Algorithms for Portfolio Management based on the Newton Method

Amit Agarwal AAGARWAL@CS.PRINCETON.EDU
Elad Hazan EHAZAN@CS.PRINCETON.EDU
Satyen Kale SATYEN@CS.PRINCETON.EDU
Robert E. Schapire SCHAPIRE@CS.PRINCETON.EDU
Princeton University, Department of Computer Science, 35 Olden Street, Princeton, NJ 08540

Abstract

We experimentally study on-line investment algorithms first proposed by Agarwal and Hazan and extended by Hazan et al. which achieve almost the same wealth as the best constant-rebalanced portfolio determined in hindsight. These algorithms are the first to combine optimal logarithmic regret bounds with efficient deterministic computability. They are based on the Newton method for offline optimization which, unlike previous approaches, exploits second order information. After analyzing the algorithm using the potential function introduced by Agarwal and Hazan, we present extensive experiments on actual financial data. These experiments confirm the theoretical advantage of our algorithms, which yield higher returns and run considerably faster than previous algorithms with optimal regret. Additionally, we perform financial analysis using mean-variance calculations and the Sharpe ratio.

1. Introduction

In the universal portfolio management problem, we seek online wealth investment strategies which enable an investor to maximize his wealth by distributing it on a set of available financial instruments without knowing the market outcome in advance. The underlying model of the problem makes no statistical assumptions on the behavior of the market (such as random walks or Brownian motion of stock prices (Luenberger, 1998)). In fact, the market is even allowed to be adversarial. The simplicity of the model permits the formulation of the centuries-old problem of wealth maximization as an online learning problem, and the application of machine learning algorithms. The study of such a model was started in the 1950s by Kelly (1956) followed by Bell and Cover (1980; 1988), Algoet and Cover (1988).

Absolute wealth maximization in an adversarial market is of course a hopeless task; we therefore aim to maximize our wealth relative to that achieved by a reasonably sophisticated investment strategy, the constant-rebalanced portfolio (Cover, 1991), abbreviated CRP. A CRP strategy rebalances the wealth each trading period to have a fixed proportion in every stock in the portfolio. We measure the performance of an online investment strategy by its *regret*, which is the relative difference between the logarithmic growth ratio it achieves over the entire trading period, and that achieved by a prescient investor — one who knows all the market outcomes in advance, but who is constrained to use a CRP. An investment strategy is said to be *universal* if it achieves sublinear regret.

Of equal importance is the computational efficiency of the online algorithm. So far, universal portfolio management algorithms were either optimal with respect to regret, but computationally inefficient (Cover, 1991), or efficient but attained sub-optimal regret (Helmbold et al., 1998). In recent work, Agarwal and Hazan (2005) introduced a new analysis technique which serves as the basis of algorithms that are both efficient and have optimal regret. These techniques were generalized by Hazan et al. (2006) to yield even more efficient algorithms.

These new algorithms are based on the well-studied *follow-the-leader* method, a natural online strategy which, simply stated, advocates the use of the best strategy so far in the game for the next iteration. This method was first proposed and analyzed by Hannan (1957) for the case of Lipschitz regret functions, and later simplified and extended by Kalai and Vempala (2005) for linear regret functions and also by Merhav

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006. Copyright 2006 by the author(s)/owner(s).

and Feder (1992). Follow-the-leader based portfolio management schemes were also analyzed by Larson (1986), when the price relative vectors are restricted to take values in a finite set and by Ordentlich and Cover (1996). The work in this paper and in Hazan et al. (2006) bring out the connection of follow-the-leader to the Newton method for offline optimization. The previous algorithm of Helmbold et al. (1998) can be seen as a variant of gradient descent. On the other hand, the new algorithm Online Newton Step takes advantage of the second derivative of the functions.

We note that this paper does not include comparisons to the recent algorithm of Borodin et al. (2004), despite its excellent experimental results. The reason is that their heuristic is not a universal algorithm since it does not theoretically guarantee low regret.

To evaluate our algorithm, we reproduce the previous experiments of (Cover, 1991) and (Helmbold et al., 1998). Also, we test the algorithms on some more datasets, and evaluate their performance on additional metrics such as Annualized Percentage Yields (APYs), Sharpe ratio and mean-variance optimality. The new algorithm outperforms previous algorithms in nearly all these experiments under all the performance metrics we tested.

2. Notation and Preliminaries

Let the number of stocks in the portfolio be n. On every trading period t, for $t = 1, \ldots, T$, the investor observes a *price relative vector* $\mathbf{r}_t \in \mathbb{R}^n$, such that $\mathbf{r}_t(j)$ is the ratio of the closing price of stock j on day t to the closing price on day $t-1$. A portfolio $\mathbf{p}$ is a distribution on the n stocks, so it is a point in the n-dimensional simplex S_n. If the investor uses a portfolio $\mathbf{p}_t$ on day t, his wealth changes by a factor of $\mathbf{p}_t \cdot \mathbf{r}_t \triangleq \mathbf{p}_t^\top \mathbf{r}_t$. Thus, after T periods, the wealth achieved per dollar invested is $\prod_{t=1}^T (\mathbf{p}_t \cdot \mathbf{r}_t)$. The logarithmic growth ratio is $\sum_{t=1}^T \log(\mathbf{p}_t \cdot \mathbf{r}_t)$. An investor using a CRP $\mathbf{p}$ achieves the logarithmic growth ratio $\sum_{t=1}^T \log(\mathbf{p} \cdot \mathbf{r}_t)$. The best CRP in hindsight $\mathbf{p}^*$ is the one which maximizes this quantity. The regret of an online algorithm, `Alg`, which produces portfolios $\mathbf{p}_t$ for $t = 1, \ldots, T$, is defined to be

$$\text{Regret}(\texttt{Alg}) \triangleq \sum_{t=1}^T \log(\mathbf{p}^* \cdot \mathbf{r}_t) - \sum_{t=1}^T \log(\mathbf{p}_t \cdot \mathbf{r}_t).$$

