Online Lazy Updates for Portfolio Selection with Transaction Costs

Puja Das, Nicholas Johnson, and **Arindam Banerjee**
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
{pdas,njohnson,banerjee}@cs.umn.edu

Abstract

A major challenge for stochastic optimization is the cost of updating model parameters especially when the number of parameters is large. Updating parameters frequently can prove to be computationally or monetarily expensive. In this paper, we introduce an efficient primal-dual based online algorithm that performs *lazy updates* to the parameter vector and show that its performance is competitive with reasonable strategies which have the benefit of hindsight. We demonstrate the effectiveness of our algorithm in the online portfolio selection domain where a trader has to pay proportional transaction costs every time his portfolio is updated. Our Online Lazy Updates (OLU) algorithm takes into account the transaction costs while computing an optimal portfolio which results in sparse updates to the portfolio vector. We successfully establish the robustness and scalability of our lazy portfolio selection algorithm with extensive theoretical and experimental results on two real-world datasets.

1 Introduction

With the ever increasing amount of data, particularly from search engines and social networks, stochastic optimization algorithms have become desirable for large-scale machine learning tasks because of their empirical efficiency and strong theoretical guarantees (Bottou 1991; 2010; Beck and Teboulle 2003; Shalev-Shwartz et al. 2009; Agarwal, Negahban, and Wainwright 2012).

However, a major challenge that is encountered is the cost of updating model parameters especially when the number of parameters can be in the order of billions. Often times when parameters are updated, their values do not change significantly. As such, the cost of updating each parameter starts to outweigh the benefit.

An important and relevant application where changing the model parameters might prove to be monetarily expensive is the domain of online portfolio selection. Here every time an investor changes his portfolio, he ends up buying or selling his stocks and incurring transaction costs. Hence, trading aggressively might sometimes hurt an investor instead of proving to be beneficial. In such a situation it might be helpful to make sparse or *lazy* updates to a portfolio.

Algorithms for automatically designing portfolios based on historical stock market data have been extensively investigated in the literature for the past five decades (Markowitz 1952; Kelly 1956; Sharpe 1964). With the realization that any statistical assumptions regarding the stock market may be inappropriate and eventually counter-productive, over the past two decades, new methods for portfolio selection have been designed which make no statistical assumptions regarding the movement of stocks (Cover 1991; Cover and Ordentlich 1996; Helmbold et al. 1998). In a well-defined technical sense, such methods are guaranteed to perform competitively with certain families of adaptive portfolios even in an adversarial market. From the theoretical perspective, algorithm design for portfolio selection has largely been a success story (Cover 1991; Helmbold et al. 1998; Cesa-Bianchi and Lugosi 2006).

Although theoretical and empirical performance of such online portfolio selection algorithms have been encouraging, they have ignored one crucial practical aspect of financial trading: transaction costs. These online algorithms (Cover 1991; Helmbold et al. 1998; Cesa-Bianchi and Lugosi 2006; Agarwal et al. 2006; Borodin, El-Yaniv, and Gogan 2004) could be trading aggressively and a major concern is the cost they would incur in a real world scenario.

In this paper, we introduce an online portfolio selection algorithm *with transaction costs*. The algorithm is penalized by a fixed percentage of the amount of transactions it makes on a per day basis. We pose this as a non-smooth online convex optimization problem and propose an efficient algorithm called Online Lazy Updates (OLU) to make lazy updates to our online portfolio. Furthermore, we go on to prove that our lazy portfolio is competitive with reasonable strategies which have the benefit of hindsight. We conduct extensive experiments on two real world datasets: 22 years of the benchmark NYSE dataset with 36 stocks and 20 years of a S&P500 dataset with 263 stocks. Our experiments show that our lazy portfolios are scalable with transaction costs and, interestingly, in some cases, can outperform their non-lazy counterparts in terms of wealth achieved. Our algorithm is especially beneficial for individual investors who are typically affected by transaction costs.

We arrange the rest of the paper as follows. We formally describe the online portfolio selection framework in a costless environment in Section 2. In Section 3, we describe our framework with transaction costs, discuss related work, propose our Online Lazy Updates (OLU) algorithm, and outline its analysis. Section 4 contains details of our experiments and their results. We conclude with directions of our future work in Section 5.

