*Mathematical Finance, 8(4):325-347, 1998.*

# On-Line Portfolio Selection Using Multiplicative Updates

David P. Helmbold
Computer and Information Sciences
University of California
Santa Cruz, CA 95064
dph@cse.ucsc.edu

Robert E. Schapire
AT&T Labs
180 Park Avenue, Room A279
Florham Park, NJ 07932
schapire@research.att.com

Yoram Singer
AT&T Labs
180 Park Avenue, Room A277
Florham Park, NJ 07932
singer@research.att.com

Manfred K. Warmuth
Computer and Information Sciences
University of California
Santa Cruz, CA 95064
manfred@cse.ucsc.edu

**Abstract**

We present an on-line investment algorithm which achieves almost the same wealth as the best constant-rebalanced portfolio determined in hindsight from the actual market outcomes. The algorithm employs a multiplicative update rule derived using a framework introduced by Kivinen and Warmuth. Our algorithm is very simple to implement and requires only constant storage and computing time per stock in each trading period. We tested the performance of our algorithm on real stock data from the New York Stock Exchange accumulated during a 22-year period. On this data, our algorithm clearly outperforms the best single stock as well as Cover's universal portfolio selection algorithm. We also present results for the situation in which the investor has access to additional "side information."

# 1 Introduction

We present an on-line investment algorithm which achieves almost the same wealth as the best constant-rebalanced portfolio investment strategy. The algorithm employs a multiplicative update rule derived using a framework introduced by Kivinen and Warmuth [20]. Our algorithm is very simple to implement and its time and storage requirements grow linearly in the number of stocks. Experiments on real New York Stock Exchange data indicate that our algorithm outperforms Cover's [9] universal portfolio algorithm.

The following simple example demonstrates the power of constant-rebalanced portfolio strategies. Assume that two investments are available. The first is a risk-free, no-growth investment stock whose value never changes. The second investment is a hypothetical highly volatile stock. On even days, the value of this stock doubles and on odd days its value is halved. The relative returns of the first stock can be described by the sequence $1, 1, 1, \ldots$ and of the second by the sequence $\frac{1}{2}, 2, \frac{1}{2}, 2, \ldots$. Neither investment alone can increase in value by more than a factor of 2, but a strategy combining the two investments can grow exponentially. One such strategy splits the investor's total wealth evenly between the two investments, and maintains this even split at the end of each day. On odd days the relative wealth decreases by a factor of $\frac{1}{2} \times 1 + \frac{1}{2} \times \frac{1}{2} = \frac{3}{4}$. However, on even days the relative wealth grows by $\frac{1}{2} \times 1 + \frac{1}{2} \times 2 = \frac{3}{2}$. Thus, after two consecutive trading days the investor's wealth grows by a factor of $\frac{3}{4} \times \frac{3}{2} = \frac{9}{8}$. It takes only twelve days to double the wealth, and over $2n$ trading days the wealth grows by a factor of $(\frac{9}{8})^n$.

Investment strategies which maintain a fixed fraction of the total wealth in each of the underlying investments, like the one described above, are called *constant-rebalanced portfolio strategies*. Previously, Cover [9] described a portfolio-selection algorithm that provably performs "almost as well" as the *best* constant-rebalanced portfolio. In this paper, we describe a new algorithm with similar properties. Like the results for Cover's algorithm, this performance property is proven without making any statistical assumptions on the nature of the stock market.

The theoretical bound we prove on the performance of our algorithm relative to the best constant-rebalanced portfolio is not as strong as the bound proved by Cover and Ordentlich [11]. However, the time and space required for our algorithm is linear in the number of stocks whereas Cover's algorithm is exponential in the number of stocks. Moreover, we tested our algorithm experimentally on historical data from the New York Stock Exchange (NYSE) accumulated over a 22-year period, and found that our algorithm clearly outperforms the algorithm of Cover and Ordentlich.

Following Cover and Ordentlich [11], we also present results for the situation in which the investor has some finite "side information," such as the current interest rate. Side information may provide hints to the investor that one or a set of stocks are likely to outperform the other stocks in the portfolio. Moreover, the side information may be dependent on the past and future behavior of the market. At the beginning of each trading day, the side information is presented to the investor as a single scalar representing the "state" of the finite side information; the significance of this information must be learned by the investor.

# 2 Preliminaries

Consider a portfolio containing $N$ stocks. Each trading day,[1] the performance of the stocks can be described by a vector of *price relatives*, denoted by $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ where $x_i$ is the next day's opening price of the $i$th stock divided by its opening price on the current day. Thus the value

[1] The unit of time "day" was chosen arbitrarily; we could equally well use minutes, hours, weeks, etc. as the time between actions.

of an investment in stock $i$ increases (or falls) to $x_i$ times its previous value from one morning to the next. A portfolio is defined by a weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_N)$ such that $w_i \geq 0$ and $\sum_{i=1}^{N} w_i = 1$. The $i$th entry of a portfolio $\mathbf{w}$ is the proportion of the total portfolio value invested in the $i$th stock. Given a portfolio $\mathbf{w}$ and the price relatives $\mathbf{x}$, investors using this portfolio increase (or decrease) their wealth from one morning to the next by a factor of

$$\mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{N} w_i x_i.$$

### 2.1 On-line portfolio selection

In this paper, we are interested in on-line portfolio selection strategies. At the start of each day $t$, the portfolio selection strategy gets the previous price relatives of the stock market $\mathbf{x}^1, \ldots, \mathbf{x}^{t-1}$. From this information, the strategy immediately selects its portfolio $\mathbf{w}^t$ for the day. At the beginning of the next day (day $t+1$), the price relatives for day $t$ are observed and the investor's wealth increases by a factor of $\mathbf{w}^t \cdot \mathbf{x}^t$.