Since scaling $\mathbf{r}_t$ by a constant affects the logarithmic growth rations of both the best CRP and `Alg` by the same additive factor, the regret does not change. So we assume without loss of generality that for all t, $\mathbf{r}_t$ is scaled so that $\max_j \mathbf{r}_t(j) = 1$. We also make the assumption that after this scaling, all the $\mathbf{r}_t(j)$ are bounded below by the *market variability parameter* $\alpha > 0$. This has been called the *no-junk-bond assumption* by Agarwal and Hazan, and can be interpreted to mean that no stock crashes to zero value over the trading period.

With this setup, Cover (1991) gave the first universal portfolio selection algorithm which had the optimal regret $O(\log T)$, without dependence on the market variability parameter α. His algorithm, however, needs $\Omega(t^n)$ time for computing the portfolio $\mathbf{p}_t$ and is clearly impractical. Kalai and Vempala (2003) gave a polynomial implementation of the algorithm using sampling of logconcave functions from convex domains (Lovász & Vempala, 2003b; Lovász & Vempala, 2003a), and this results in a randomized polynomial time algorithm, though the polynomial is still quite large. Helmbold et al. gave an algorithm which needs just linear (in n) time and space per period but has $O(\sqrt{T})$ regret, under the no-junk-bonds assumption.

3. Online Newton Step

Our algorithm, Online Newton Step, is presented below. It takes parameters η, and β which are required for the theoretical analysis. It also takes a heuristic tuning parameter, δ, which we set only for the purpose of experimentation.

ONS(η, β, δ)

- On period 1, use the uniform portfolio $\mathbf{p}_1 = \frac{1}{n}\mathbf{1}$.
- On period $t > 1$: Play strategy $\tilde{\mathbf{p}}_t \triangleq (1-\eta)\mathbf{p}_t + \eta \cdot \frac{1}{n}\mathbf{1}$, such that:

$$\mathbf{p}_t = \Pi_{S_n}^{\mathbf{A}_{t-1}} \left(\delta \mathbf{A}_{t-1}^{-1} \mathbf{b}_{t-1}\right)$$

  where $\mathbf{b}_{t-1} = (1 + \frac{1}{\beta}) \sum_{\tau=1}^{t-1} \nabla[\log_\tau(\mathbf{p}_\tau \cdot \mathbf{r}_\tau)]$, $\mathbf{A}_{t-1} = \sum_{\tau=1}^{t-1} -\nabla^2[\log(\mathbf{p}_\tau \cdot \mathbf{r}_\tau)] + \mathbf{I}_n$, and $\Pi_{S_n}^{\mathbf{A}_{t-1}}$ is the projection in the norm induced by $\mathbf{A}_{t-1}$, viz.,

$$\Pi_{S_n}^{\mathbf{A}_{t-1}}(\mathbf{q}) = \arg\min_{\mathbf{p} \in S_n} (\mathbf{q} - \mathbf{p})^\top \mathbf{A}_{t-1} (\mathbf{q} - \mathbf{p})$$

Figure 1. The Online Newton Step algorithm.

The Online Newton Step algorithm, shown in Figure 1, has optimal regret and efficient computability. It is a Newton-based approach which utilizes the gradient (denoted ∇) and the Hessian (denoted ∇^2) of the log function. It can be implemented very efficiently:

all it needs to do, per iteration, is compute an $n \times n$ matrix inverse, a matrix-vector product, and a projection into the simplex. Aside from the projection, all the other operations can be implemented in $O(n^2)$ time and space using the matrix inversion lemma (Brookes, 2005). The projection itself can be implemented very efficiently in practice using projected gradient descent methods.

We now proceed to analyze the algorithm theoretically, and show that under the no-junk-bond assumption, the algorithm has $O(\log T)$ regret, whereas without the assumption, the regret becomes $O(\sqrt{T})$.