2 Online Portfolio Selection

We consider a stock market consisting of n stocks $\{s_1, \ldots, s_n\}$ over a span of T periods. For ease of exposition, we will consider a period to be a day, but the analysis presented holds for any valid definition of a 'period' such as an hour or a month. Let $x_t(i)$ denote the *price relative* of stock s_i in day t, i.e., the multiplicative factor by which the price of s_i changes in day t. Hence, $x_t(i) > 1$ implies a gain, $x_t(i) < 1$ implies a loss, and $x_t(i) = 1$ implies the price remained unchanged. We assume, $x_t(i) > 0 \, \forall \, i, t$. Let $\mathbf{x}_t = \langle x_t(1), \ldots, x_t(n) \rangle$ denote the vector of price relatives for day t, and let $\mathbf{x}_{1:t}$ denote the collection of such price relative vectors up to and including day t. A portfolio $\mathbf{p}_t = \langle p_t(1), \ldots, p_t(n) \rangle$ on day t can be viewed as a probability distribution over the stocks that prescribes investing $p_t(i)$ fraction of the current wealth in stock s_i. Note that the portfolio $\mathbf{p}_t$ has to be decided before knowing $\mathbf{x}_t$ which will be revealed only at the end of the day. The multiplicative gain in wealth at the end of day t, is then simply $\mathbf{p}_t^T \mathbf{x}_t = \sum_{i=1}^n p_t(i) x_t(i)$. For a sequence of price relatives $\mathbf{x}_{1:t-1} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}\}$ up to day $(t-1)$, the sequential portfolio selection problem in day t is to determine a portfolio $\mathbf{p}_t$ based on past performance of the stocks. At the end of day t, $\mathbf{x}_t$ is revealed and the actual performance of $\mathbf{p}_t$ gets determined by $\mathbf{p}_t^T \mathbf{x}_t$. Over T periods, for a sequence of portfolios $\mathbf{p}_{1:T} = \{\mathbf{p}_1, \ldots, \mathbf{p}_T\}$, the multiplicative gain in wealth and the logarithmic gain in wealth is then,

$$S_T(\mathbf{p}_{1:T}, \mathbf{x}_{1:T}) = \prod_{t=1}^{T} \left(\mathbf{p}_t^T \mathbf{x}_t\right) \tag{1}$$

$$LS_T(\mathbf{p}_{1:T}, \mathbf{x}_{1:T}) = \sum_{t=1}^{T} \log \left(\mathbf{p}_t^T \mathbf{x}_t\right) \tag{2}$$

respectively. Ideally, for a costless environment (no transaction costs) we would like to maximize $LS_T(\mathbf{p}_{1:T}, \mathbf{x}_{1:T})$ over $\mathbf{p}_{1:T}$. However, online portfolio selection cannot be posed as an optimization problem due to the temporal nature of the choices: $\mathbf{x}_t$ is not available when one has to decide on $\mathbf{p}_t$. Further, in a stock market, (statistical) assumptions regarding $\mathbf{x}_t$ can be difficult to make.

3 Online Portfolio with Transaction Costs

Typically, there can be two types of transaction costs in real markets: (1) a *fixed percentage* of each transaction that the investor has to pay to a broker or (2) a *fixed amount* paid per transaction (sell or buy). In this work we look at costs of the first type also known as *proportional transaction costs* in financial modeling (Davis and Norman 1990; Magill and Constantinides 1976). To fully specify our model for online portfolio selection with transaction costs, we proceed by discussing related work, our problem formulation, followed by our Online Lazy Updates (OLU) algorithm and its analysis.

3.1 Related Work

The need for considering transaction costs in the design and analysis of online portfolio selection algorithms has been raised in (Helmbold et al. 1998; Cover and Ordentlich 1996; Ordentlich and Cover 1996; Cesa-Bianchi and Lugosi 2006; Agarwal et al. 2006; Borodin, El-Yaniv, and Gogan 2004). So far, only (Blum and Kalai 1997) have extended the analysis of (Cover 1991) to include proportional transaction costs. Their strategy involved first computing a target portfolio using (Cover 1991) and then paying for the transactions proportionally from each stock. Their analysis shows that the performance guarantee of the Universal Portfolio still holds (and gracefully degrades) in the case of proportional commissions. However, (Cover 1991) is computationally demanding and has been shown to have sobering empirical performance (Helmbold et al. 1998; Das and Banerjee 2011). (Blum and Kalai 1997) and heuristics like Anticor (Borodin, El-Yaniv, and Gogan 2004) and OLMAR (Li and Hoi 2012) do not account for transaction costs in their algorithm design. Anticor and OLMAR rely on empirical results to show scalability of their strategies to small transaction costs only as a post-processing step.

3.2 Problem Formulation

We present a general formulation for our online lazy updates problem and go on to show how portfolio selection with transaction costs is a special case of this setting. In an online lazy updates setting the optimization proceeds in rounds where in round t the algorithm has to pick a solution, $\theta_t \in F$, from the feasible set such that it is close to the previous solution θ_{t-1}. Nature then reveals the convex loss function, ϕ_t, and we observe its value $\phi_t(\theta_t)$. Ideally, over T rounds we would like to minimize the quantity,

$$\sum_{t=1}^{T} \phi_t(\theta_t) + \gamma \sum_{t=2}^{T} ||\theta_t - \theta_{t-1}||_1 \tag{3}$$

