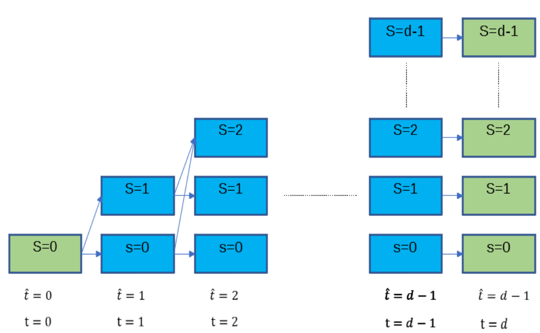

Figure 1: LSPE's update scheme: Green rectangles indicate times when the learner has to choose a long-term expert. Blue rectangles represent times when the learner is given the option to switch. S denotes the last switching time of the corresponding transition paths.

Choosing d As one might suspect, the performance of our method will depend on the choice of the parameter d. Given that the long-term experts by themselves are sensitive to this parameter choice, this fact should not be surprising. On the one hand, choosing a lower d, $d = 2$ for instance, will result in a lot of opportunities to utilize market fluctuations while increasing the transaction costs overhead; on the other hand, a larger d (say, 10) will result in a small transaction costs overhead but will reduce the possibilities to utilize market opportunities. Clearly, this choice should be market-dependent. Therefore, to enable the option of choosing the parameter, we run the SD algorithm on top of five instances of LSPE with the following choices of $d: 2, 4, 6, 8, 10$. This will give us the theoretical guarantee that this strategy will have a bounded regret from the best choice of d. We call this algorithm LSPE$_{META}$. We now state the guarantee for LSPE$_{META}$. The proof is a straightforward application of the guarantees of SD and Theorem 1.

Theorem 2. *Let $\{\mathbf{X}_t\}_{t=0}^{T-1}$ be an arbitrary sequence of price-relative vectors such that $\mathbf{X}_t \in \mathbb{R}^n$ for all t. Then, for any k, T, the expected regret of LSPE$_{META}$ for any $\mathcal{S}_{k,T}$ satisfies*

$$\textbf{\textit{Regret}}(\mathbf{B}_{LSPE_{META}}, \mathcal{S}_{k,T}, T) \leq O(k\sqrt{T})$$

5 Empirical Study

In this section we present an empirical study of LSPE$_{META}$, where we examine how well LSPE$_{META}$ can control the set of long-term experts with the opportunity to switch its decision to the short-term experts, and compare LSPE$_{META}$ to several baselines. The implementations of all the baseline algorithms are from the OLPS simulator (Li et al., 2015). Unless otherwise specified, all critical parameters of the baselines were set to the parameters recommended by their authors. The baseline algorithms are all mentioned in Section 3: (i) UCRP: Uniform CRP, which is an instance of CRP with equal weights over all the existing stocks; (ii) UP: Universal portfolios with commissions (Cover and Ordentlich, 1996); (iii) OLMAR and RMR: Two state-of-the-art commission-oblivious algorithms (Li and Hoi, 2012) (iv) SCRP: Semi-constant rebalanced portfolios introduced by Kozat and Singer (2011); (v) SUP: An improvement of SCRP (Huang et al., 2015); (vi) OLU: Utilizes gradient steps with an added ℓ_1 regularization term to encourage "lazy" portfolio updates (Das et al., 2013); (vii) CAPE: The portfolio ensemble method of (Uziel and El-Yaniv, 2016).

For the implementation of our algorithm, we chose to take several instances of OLMAR whose variants are considered state-of-the-art. The three instances that we used have different window sizes w, which were set to $(3, 4, 5)$. To fit OLMAR to our long-term transaction aware prediction, we had to modify its implementation slightly such that its predictions are

Table 2: Cumulative wealth of known commission aware algorithms and LSPE_{META}

	NYSE				TSE				DJIA			
ALGORITHM	.25%	.5%	.75%	1%	.25%	.5%	.75%	1%	.25%	.5%	.75%	1%
UCRP	28.59	25.9	23.4	21.2	1.55	1.52	1.48	1.45	0.78	0.78	0.77	0.77
UP	30.7	29.35	28.45	26.37	1.46	1.45	1.43	1.43	0.82	0.82	0.81	0.81
OLMAR	272.5	5.4	0.2	0	19.25	5.96	1.83	0.56	**1.47**	0.88	0.38	0.1
RMR	534.8	7.5	0.7	0	22.73	7.9	1.92	0.32	1.42	0.9	0.41	0
SUP	31.6	31.4	31.3	31.3	1.63	1.61	1.58	1.57	0.70	0.67	0.67	0.55
OLU	18.06	18.02	18.01	17.98	1.63	1.62	1.61	1.61	0.82	0.81	0.77	0.76
SCRP	18.94	18.8	18.65	18.6	1.61	1.61	1.60	1.60	0.78	0.77	0.77	0.76
CAPE	407	144.4	65.66	36	7.84	4.33	2.85	1.69	1.03	1.01	0.98	0.92
LSPE_{META}	**3.8E4**	**9.3E3**	**624.9**	**137.5**	**32.5**	**15.63**	**9.37**	**5.59**	1.38	**1.22**	**1.14**	**1.05**