Theorem 1. *The ONS algorithm has the following performance guarantees:*

1. *Assume that the market has variability parameter α. Then setting $\eta = 0$, $\beta = \frac{\alpha}{8\sqrt{n}}$, and $\delta = 1$, we have*

$$\text{Regret(ONS)} \;\leq\; \frac{10n^{1.5}}{\alpha} \log\left[\frac{nT}{\alpha^2}\right].$$

2. *With no assumptions on the market variability parameter, by setting $\eta = \frac{n^{1.25}}{\sqrt{T\log(nT)}}$, $\beta = \frac{1}{8n^{0.25}\sqrt{T\log(nT)}}$, and $\delta = 1$, we have*

$$\text{Regret(ONS)} \;\leq\; 22n^{1.25}\sqrt{T\log(nT)}.$$

Being a specialization of the Online Newton Step algorithm of Hazan et al., the analysis proceeds along the same lines. First, define $\nabla_t = \nabla[\log(\mathbf{p}_t \cdot \mathbf{r}_t)] = \frac{1}{\mathbf{p}_t \cdot \mathbf{r}_t}\mathbf{r}_t$. Note that $\nabla^2[\log(\mathbf{p}_t \cdot \mathbf{r}_t)] = \frac{-1}{(\mathbf{p}_t \cdot \mathbf{r}_t)^2}\mathbf{r}_t\mathbf{r}_t^\top = -\nabla_t\nabla_t^\top$, so $\mathbf{A}_t = \sum_{\tau=1}^{t} \nabla_\tau\nabla_\tau^\top + \mathbf{I}_n$. We will use this expression for $\mathbf{A}_t$ throughout the analysis.

Now define the functions $f_t : S_n \to \mathbb{R}$ as follows:

$$f_t(\mathbf{p}) \;\triangleq \log(\mathbf{p}_t \cdot \mathbf{r}_t) + \nabla_t^\top(\mathbf{p} - \mathbf{p}_t) - \frac{\beta}{2}[\nabla_t^\top(\mathbf{p} - \mathbf{p}_t)]^2$$

where $\beta = \frac{\alpha}{8\sqrt{n}}$. Note that $f_t(\mathbf{p}_t) = \log(\mathbf{p}_t \cdot \mathbf{r}_t)$. Furthermore, by the Taylor expansion applied to the logarithm function (see also Lemma (2) in (Hazan et al., 2006)), we get that, for all $\mathbf{p} \in S_n$: $\log(\mathbf{p} \cdot \mathbf{r}_t) \leq f_t(\mathbf{p})$. This implies that

$$\max_{\mathbf{p}} \sum_t \log(\mathbf{p}\cdot\mathbf{r}_t) - \log(\mathbf{p}_t\cdot\mathbf{r}_t) \leq \max_{\mathbf{p}} \sum_t f_t(\mathbf{p}) - f_t(\mathbf{p}_t), \tag{1}$$

so it suffices to bound the RHS of (1).

Lemma 2. *For all t, we have*

$$\mathbf{p}_t \;=\; \arg\max_{\mathbf{p}\in S_n} \sum_{\tau=1}^{t-1} f_t(\mathbf{p}) - \frac{\beta}{2}\|\mathbf{p}\|^2.$$

Proof. For $t = 1$, the uniform portfolio $\mathbf{p}_1 = \frac{1}{n}\mathbf{1}$ maximizes $-\frac{\beta}{2}\|\mathbf{p}\|^2$. For $t > 1$, expanding out the expressions for $f_\tau(\mathbf{p})$, multiplying by $\frac{2}{\beta}$ and getting rid of constants, the problem reduces to maximizing the following function over $\mathbf{p} \in S_n$:

$$\sum_{\tau=1}^{t-1}\left[-\mathbf{p}^\top\nabla_\tau\nabla_\tau^\top\mathbf{p} + 2\left(\mathbf{p}_\tau^\top\nabla_\tau\nabla_\tau^\top + \frac{1}{\beta}\nabla_\tau^\top\right)\mathbf{p}\right] - \mathbf{p}^\top\mathbf{p}$$
$$= \;-\mathbf{p}^\top\mathbf{A}_{t-1}\mathbf{p} + 2\mathbf{b}_{t-1}^\top\mathbf{p}.$$

The solution of this maximization is exactly the projection $\Pi_K^{\mathbf{A}_{t-1}}(\mathbf{A}_{t-1}^{-1}\mathbf{b}_{t-1})$ as specified by Online Newton Step. □

Proof. **(Theorem 1)**
Part 1. We need to bound the RHS of (1), $\max_{\mathbf{p}} \sum_t f_t(\mathbf{p}) - f_t(\mathbf{p}_t)$. A simple induction (see (Hazan et al., 2006)) shows that for any $\mathbf{p}$,

$$-\frac{\beta}{2}\|\mathbf{p}_1\|^2 + \sum_t f_t(\mathbf{p}_{t+1}) \geq \sum_t f_t(\mathbf{p}) - \frac{\beta}{2}\|\mathbf{p}\|^2.$$

In Lemma 4 below, we bound $\sum_t f_t(\mathbf{p}_{t+1}) - f_t(\mathbf{p}_t)$. Since $\frac{\beta}{2}[\|\mathbf{p}\|^2 - \|\mathbf{p}_1\|^2] \leq \frac{\beta}{2}$, we can bound the regret as:

$$\text{Regret(ONS)} \;\leq\; \frac{1}{\beta}n\log\left[\frac{nT}{\alpha^2}\right] + \frac{\beta}{2}.$$

Now the stated regret bound follows by plugging in the specified choice of parameters.

Part 2. The following lemma can be deduced from Theorem 2 in (Helmbold et al., 1998). The stated regret bound follows by using the lemma with the specified choice of parameters with the regret bound from part 1.