The ℓ_1 penalty term ensures that the updates to the solution θ_t are lazy. Absolute minimization of (3) is not reasonable because we do not know the sequence of ϕ_t *a priori*. If the ϕ_ts are known, (3) reduces to a batch optimization problem: a special case is the fused lasso when ϕ_t is quadratic (Tibshirani et al. 2005). Alternatively, over T iterations we intend to get a sequence of θ_t such that the following *regret bound* is sublinear in T,

$$R_T = \sum_{t=1}^{T} f_t(\theta_t) - \min_{\theta^*} \sum_{t=1}^{T} f_t(\theta^*) \leq o(T) \tag{4}$$

where $f_t(\theta) = \phi_t(\theta) + \gamma||\theta - \theta_{t-1}||_1$ is non-smooth. θ^* is the minimizer of $\sum_{t=1}^{T} f_t$ in hindsight. Note that while the θ_ts can change over time, θ^* is fixed. That is, the minimizer,

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{t=1}^{T} f_t(\theta) = \underset{\theta}{\operatorname{argmin}} \sum_{t=1}^{T} \phi_t(\theta), \tag{5}$$

Algorithm 1 Online Lazy Update (OLU) with ADMM

```
1: Input $\mathbf{p}_t, \mathbf{x}_t, \eta, \alpha, \beta$
2: Initialize $\mathbf{p}, z, u \in 0^n, k = 0$
3: ADMM iterations
   $\mathbf{p}^{(k+1)} = \prod_{\triangle_n}\Big\{- \frac{\eta \mathbf{x}_t}{(\beta+1)\mathbf{p}_t^T\mathbf{x}_t} + \mathbf{p}_t + \frac{\beta z^{(k)}}{(\beta+1)} - \frac{\beta u^{(k)}}{(\beta+1)}\Big\}$
   $z^{(k+1)} = S_{\alpha/\beta}(\mathbf{p}^{(k+1)} - \mathbf{p}_t + u^{(k)})$
   $u^{(k+1)} = u^{(k)} + (\mathbf{p}^{(k+1)} - \mathbf{p}_t - z^{(k+1)})$ .
   where $\prod_{\triangle_n}$ is a projection to the simplex and $S_\rho$ is the shrinkage operator.
4: Continue until **Stopping Criteria** is satisfied
```

since it incurs zero ℓ_1 penalty in every iteration. Online portfolio selection with transaction costs can now be viewed as a special case of our online lazy updates setting where $f_t(\mathbf{p}) = -\log(\mathbf{p}^T\mathbf{x}_t) + \gamma||\mathbf{p} - \mathbf{p}_{t-1}||_1$. The ℓ_1 penalty term on the difference of two consecutive portfolios measures the fraction of wealth traded. The parameter γ controls the amount that can be traded every day. Note that on setting $\gamma = 0$, our formulation reduces to the costless case as seen in (2).

3.3 Online Lazy Update (OLU) Algorithm

We now formulate an online lazy portfolio selection strategy that allows us to control the total amount of transaction everyday. It decides to trade or not depending on if the benefits of changing the portfolio outweigh the transaction costs. We find a new lazy portfolio vector $\mathbf{p}_{t+1}$ as follows: $\mathbf{p}_{t+1} = \underset{\mathbf{p}\in\triangle_n}{\text{argmin}} \; -\log(\mathbf{p}^T\mathbf{x}_t) + \gamma||\mathbf{p} - \mathbf{p}_t||_1 + \frac{1}{2\eta}\,||\mathbf{p} - \mathbf{p}_t||_2^2$. This can be rewritten as,

$$\mathbf{p}_{t+1} = \underset{\mathbf{p}\in\triangle_n}{\text{argmin}} \; -\eta\log(\mathbf{p}^T\mathbf{x}_t) + \alpha||\mathbf{p} - \mathbf{p}_t||_1 + \frac{1}{2}\,||\mathbf{p} - \mathbf{p}_t||_2^2, \tag{6}$$

which we will use from here on (where $\alpha = \eta * \gamma$). In this online framework, the new portfolio vector is computed as a function of $\mathbf{p}_t$ and the price relatives $\mathbf{x}_t$ and lies in the probability simplex in n-dimension. The purpose of the first term is to maximize the logarithmic wealth if the current price relative $\mathbf{x}_t$ is replicated. The second term is the ℓ_1 penalty which accounts for the amount of transaction that would take place to update to a new portfolio. The parameter $\alpha > 0$ decides how often we trade; high values of α lead to *lazy* updates of the portfolio with small amount of transactions while low values allow the portfolio to change more often. Our framework for updating a portfolio vector is analogous to the framework of the EG algorithm (Helmbold et al. 1998). We use $||\cdot||_{\ell_2}^2$ as the distance function instead of the relative entropy in EG. Unlike EG, we solve a non-smooth problem.