based on the relative prices between the prediction periods $\{t \mid \mod(t, d) = 0\}$. This simple modification can straightforwardly transform any given algorithm to a long-term expert. The short-term expert, chosen to be OLMAR with $w = 5$, is aware of all the price changes and thus is able to provide short-term predictions.

Our experimental protocol followed the experimental design of Li et al. (2017), and we considered three standard datasets, NYSE, TSE and DJIA, which are used in the relevant literature and appear in the public domain. These datasets span several types of market conditions, a variety of stocks, and total trading periods (see Table 3). The most challenging dataset among the three is DJIA (the Dow Jones Industrial Average), which captures a bear market where 25 of the 30 DJIA stocks declined. We considered transaction costs γ in the set $\{.25\%, .5\%, .75\%, 1\%\}$, comprising a considerable range of transaction costs (Das et al., 2013; Li et al., 2017). As our algorithm is randomized we repeated each experiment 50 times and averaged the results. The average cumulative wealth is reported in Table 2.

In Table 2, we first examine the first two baselines, which serve as naive benchmarks. We can see that both UP and UCRP are resilient to increasing transaction costs; however, they are no match for the state-of-the-art algorithms when commissions are absent. On the other hand, both OLMAR and RMR are not suitable for settings with commissions. They exhibit nasty performance degradation when commissions are introduced. SUP, OLU and SCRP, the CRP based algorithms, are indeed an improvement over the commission-oblivious algorithms. Nevertheless, unfortunately, this improvement is only marginal.

The last block of rows in Table 2 presents the results of non-CRP-based algorithms (CAPE, LSPE), which can utilize stronger (than CRP) algorithms in their expert base. It is evident that these algorithms exhibit impressive improvements over the CRP-based methods (and the naive baselines). It is striking that LSPE is superior across almost all the datasets and across the entire range of transaction costs. Moreover, it is the only algorithm that remains profitable in a declining market (DJIA) when commission rates are larger than .75%.

In another set of experiments, we examined the contribution of the switching option. Results are presented in Table 4, where LONG represents applying the SD algorithm over the diluted experts without any option to use the short-term experts. It is evident that the addition of this option is beneficial across all datasets. We note that the benefit of using the switching option can be observed for other values of d and γ.

Table 3: Some properties of the datasets

Dataset	Starting day	# Days	# stocks
NYSE	1/1/1983	6431	23
DJIA	1/14/2001	507	30
TSE	1/4/1994	1258	88

Table 4: Long Vs. LSPE $\gamma = .5\%$

	NYSE	TSE	DJIA
Long	6.6E3	7.83	1.14
$\text{LSPE}_{d=6}$	7.9E3	10.11	1.24

6 Conclusions

Focusing on online portfolio selection with transaction costs, we presented a novel algorithm that can cleverly combine the predictions of long-term and short-term experts. We proved a regret bound for our algorithm, and presented the results of an empirical study demonstrating the superiority of the proposed algorithm over all relevant baselines. An important challenge that we wish to address would be to extend the above setting to a non-parametric framework. Another interesting direction would be to combine selective prediction techniques (Wiener and El-Yaniv, 2015) to control risky long-term predictions by preempting uncertain long-term predictions.

7 Acknowledgments

This research was supported by The Israel Science Foundation, grant No. 710/18.

References

O. Anava, E. Hazan, and S. Mannor. Online convex optimization against adversaries with memory and application to statistical arbitrage. *arXiv preprint arXiv:1302.6937*, 2013.

R. Arora, O. Dekel, and A. Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. *arXiv preprint arXiv:1206.6400*, 2012.

A. Blum and A. Kalai. Universal portfolios with and without transaction costs. *Machine Learning*, 35(3): 193–205, 1999.

A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 2005.

A. Borodin, R. El-Yaniv, and V. Gogan. Can we learn to beat the best stock? *Journal of Artificial Intelligence Research*, pages 579–594, 2004.