Lemma 3. *For an online algorithm* `Alg`*, let the derived algorithm* `SmoothAlg` *use the smoothened portfolio $\tilde{\mathbf{p}}_t = (1-\eta)\mathbf{p}_t + \eta \cdot \frac{1}{n}\mathbf{1}$ where $\mathbf{p}_t$ is the portfolio computed by* `Alg` *on day t. Then the regret can be bounded as:*

$$\text{Regret}(\texttt{SmoothAlg}) \;\leq\; \text{Regret}(\texttt{Alg}) + 2\eta T$$

where Regret(`Alg`) *is computed assuming the variability parameter α is at least $\frac{\eta}{n}$.*

□

Lemma 4.

$$\sum_{t=1}^{T}[f_t(\mathbf{p}_{t+1}) - f_t(\mathbf{p}_t)] \;\leq\; \frac{1}{\beta}n\log\left[\frac{nT}{\alpha^2}\right].$$

Lemma 4. For the sake of readability, we introduce some notation. Define the function $F_t \triangleq \sum_{\tau=1}^{t-1} f_\tau$. Note that $\nabla f_t(\mathbf{p}_t) = \nabla_t$ by the definition of f_t. Finally, let Δ be the forward difference operator, for example, $\Delta\mathbf{p}_t = (\mathbf{p}_{t+1} - \mathbf{p}_t)$ and $\Delta\nabla F_t(\mathbf{p}_t) = (\nabla F_{t+1}(\mathbf{p}_{t+1}) - \nabla F_t(\mathbf{p}_t))$.

We use the gradient bound, which follows from the concavity of f_t:

$$f_t(\mathbf{p}_{t+1}) - f_t(\mathbf{p}_t) \le \nabla f_t(\mathbf{p}_t)^\top(\mathbf{p}_{t+1} - \mathbf{p}_t) = \nabla_t^\top \Delta\mathbf{p}_t. \tag{2}$$

The gradient of F_{t+1} can be written as:

$$\nabla F_{t+1}(\mathbf{p}) = \sum_{\tau=1}^{t}[\nabla_\tau - \beta\nabla_\tau\nabla_\tau^\top(\mathbf{p} - \mathbf{p}_\tau)]. \tag{3}$$

Therefore,

$$\nabla F_{t+1}(\mathbf{p}_{t+1}) - \nabla F_{t+1}(\mathbf{p}_t) \;=\; -\beta\mathbf{A}_t\Delta\mathbf{p}_t. \tag{4}$$

The LHS of (4) is

$$\nabla F_{t+1}(\mathbf{p}_{t+1}) - \nabla F_{t+1}(\mathbf{p}_t) \;=\; \Delta\nabla F_t(\mathbf{p}_t) - \nabla_t. \tag{5}$$

Putting (4) and (5) together we get

$$-\beta\mathbf{A}_t\Delta\mathbf{p}_t = \Delta\nabla F_t(\mathbf{p}_t) - \nabla_t. \tag{6}$$

Pre-multiplying by $-\frac{1}{\beta}\nabla_t^\top\mathbf{A}_t^{-1}$, we get an expression for the gradient bound (2):

$$\begin{aligned}\nabla_t^\top\Delta\mathbf{p}_t &= \frac{1}{\beta}\nabla_t^\top\mathbf{A}_t^{-1}[\Delta\nabla F_t(\mathbf{p}_t) - \nabla_t] \\ &= -\frac{1}{\beta}\nabla_t^\top\mathbf{A}_t^{-1}[\Delta\nabla F_t(\mathbf{p}_t)] + \frac{1}{\beta}\nabla_t^\top\mathbf{A}_t^{-1}\nabla_t.\end{aligned} \tag{7}$$

Claim 1. *The first term of (7) is bounded as follows:*

$$-\frac{1}{\beta}\nabla_t^\top\mathbf{A}_t^{-1}[\Delta\nabla F_t(\mathbf{p}_t)] \;\le\; 0.$$

Proof. Since $\mathbf{p}_\tau$ maximizes F_τ over S_n, we have

$$\nabla F_\tau(\mathbf{p}_\tau)^\top(\mathbf{p} - \mathbf{p}_\tau) \;\le\; 0. \tag{8}$$

for any point $\mathbf{p} \in S_n$. Using (8) for $\tau = t$ and $\tau = t+1$, we get

$$\begin{aligned}0 \;&\ge\; \nabla F_{t+1}(\mathbf{p}_{t+1})^\top(\mathbf{p}_t - \mathbf{p}_{t+1}) + \nabla F_t(\mathbf{p}_t)^\top(\mathbf{p}_{t+1} - \mathbf{p}_t) \\ &= -[\Delta\nabla F_t(\mathbf{p}_t)]^\top\Delta\mathbf{p}_t \\ &= \frac{1}{\beta}[\Delta\nabla F_t(\mathbf{p}_t)]^\top\mathbf{A}_t^{-1}[\Delta\nabla F_t(\mathbf{p}_t) - \nabla_t] \\ &\qquad \text{(by solving for } \Delta\mathbf{p}_t \text{ in (6))} \\ &= \frac{1}{\beta}[\Delta\nabla F_t(\mathbf{p}_t)]^\top\mathbf{A}_t^{-1}[\Delta\nabla F_t(\mathbf{p}_t)] \\ &- \frac{1}{\beta}[\Delta\nabla F_t(\mathbf{p}_t)]^\top\mathbf{A}_t^{-1}\nabla_t \\ &\ge\; -\frac{1}{\beta}[\Delta\nabla F_t(\mathbf{p}_t)]^\top\mathbf{A}_t^{-1}\nabla_t. \\ &\qquad \text{(since } \mathbf{A}_t^{-1} \succeq 0 \Rightarrow \forall\mathbf{p}:\ \mathbf{p}^\top\mathbf{A}_t^{-1}\mathbf{p} \ge 0)\end{aligned}$$