We propose an ADMM (Alternating Direction Method of Multipliers (Boyd et al. 2011)) based efficient primal-dual algorithm to obtain the lazy portfolio $\mathbf{p}_{t+1}$ by solving (6). ADMM is an efficient distributed optimization method closely related to Bregman iterative algorithms for l_1 problems and proximal point methods. It has been applied in many large scale problems in statistics and machine learning because of its computational benefits and fast convergence in practice (Boyd et al. 2011). We can rewrite (6) in the ADMM form by introducing an auxiliary variable z as,

$$\underset{\mathbf{p}\in\triangle_n,\mathbf{p}-\mathbf{p}_t=z}{\text{argmin}} \; -\eta\log(\mathbf{p}^T\mathbf{x}_t) + \alpha||z||_1 + \frac{1}{2}||\mathbf{p}-\mathbf{p}_t||_2^2 \tag{7}$$

This ADMM formulation naturally lets us decouple the non-smooth l_1 term from the smooth terms, which is computationally advantageous.We replace the log term in (7) by its first order Taylor expansion around $\mathbf{p}_t$. The *augmented Lagrangian* for the above problem is then,

$$\begin{aligned} L_\beta(\mathbf{p}, z, u) = \underset{\mathbf{p}\in\triangle_n}{\text{argmin}} \; -\eta\left(\log(\mathbf{p}_t^T\mathbf{x}_t) + \frac{\mathbf{x}_t^T(\mathbf{p}-\mathbf{p}_t)}{\mathbf{p}_t^T\mathbf{x}_t}\right) \\ + \alpha||z||_1 + \frac{1}{2}||\mathbf{p}-\mathbf{p}_t||_2^2 + \frac{\beta}{2}||\mathbf{p}-\mathbf{p}_t - z + u||_2^2 \end{aligned} \tag{8}$$

where $u = \frac{1}{\beta}\lambda$ is the scaled dual variable and λ is the dual variable. ADMM consists of the following iterations for solving $\mathbf{p}_{t+1}$,

$$\begin{aligned} \mathbf{p}_{t+1}^{(k+1)} = \underset{\mathbf{p}\in\triangle_n}{\text{argmin}} \; -\eta\left(\log(\mathbf{p}_t^T\mathbf{x}_t) + \frac{\mathbf{x}_t^T(\mathbf{p}-\mathbf{p}_t)}{\mathbf{p}_t^T\mathbf{x}_t}\right) \\ + \frac{1}{2}||\mathbf{p}-\mathbf{p}_t||_2^2 + \frac{\beta}{2}||\mathbf{p}-\mathbf{p}_t - z^{(k)} + u^{(k)}||_2^2 \end{aligned} \tag{9}$$

$$z^{(k+1)} = \underset{z}{\text{argmin}} \; \alpha||z||_1 + \frac{\beta}{2}||\mathbf{p}_{t+1}^{(k+1)} - \mathbf{p}_t - z + u^{(k)}||_2^2 \tag{10}$$

$$u^{(k+1)} = u^{(k)} + (\mathbf{p}_{t+1}^{(k+1)} - \mathbf{p}_t - z^{(k+1)}) \tag{11}$$

Algorithm 1 shows the closed form updates derived for $\mathbf{p}_{t+1}^{k+1}$, z^{k+1}, and u^{k+1}. The update for $\mathbf{p}_{t+1}^{k+1}$ is derived by taking the derivative of (9) and setting it to zero. The projection to the simplex ($\prod_{\triangle_n}$) is carried out as in (Duchi, Singer, and Chandra 2008). The stopping criteria for the OLU algorithm is based on the primal and dual residuals from (Boyd et al. 2011).

Algorithm 2 is our online portfolio selection algorithm with transaction costs. It uses the OLU Algorithm to compute the lazy updates to the portfolio $\mathbf{p}_{t+1}$. It takes in an additional parameter γ which is a fixed percentage charged for the total amount of transaction everyday. S_t^γ is the transaction cost-adjusted cumulative wealth gain at the end of t days.

3.4 Analysis

We sequentially invest with the lazy portfolios $\mathbf{p}_1, \cdots, \mathbf{p}_T$ obtained from Algorithm 1 and on day t suffer a loss $f_t(\mathbf{p}_t) = \phi_t + \gamma||\mathbf{p}_t - \mathbf{p}_{t-1}||_1$, where $\phi_t = -\log(\mathbf{p}_t^T\mathbf{x}_t)$. Our goal is to minimize the *regret* with respect to the best fixed (non-shifting) portfolio $\mathbf{q}^*$ in hindsight. We establish the standard regret bound in portfolio selection literature (Cover 1991; Helmbold et al. 1998; Agarwal et al. 2006) using Theorem 1 :

Algorithm 2 Portfolio Selection with Transaction costs

```
1: Input $\eta, \gamma, \beta$; Compute $\alpha = \eta\gamma$
2: Initialize $p_{1,h} = \frac{1}{n}, h = 1, \dots, n; p_0 = p_1; S_0^\gamma = 1$
3: For $t = 1, \dots, T$
4:     Receive $\mathbf{x}_t$ vector of price relatives
5:     Compute cumulative wealth: $S_t^\gamma = S_{t-1}^\gamma \times (\mathbf{p}_t^T \mathbf{x}_t) - \gamma \times S_{t-1}^\gamma \times ||\mathbf{p}_t - \mathbf{p}_{t-1}||_1$
6:     Update portfolio: $\mathbf{p}_{t+1} = \text{OLU}(\mathbf{p}_t, \mathbf{x}_t, \eta, \alpha, \beta)$
7: end for
```