E. Burnaev, A. Korotin, and V. V'yugin. Long-term sequential prediction using expert advice. *arXiv preprint arXiv:1711.03194*, 2017.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

T.M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991.

T.M. Cover and E. Ordentlich. Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2):348–363, 1996.

P. Das, N. Johnson, and A. Banerjee. Online lazy updates for portfolio selection with transaction costs. In *AAAI*, 2013.

P. Das, N. Johnson, and A. Banerjee. Online portfolio selection with group sparsity. In *28th AAAI Conference on Artificial Intelligence*, 2014.

M.H.A. Davis and A.R. Norman. Portfolio selection with transaction costs. *Mathematics of Operations Research*, 15(4):676–713, 1990.

S. Geulen, B. Vöcking, and M. Winkler. Regret minimization for online buffering problems using the weighted majority algorithm. In *COLT*, pages 132–143, 2010.

L. Györfi, G. Lugosi, and F. Udina. Nonparametric kernel-based sequential investment strategies. *Mathematical Finance*, 16(2):337–357, 2006.

L. Györfi, A. Urbán, and I. Vajda. Kernel-based semi-log-optimal empirical portfolio selection strategies. *International Journal of Theoretical and Applied Finance*, 10(03):505–516, 2007.

A. Gyorgy, T. Linder, and G. Lugosi. Efficient tracking of large classes of experts. *IEEE Transactions on Information Theory*, 58(11):6709–6725, 2012.

E. Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 393–400. ACM, 2009.

D.P. Helmbold, R.E. Schapire, Y. Singer, and M.K. Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

D. Huang, J. Zhou, B. Li, S.C.H. Hoi, and S. Zhou. Robust median reversion strategy for on-line portfolio selection. In *Proceedings of the 23th International Joint Conference on Artificial Intelligence*, pages 2006–2012. AAAI Press, 2013.

D. Huang, Y. Zhu, B. Li, S. Zhou, and S.C.H. Hoi. Semi-universal portfolios with transaction costs. *Proceedings of the International Joint Conference on Artificial Intelligence*, 2015.

A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

H. Konno and A. Wijayanayake. Portfolio optimization problem under concave transaction costs and minimal transaction unit constraints. *Mathematical Programming*, 89(2):233–250, 2001.

S.S. Kozat and A.C. Singer. Universal switching portfolios under transaction costs. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5404–5407. IEEE, 2008.

S.S. Kozat and A.C. Singer. Switching strategies for sequential decision problems with multiplicative loss with application to portfolios. *IEEE Transactions on Signal Processing*, 57(6):2192–2208, 2009.

S.S. Kozat and A.C. Singer. Universal semiconstant rebalanced portfolios. *Mathematical Finance*, 21(2): 293–311, 2011.

Raphail Krichevsky and Victor Trofimov. The performance of universal encoding. *IEEE Transactions on Information Theory*, 27(2):199–207, 1981.

B. Li and S.C.H. Hoi. On-line portfolio selection with moving average reversion. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 273–280, 2012.

B. Li and S.C.H. Hoi. Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)*, 46(3):35, 2014.

B. Li, P. Zhao, S.C.H. Hoi, and V. Gopalkrishnan. Pamr: Passive aggressive mean reversion strategy for portfolio selection. *Machine Learning*, 87(2):221–258, 2012.

B. Li, D. Sahoo, and S.C.H. Hoi. Olps: A toolbox for online portfolio selection. *Journal of Machine Learning Research (JMLR)*, 2015. URL `https://github.com/OLPS`.

B. Li, J. Wang, D. Huang, and SCH Hoi. Transaction cost optimization for online portfolio selection. *Quantitative Finance*, pages 1–14, 2017.

M.S. Lobo, M. Fazel, and S. Boyd. Portfolio optimization with linear and fixed transaction costs. *Annals of Operations Research*, 152(1):341–365, 2007.

G. Uziel and R. El-Yaniv. Online learning of commission avoidant portfolio ensembles. *arXiv preprint arXiv:1605.00788*, 2016.

M. Weinberger and E. Ordentlich. On delayed prediction of individual sequences. In *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*, page 148. IEEE, 2002.

Yair Wiener and Ran El-Yaniv. Agnostic pointwise-competitive selective classification. *Journal of Artificial Intelligence Research*, 52:171–201, 2015.

F. Willems. Coding for a binary independent piecewise-identically-distributed source. *IEEE transactions on Information Theory*, 42(6):2210–2217, 1996.