as required. □

Now we bound the second term of (7). Sum up from $t = 1$ to T, and apply Lemma 5 (see Lemma 6 from (Hazan et al., 2006)) below with $\mathbf{A}_0 = \mathbf{I}_n$ and $v_t = \nabla_t$.

$$\frac{1}{\beta}\sum_{t=1}^{T}\nabla_t^\top\mathbf{A}_t^{-1}\nabla_t \;\le\; \frac{1}{\beta}\log\left[\frac{|\mathbf{A}_T|}{|\mathbf{A}_0|}\right] \;\le\; \frac{1}{\beta}n\log\left[\frac{nT}{\alpha^2}\right].$$

The second inequality follows since $\mathbf{A}_T = \sum_{t=1}^{T}\nabla_t\nabla_t^\top$ and $\|\nabla_t\| \le \frac{\sqrt{n}}{\alpha}$, and so $|\mathbf{A}_T| \le (\frac{nT}{\alpha^2})^n$. □

Lemma 5. *For $t = 1, 2, \ldots, T$, let $\mathbf{A}_t = \mathbf{A}_0 + \sum_{\tau=1}^{t} v_\tau v_\tau^\top$ for a positive definite matrix $\mathbf{A}_0$ and vectors $v_1, v_2, \ldots, v_T$. Then the following inequality holds:*

$$\sum_{t=1}^{T} v_t^\top\mathbf{A}_t^{-1}v_t \;\le\; \log\left[\frac{|\mathbf{A}_T|}{|\mathbf{A}_0|}\right].$$

3.1. Internal Regret

Stoltz and Lugosi (2005) extend the game-theoretic notion of internal regret to the case of online portfolio selection problems. The notion captures the following cause of regret to an online investor: in hindsight, how much more money could she have made, had she transferred all the money she invested in stock i, to stock j on all the trading days?

Formally, for a portfolio $\mathbf{p}$, define $\mathbf{p}^{i\to j}$ as follows: $\mathbf{p}_i^{i\to j} = 0$, $\mathbf{p}_j^{i\to j} = \mathbf{p}_i + \mathbf{p}_j$, and $\mathbf{p}_k^{i\to j} = \mathbf{p}_k$ if $k \ne i, j$. The internal regret is defined to be

$$\max_{ij}\sum_{t=1}\log(\mathbf{p}_t^{i\to j}\cdot\mathbf{r}_t) - \sum_{t=1}^{T}\log(\mathbf{p}_t\cdot\mathbf{r}_t).$$

A straightforward application of the technique of Stoltz and Lugosi (2005) results in an algorithm, called IR-ONS, that achieves logarithmic internal regret. In the full version of the paper, we prove:

Theorem 6. *Assume that the market has variability parameter α. Then setting $\eta = 0$, $\beta = \frac{\alpha}{8\sqrt{n}}$, and $\delta = 1$, we have*

$$\text{InternalRegret(IR-ONS)} \;\le\; \frac{20n^3}{\alpha}\log\left[\frac{nT}{\alpha^2}\right].$$

4. Experimental Results

We implemented the algorithms presented in (Agarwal & Hazan, 2005) and (Hazan et al., 2006) as well as the algorithms of Cover (1991)[1], the Multiplicative Weights algorithm of Helmbold et al. (1998), and the uniform CRP. We also applied the technique of Stoltz and Lugosi (2005) to the algorithms of Helmbold et al. (1998) and this paper to get variants which minimize *internal regret*. We implemented Online Newton Step with parameters $\eta = 0$, $\beta = 1$, and $\delta = \frac{1}{8}$. Unless otherwise noted, we omit the results for IR-ONS because it was inferior to ONS.

We performed tests on the historical stock market data from the New York Stock Exchange (NYSE) used by Cover and Helmbold et al. In addition we randomly selected portfolios of various sizes from a set of 50 randomly chosen S&P 500 stocks[2] and performed experiments over the past 4 years data from 12th December, 2001 to 30th November, 2005 obtained from Yahoo! Finance.

Table 1. Abbreviations used in the experiments.

BCRP	Best CRP
UCRP	Uniform CRP
Universal	(Cover, 1991)
MW	(Helmbold et al., 1998)
IR-MW	Internal regret variant of MW
ONS	Online Newton Step
IR-ONS	Internal regret variant of ONS

Performance Measures. The performance measures we used were Annualized Percentage Yields (APYs), Sharpe ratio and mean-variance optimality.

4.1. Performance vs. Portfolio Size

To measure the dependence of the performance of various algorithms on portfolio size we picked 50 sets of n random stocks from the data set, for values of n ranging from 5 to 40. All algorithms were run on the data, trading once every two weeks. The choice of trading period was to permit completion of the Universal algorithm in reasonable time. The trading period did not seem to affect the relative performance of the algorithms. The results are shown in Figure 4.