Theorem 1 *Let $\mathbf{q}^* \in \triangle_n$ be the fixed portfolio obtained from $\min_{\mathbf{q}} \sum_{t=1}^T \phi_t(\mathbf{q})$. For $\eta = \sqrt{T}$ and $\|\nabla\phi_t(\mathbf{p}_t)\| \leq G$, the regret can be bounded as,*

$$\sum_{t=1}^T \phi_t(\mathbf{p}_t) + \gamma \sum_{t=2}^T \|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1 - \sum_{t=1}^T \phi_t(\mathbf{q}^*) \leq O(\sqrt{T}), \tag{12}$$

where ϕ_t is a strongly convex function and the sequence $\mathbf{p}_t$ and the fixed optimal portfolio $\mathbf{q}^*$ all lie in the probability simplex in n-dimensions.

The complete proof of the above Theorem and related results will be available in a longer version of the paper.

4 Experiments and Results

4.1 Datasets

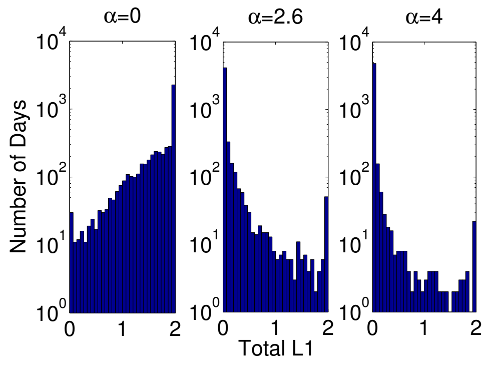
The experiments were conducted on two real-world datasets: the New York Stock Exchange (NYSE) (Cover 1991) and the Standard & Poor's 500 (S&P 500) (Das and Banerjee 2011) datasets. The NYSE dataset consists of 36 stocks with data accumulated over a period of 22 years from July 3, 1962 to December 31, 1984. The dataset captures the bear market that lasted between January 1973 and December 1974. However, all of the 36 stocks increase in value in the 22-year run. This is a benchmark dataset that has been used extensively in the online portfolio selection literature for demonstration of empirical results (Helmbold et al. 1998; Agarwal et al. 2006; Borodin, El-Yaniv, and Gogan 2004; Cover 1991). The S&P500 dataset consists of 263 stocks which were present in the S&P500 index in 2010 and were alive since 1990. This period of 20 years from 1990 to 2010 covers the bull and bear markets of recent times such as the dot-com bubble which occurred between 1997-2000, the following bubble burst during March 2000, and the recent housing bubble burst occurring between 2006-2007.

4.2 Methodology and Parameter Setting

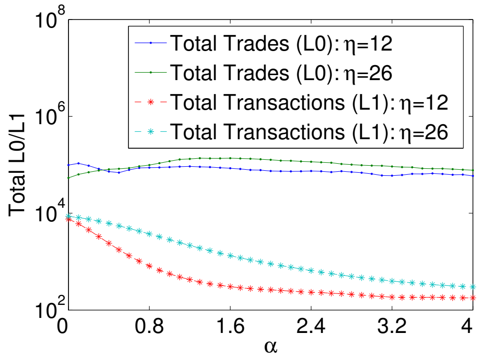
In all our experiments we start with $1 as our initial investment and an initial portfolio which is uniformly distributed over all the stocks. We use Algorithm 2 to obtain our portfolios sequentially and compute the transaction cost-adjusted wealth for each day. The parameters consist of $\boldsymbol{\eta}$: weight on logarithmic gain in wealth, $\boldsymbol{\gamma}$: fixed percentage transaction cost, and $\boldsymbol{\beta}$: the parameter for the augmentation term of the OLU algorithm. For all our experiments, we set $\beta = 0.1$ which we found to give reasonable accuracy.

Since the two datasets are very different in nature (stock composition and duration), we experimented extensively with a large range of η and α values to observe their effect on the lazy updates of our portfolio. Moreover, we chose a reasonable range of γ values (in percentage) to compute the proportional transaction costs incurred due to the portfolio update every day. The range of γ values we experimented with were between 0% and 2%. We have illustrated some of our results with representative plots from either the NYSE or S&P500 dataset.

We use the wealth obtained (without transaction costs) EG algorithm and a Buy-and-Hold strategy as benchmarks for our experiments. EG has been shown to outperform a uniform constantly rebalanced portfolio (Helmbold et al. 1998; Das and Banerjee 2011). For the Buy-and-Hold case we start with a uniformly distributed portfolio and do a hold on the positions thereafter (i.e. no trades). We also use the S&P500 Index as a representative index for the US stock market to analyze the activity of our lazy update algorithm. We do not compare our method with Anticor or OLMAR because these heuristics do not account for transaction costs in their algorithmic framework and have no theoretical guarantees.

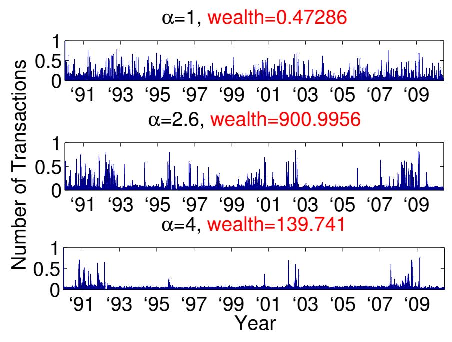


(a) Histogram of ℓ_1 values for the S&P500 dataset.

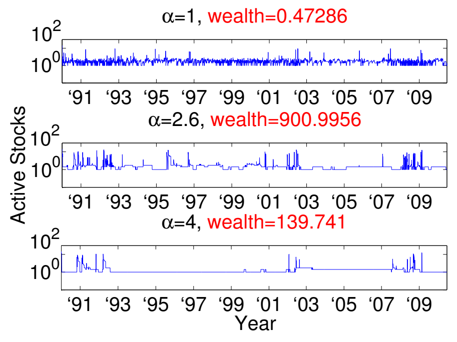


(b) Number of trades (ℓ_0 norm) and amount of trade (ℓ_1 norm) for varying α (S&P500 dataset).

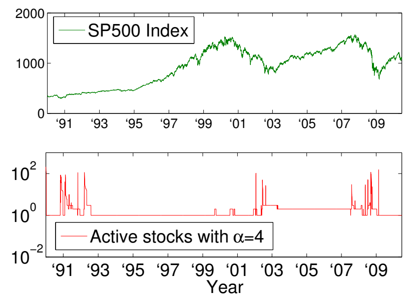
Figure 1: Effect of α: as α increases, the total amount of the transaction decreases but the total number of trades may not decrease monotonically.



(a) Fraction of transactions per day.



(b) Stocks which comprise 80% of the wealth.



(c) S&P 500 Index value vs. Active stocks.

Figure 2: As α increases, there is a decline in the number of transactions and OLU tends to hold on to stocks interspersed with days of high activity (transactions). Days of high stock activity coincides with major movements in the market.

4.3 Effect of α and the ℓ_1 penalty

The parameter α is the weight on the ℓ_1 penalty term and can influence (a) the total amount of transactions, (b) the total amount of trades and transactions, (c) the daily amount of transactions, and (d) the stock activity. We now investigate the effect of α in the two datasets.

(a) Total Amount of Transactions: Let $\Upsilon_t = ||\mathbf{p}_{t+1} - \mathbf{p}_t||_1$, then $\sum_{t=1}^{T-1} \Upsilon_t$ is a measure of the total amount that a trader had to pay in transaction costs over T days (as a fraction of his wealth). Figure 1(a) plots a histogram of Υ_t for varying α values for S&P500 dataset. We observe that as α increases, the Υ_t value is small for most days. With $\alpha = 0$, Υ_t was 2 for most days denoting non-lazy portfolios which is how portfolios are expected to trade in a costless environment. The plot for the NYSE dataset shows similar trends.

(b) Total Amount of Trades and Transactions: We now analyze the behavior of the total number of trades (ℓ_0 norm) and the total amount of transactions ($\sum_{t=1}^{T-1} \Upsilon_t$, ℓ_1 norm) as we increase the value of α. Figure 1(b) gives a holistic overview of the behavior of the aforementioned quantities as we increase α. Figure 1(b) confirms that the total ℓ_1 norm decreases as we increase α. The total ℓ_0 norm, however, does not always decrease as we increase α. Figure 1(b) shows such a situation for the S&P500 dataset.

(c) Daily Amount of Transactions: Figure 2(a) plots the fraction of stocks traded per day for the S&P500 dataset for three values of α. We observe that as we increase the weight on the ℓ_1 penalty term by tuning our parameter α, the number of transactions decreases. Whereas a large amount of the 263 stocks were traded everyday for $\alpha = 0$, with higher values of α the number of transactions reduces significantly. We observe a similar trend for the NYSE dataset.

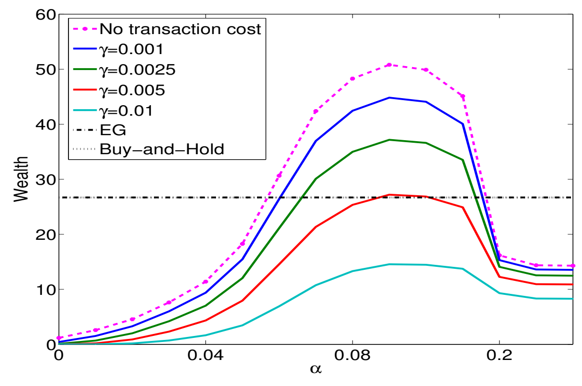
(d) Active Stocks: Figure 2(b) plots the number of stocks which comprise 80% of the total wealth on a per day basis which we call the *active* stocks. As α increases, the lazy behavior of the portfolios becomes more apparent. We observe that high weight on the ℓ_1 penalty term forces the online portfolios to change their composition only on a handful of days.