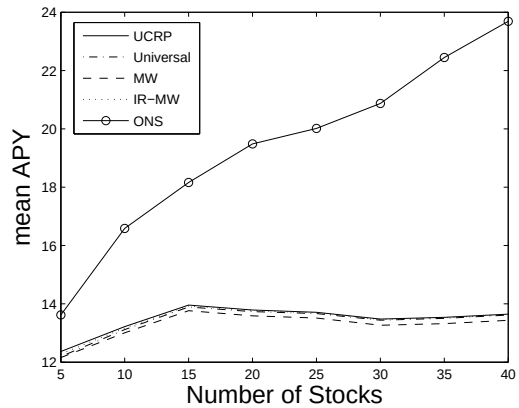

Figure 2. Performance vs. Portfolio Size

The improvement in the performance of ONS with increasing number of stocks is quite stark. The reason for this seems to be that ONS does an extremely good job of tracking the best stock in a given portfolio. Adding more stocks causes some good stock to get added, which ONS proceeds to track. Other algorithms behave more like the uniform CRP and so average out the increase in wealth due to the addition of a good stock. Figure 3 shows how ONS tracks CMC, which out-performs Kin-Ark for the test period, in a dataset composed of Kin-Ark and CMC (also used by Cover) while other algorithms have a nearly uniform distribution on both the stocks. This is the reason ONS outperforms all other algorithms on this dataset, as can be seen in Figure 5.

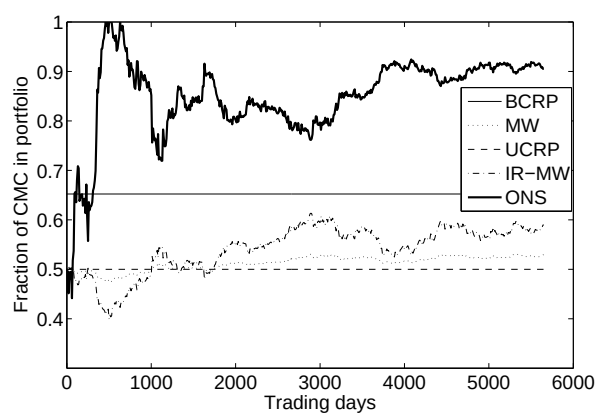

Figure 3. How ONS tracks CMC.

[1]Since we implemented Cover's algorithm by random sampling, there is a small degree of variability in the measurements recorded here. We used 1000 samples, which as suggested by (Stoltz & Lugosi, 2005), is sufficient to get a good estimate of the behavior of that algorithm.

[2]The set of stocks used was RTN, SLB, ABK, PEG, KMG, FITB, CL, PSA, DOV, NKE, AT, NEM, VMC, D, CPWR, NVDA, SRE, HPQ, CMX, LXK, GPC, ABI, PGL, QLGC, OMX, QCOM, KO, PMTC, SWK, CTXS, FSH, HON, COF, LH, KMG, BLL, WB, OMX, K, LUV, DIS, SFA, APOL, HUM, CVH, IR, SPG, WY, TYC, NKE.

4.2. Random Stocks from S&P 500

We tested the average APYs (over 50 trials of 10 random stocks from the S&P 500 list mentioned before) of the algorithms, for different frequencies of rebalancing, namely daily, weekly, fortnightly and monthly. As can be seen in figure 4 the performance of the ONS algorithm is superior to all other algorithms in all the 4 cases. As is expected the performance of all algorithms degrades as trading frequency decreases, but not very significantly. The simple strategy of maintaining a uniform constant-rebalanced portfolio seems to outperform all previous algorithms. This rather surprising fact has been observed by Borodin et al. (2004) also.

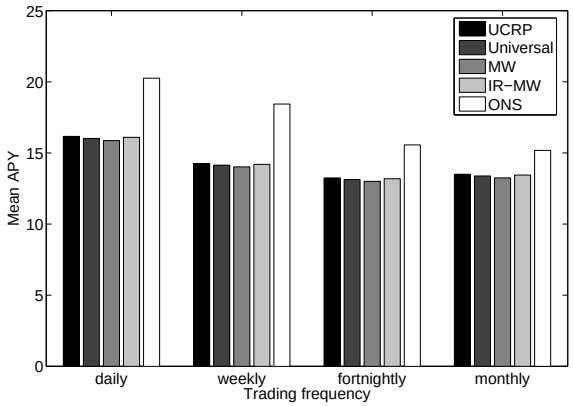

Figure 4. Performance vs. Trading Period.

4.3. Cover's Experiments

We replicated the experiments of Cover and Helmbold et al. on Iroquios Brands Ltd. and Kin Ark Corp., Commercial Metals (CMC) and Kin Ark, CMC and Meicco Corp., IBM and Coca Cola for the same 22 year period from 3rd July, 1962 to 31st December, 1984. As can be seen from Figure 5, ONS outperforms all other algorithms except on the Iroquios Brands Ltd. and Kin Ark Corp. dataset.

Figure 6 shows how the total wealth (per dollar invested) varies over the entire period using the different algorithms for a portfolio of IBM and Coke. The ONS algorithm, and its internal regret variant IR-ONS, outperform even the best constant-rebalanced portfolio.