Correlation with the US market: In Figure 2(b), we observe significant activity between years 2002-2003 and between years 2008-2009. On plotting the value of the S&P500 index for the US market between 1990 and 2010 in Figure 2(c), we realized that the increase in trading activity reflected two major market movements: the dot-com and housing bubble bursts. Figure 2(c) shows that the days of high stock activity coincides with major market movements. Similar trends were observed for the NYSE dataset.

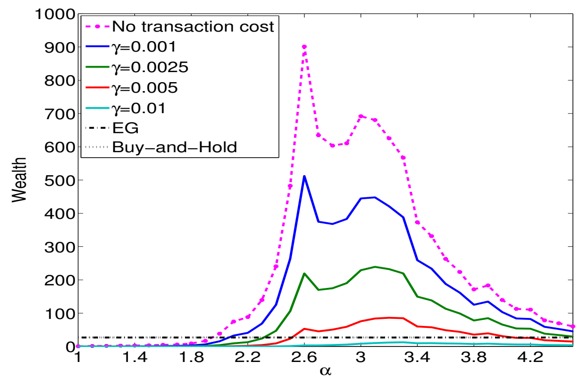
4.4 Wealth with Transaction Costs (S_T^γ)

To evaluate the practical application of our proposed algorithm, we now analyze its performance when calculating the transaction cost-adjusted cumulative wealth. Figure 3 shows how the choice of different α values affect the transaction cost-adjusted cumulative wealth for the two datasets (for a fixed η value). Figures 3(a) and 3(b) demonstrate that there exists a regime of $\alpha = \eta\gamma$ which makes an optimum choice between exploration and exploitation of stocks. Since γ can be fixed, the learning rate η can be adequately chosen to maximize our wealth. Very low values of η tends to aggressively change the portfolio too often. Whereas, with very high values of η, the algorithm becomes too conservative and might not be able to take advantage of short trends in the market. For a fixed α, decreasing η makes the portfolios lazy (higher γ) and increasing η (lower γ) encourages trading.

EG and Buy-and-Hold: We compared the total wealth without transaction costs of OLU with that of EG and a Buy-and-Hold strategy. EG and Buy-and-Hold are plotted as horizontal lines and we can see that for the NYSE dataset EG returns \$26.70 and Buy-and-Hold returns \$26.78. For the S&P500 dataset EG returns \$26.68 and Buy-and-Hold returns \$27.25. In comparison, OLU returns \$50.80 and \$901.00 respectively without transaction costs. OLU returns almost 2x as much wealth for the NYSE dataset and 33x as much wealth for the S&P500 dataset as EG or Buy-and-Hold do. Figures 3(a) and 3(b) also show that OLU is able to return more wealth than EG and Buy-and-Hold with reasonable transaction costs (0.1%, 0.25%, and 0.5%).

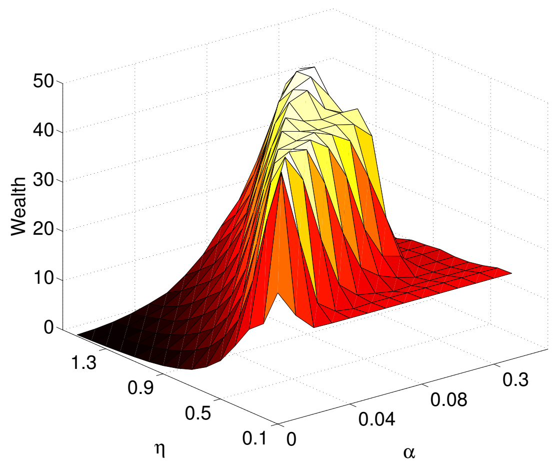


(a) Transaction cost-adjusted wealth: S_T^γ for NYSE dataset.

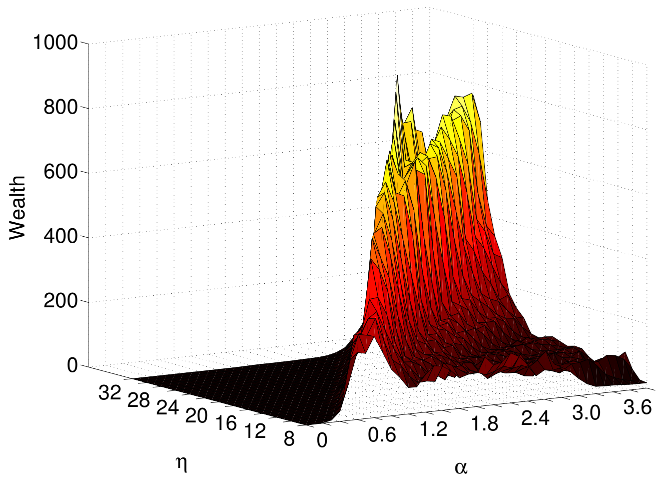


(b) Transaction cost-adjusted wealth: S_T^γ for S&P500 dataset

Figure 3: OLU with transaction costs can outperform (in terms of wealth) EG and Buy-n-Hold (without transaction costs).