4.4. Stock Volatility

We took the 50 stock data set used in previous experiments which had a history for 1000 days traded fortnightly and sorted them according to volatility and created two sets: the 10 stocks with largest and smallest price variance. Then we applied the different algorithms on the two different sets.

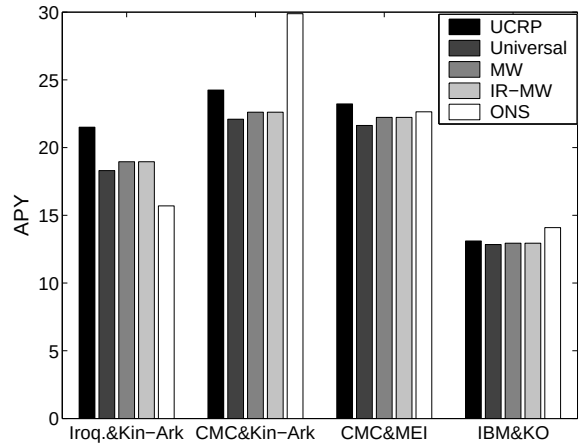

Figure 5. Four pairs of stocks tested by Cover (1991) and Helmbold et al. (1998).

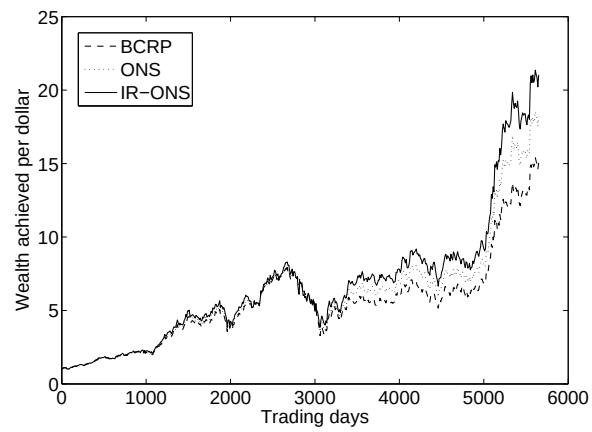

Figure 6. Wealth achieved by various algorithms on a portfolio consisting of IBM and Coke.

Figure 7 shows that the performance of ONS increases with market volatility whereas the performance of other algorithms decreases.

4.5. Margins Loans

In line with Cover (1991) and Helmbold et al. (1998), we also tested the case where the portfolio can buy stocks on margin. The data set we tested on was the 22 year IBM and Coca Cola data mentioned earlier. Results for this case are given in Table 4. The margin purchases we incorporate are 50% down and 50% loan. The ONS algorithm in fact enhances its performance edge over other algorithms if margin loans are allowed.

Table 2. Sharpe ratios for various algorithms on different datasets.

	Universal	**UCRP**	**MW**	**IR-MW**	**ONS**
Iro.&Kin-Ark	0.4986	0.5497	0.5390	0.5078	0.4578
CMC & Kin-Ark	0.5740	0.6020	0.5980	0.5812	0.7466
CMC & Meicco	0.3885	0.3834	0.3856	0.3854	0.5177
IBM & Coke	0.5246	0.5376	0.5356	0.5295	0.5824

Table 3. Minimum variance CRPs for various algorithms on different datasets. The number to the left of the slash is the volatility of the minimum variance CRP and the number to the right is the volatility of the algorithm on the dataset.

	Universal	**UCRP**	**MW**	**IR-MW**	**ONS**
Iro. & Kin-Ark	0.4598/0.4948	0.4929/0.4929	0.4803/0.4928	0.4606/0.4930	0.4603/0.5451
CMC & Meicco	0.1911/0.2728	0.1909/0.2723	0.1909/0.2717	0.1909/0.2718	0.2070/0.3510

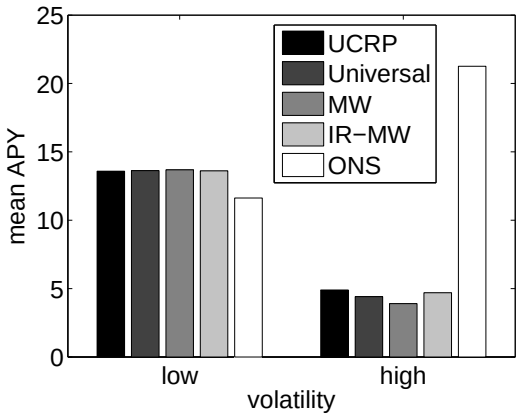

Figure 7. Performance of algorithms on high and low volatility datasets.

Table 4. Incorporating margin loans.

Algorithm	APY, no margin	APY with margin
UCRP	12.73	14.84
Universal	12.46	14.40
MW	12.57	14.39
IR-MW	12.57	14.62
ONS	13.68	16.15

4.6. Sharpe Ratio and Mean-Variance Optimal CRPs

It is a well-known fact that one can achieve higher returns by investing in riskier assets (Luenberger, 1998). So it is important to rule out the possibility of the ONS algorithm achieving higher returns compared to other algorithms by trading more riskily. Parameters like the Sharpe ratio and the optimal mean-variance portfolio are used to measure this risk versus reward tradeoff. Sharpe ratio is defined as $\frac{R_p - R_f}{\sigma_p}$ where R_p is the average yearly return of the algorithm, which indicates reward, R_f is the risk-free rate (typically the average rate of return of Treasury bills), and σ_p is the standard deviation of the returns of the algorithm, which indicates its volatility risk. Higher the Sharpe Ratio the better is the algorithm at balancing high rewards with low risk.

The mean-variance optimal CRP for an algorithm is the CRP which achieves the same return as the algorithm but has minimum variance. This is the least risky CRP one could have used in hindsight to produce the same returns. The closer the volatility of the CRP to that of the algorithm, the better the algorithm is avoiding risk.

Table 2 shows that ONS has either the best or slightly smaller Sharpe ratio among all algorithms. In Table 3, it can be seen that ONS has comparable volatility to the minimum variance CRP, implying that ONS does not take excessive risk in its portfolio selection. In the case of IBM & Coke and Kin-Ark & CMC, ONS beats the Best CRP in hindsight. Hence the concept of the optimal mean-variance CRP does not apply and the results for this case are omitted.

4.7. Running Times

As expected, ONS runs slightly slower than MW, but both are much faster than Universal. We measured the running time (in seconds) of these algorithms on the 22 year data sets mentioned earlier. The machine used was a dual Intel 933MHz PIII processor with 1GB operated with Linux Fedora Core 3 operating system. The average running time, on the four data sets we considered, was 4882 seconds for the Universal algorithm[3], whereas MW and ONS took 3.7 and 26.7 seconds, respectively. This clearly shows the significant advantage of ONS over Universal and that it is comparable with MW in terms of computational efficiency.

[3]With 1000 samples.

5. Conclusions

We experimentally tested the recently proposed algorithms of (Agarwal & Hazan, 2005; Hazan et al., 2006) for the universal portfolio selection problem. The Online Newton Step algorithm is extremely fast in practice as expected from the theoretical guarantees. Moreover, it seems to be better than previous algorithms at tracking the best stock.

It would be interesting to combine the anti-correlated heuristic of Borodin et al. (2004) with the best stock tracking ability of our algorithm. Another open problem is to incorporate transaction costs into the algorithm, as done by Blum and Kalai (1999) for Cover's algorithm.

Acknowledgements

We would like to thank Sanjeev Arora and Moses Charikar for helpful suggestions. Elad Hazan and Satyen Kale were supported by Sanjeev Arora's NSF grants MSPA-MCS 0528414, CCF 0514993, ITR 0205594. We would also like to thank Gilles Stoltz for providing us with the data sets for experiments and helpful suggestions.

References

Agarwal, A., & Hazan, E. (2005). New algorithms for repeated play and universal portfolio management. *Princeton University Technical Report TR-740-05*.

Algoet, P., & Cover, T. (1988). Asymptotic optimality and asymptotic equipartition properties of log-optimum investment. *Annals of Probability*, *2*, 876–898.

Bell, R., & Cover, T. (1980). Competitive optimality of logarithmic investment. *Mathematics of Operations Research*, *2*, 161–166.

Bell, R., & Cover, T. (1988). Game-theoretic optimal portfolios. *Management Science*, *6*, 724–733.

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

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

Brookes, M. (2005). The matrix reference manual. *[online] www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html*.

Cover, T. (1991). Universal portfolios. *Mathematical Finance*, *1*, 1–19.

Hannan, J. (1957). Approximation to bayes risk in repeated play. *In M. Dresher, A. W. Tucker and P. Wolfe, editors, Contributions to the Theory of Games*, *III*, 97–139.

Hazan, E., Kalai, A., Kale, S., & Agarwal, A. (2006). Logarithmic regret algorithms for online convex optimization. *To appear in the 19th Annual Conference on Learning Theory (COLT)*.

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

Kalai, A., & Vempala, S. (2003). Efficient algorithms for universal portfolios. *Journal Machine Learning Research*, *3*, 423–440.

Kalai, A., & Vempala, S. (2005). Efficient algorithms for on-line optimization. *Journal of Computer and System Sciences*, *71(3)*, 291–307.

Kelly, J. (1956). A new interpretation of information rate. *Bell Systems Technical Journal*, 917–926.

Larson, D. C. (1986). Growth optimal trading strategies. *Ph.D. dissertation, Stanford Univ., Stanford, CA*.

Lovász, L., & Vempala, S. (2003a). *The geometry of logconcave functions and an $O^*(n^3)$ sampling algorithm* (Technical Report MSR-TR-2003-04). Microsoft Research.

Lovász, L., & Vempala, S. (2003b). Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. *Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS)* (pp. 650–659).

Luenberger, D. G. (1998). *Investment science*. Oxford: Oxford University Press.

Merhav, N., & Feder, M. (1992). Universal sequential learning and decision from individual data sequences. *5th COLT* (pp. 413–427). Pittsburgh, Pennsylvania, United States.

Ordentlich, E., & Cover, T. M. (1996). On-line portfolio selection. *9th COLT* (pp. 310–313). Desenzano del Garda, Italy.

Stoltz, G., & Lugosi, G. (2005). Internal regret in on-line portfolio selection. *Machine Learning*, *59*, 125–159.