(a) S_T^γ for NYSE dataset.



(b) S_T^γ for S&P500 dataset.

Figure 4: Transaction cost-adjusted wealth: S_T^γ as a function of η and α for NYSE and S&P500 datasets.

4.5 Parameter Sensitivity (η and α)

Figure 4 gives us more insight into how the transaction cost-adjusted wealth behaves as a function of $\frac{\eta}{\alpha} = \frac{1}{\gamma}$ for the two datasets. We can see that the cumulative wealth looks like a hill or ridge and that on either sides of the ridge the wealth is small. This particularly occurs when either η or α are too high or too low. Only when both η and α are in relative balance are we able to obtain significant cumulative wealth.

5 Conclusion and Future Work

In this paper, we have developed a framework and an online algorithm (OLU) to allow for lazy updates for the problem of portfolio selection with transaction costs. Our analysis shows that OLU is competitive with reasonable fixed strategies which have the power of hindsight. Our experimental results describe the behavior of such lazy updates and show that OLU is able to outperform EG and Buy-and-Hold even with reasonable transaction costs.

In the future we wish to explore the possibility of incorporating transactions costs and extending our analysis to other portfolio selection algorithms such as ONS (Agarwal et al. 2006) and meta-optimization algorithms (Das and Banerjee 2011) which have been shown to be theoretically grounded and empirically competitive. Moreover, we think that it is possible to generalize our theoretical bound to the shifting case, where the comparator class can also change over time, albeit lazily. Finally, we think that our lazy updates framework has the potential to be generalized to help solve complex problems in other domains such as climate sciences, image processing, and social media analytics.

Acknowledgements: The research was supported by NSF CAREER award IIS-0953274, and NSF grants IIS-0916750, IIS-0812183, and IIS-1029711, and NASA grant NNX12AQ39A. The authors also wish to thank Maria Gini for her help.

References

Agarwal, A.; Hazan, E.; Kale, S.; and Schapire, R. 2006. Algorithms for portfolio management based on the newton method. In *Proceedings of the 23rd International Conference on Machine Learning*, 9–16.

Agarwal, A.; Negahban, S.; and Wainwright, M. 2012. Stochastic optimization and sparse statistical recovery: Optimal algorithms for high dimensions. In *NIPS*, 1547–1555.

Beck, A., and Teboulle, M. 2003. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters* 31(3):167–175.

Blum, A., and Kalai, A. 1997. Universal portfolios with and without transaction costs. In *Proceedings of the 10th Annual Conference on Learning Theory*.

Borodin, A.; El-Yaniv, R.; and Gogan, V. 2004. Can we learn to beat the best stock. *Journal of Artificial Intelligence Research* 21:579–594.

Bottou, L. 1991. Stochastic gradient learning in neural networks. In *Proceedings of Neuro-Nîmes*.

Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the 19th International Conference on Computational Statistics*, 177–187.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3(1):1–122.

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

Cover, T., and Ordentlich, E. 1996. Universal portfolios with side information. *IEEE Transactions of Information Theory* 42:348–363.

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

Das, P., and Banerjee, A. 2011. Meta optimization and its application to portfolio selection. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

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

Duchi, J.; Singer, Y.; and Chandra, T. 2008. Efficient projections onto the L1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*.

Helmbold, D.; Scahpire, E.; Singer, Y.; and Warmuth, M. 1998. Online portfolio selection using multiplicative weights. *Mathematical Finance* 8(4):325–347.

Kelly, J. L. 1956. A new interpretation of information rate. *Bell Systems Technical Journal* 35:917–926.

Li, B., and Hoi, S. 2012. On-line portfolio selection with moving average reversion. In *International Conference of Machine Learning*.

Magill, M., and Constantinides, G. 1976. Portfolio selection with transaction costs. *Journal of Economic Theory* 13(2):245–263.

Markowitz, H. 1952. Portfolio selection. *Journal of Finance* 7:77–91.

Ordentlich, E., and Cover, T. 1996. Online portfolio selection. In *Proceedings of the 9th Annual Conference on Learning Theory*.

Shalev-Shwartz, S.; Shamir, O.; Srebro, N.; and Sridharan, K. 2009. Stochastic convex optimization. In *Proceedings of the Conference on Learning Theory (COLT)*.

Sharpe, W. 1964. A theory of market equilibrium. *Journal of Finance* 19:425–442.

Tibshirani, R.; Saunders, M.; Rosset, S.; Zhu, J.; and Knight, K. 2005. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B* 91–108.