Over time, a sequence of daily price relatives $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^T$ is observed and a sequence of portfolios $\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^T$ is selected. From the beginning of day 1 through the beginning of day $T+1$, the wealth will have increased by a factor of

$$S_T(\{\mathbf{w}^t\}, \{\mathbf{x}^t\}) \stackrel{\text{def}}{=} \prod_{t=1}^{T} \mathbf{w}^t \cdot \mathbf{x}^t.$$

Since in a typical market the wealth grows exponentially fast, the formal analysis of our algorithm will be presented in terms of the normalized logarithm of the wealth achieved. We denote this normalized logarithm of the wealth by

$$LS_T(\{\mathbf{w}^t\}, \{\mathbf{x}^t\}) \stackrel{\text{def}}{=} \frac{1}{T} \sum_{t=1}^{T} \log\left(\mathbf{w}^t \cdot \mathbf{x}^t\right).$$

### 2.2 Constant-rebalanced portfolios

With the benefit of hindsight, on each day one can invest all of one's wealth in the single best-performing stock for that day. It is certainly absurd to hope to perform as well as a prescient agent with this level of information about the future. Instead, in this paper, we compete against a more restricted class of investment strategies called constant-rebalanced portfolios. As noted in the introduction, a constant-rebalanced portfolio is rebalanced each day so that a fixed fraction of the wealth is held in each of the underlying investments. Therefore, a constant-rebalanced portfolio strategy employs the same investment vector $\mathbf{w}$ on each trading day and the resulting wealth and normalized logarithmic wealth after $T$ trading days are

$$S_T(\mathbf{w}) \stackrel{\text{def}}{=} S_T(\mathbf{w}, \{\mathbf{x}^t\}) = \prod_{t=1}^{T} \mathbf{w} \cdot \mathbf{x}^t \quad , \quad LS_T(\mathbf{w}) \stackrel{\text{def}}{=} LS_T(\mathbf{w}, \{\mathbf{x}^t\}) = \frac{1}{T} \sum_{t=1}^{T} \log\left(\mathbf{w} \cdot \mathbf{x}^t\right) \quad .$$

Note that such a strategy might require vast amounts of trading, since at the beginning of each day $t$ the investment proportions are rebalanced back to the vector $\mathbf{w}$. In this paper we ignore commission costs (however, see the discussion in Section 6).

Given a sequence of daily price relatives $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^T$ we can define, in retrospect, the best rebalanced portfolio vector which would have achieved the maximum wealth $S_T$, and hence also the maximum logarithmic wealth, $LS_T$. We denote this portfolio by $\mathbf{w}^\star = \mathbf{w}^\star(T)$. That is,

$$\mathbf{w}^\star \stackrel{\text{def}}{=} \arg\max_{\mathbf{w}} S_T(\mathbf{w}) = \arg\max_{\mathbf{w}} LS_T(\mathbf{w}),$$

where the maximum is taken over all possible portfolio vectors (i.e., vectors in $\mathbb{R}^N$ with non-negative components that sum to one). Iterative methods for finding this vector using the entire sequence of price relatives $\mathbf{x}^1, \ldots, \mathbf{x}^T$ are discussed in our earlier paper [15] which gives several updates for solving a general mixture estimation problem, including multiplicative updates like those described in this paper. We denote the logarithmic wealth achieved using the optimal constant-rebalanced portfolio $\mathbf{w}^\star$ by $LS_T^\star(\{\mathbf{x}^t\})$. Whenever it is clear from the context, we will omit the dependency on the price relatives and simply denote the above by $LS^\star$. Clearly, $\mathbf{w}^\star$ depends on the entire sequence of price relatives $\{\mathbf{x}^t\}$ and may be dramatically different for different market behaviors.

Obviously, the optimal vector $\mathbf{w}^\star$ can only be computed after the entire sequence of price relatives is known (at which point, it is no longer of value). However, the algorithm described in this paper (as well as Cover's [9] algorithm) performs almost as well as $\mathbf{w}^\star$ while using only the previously observed history of price relatives to make each day's investment decision.

### 2.3 Universal portfolios

Cover [9] introduced the notion of universal portfolio. An on-line portfolio selection algorithm that results in the sequence $\{\mathbf{w}^t\}$ is said to be *universal* (relative to the set of all constant-rebalanced portfolios) if

$$\lim_{T\to\infty} \max_{\{\mathbf{x}^t\}} \left[ LS^\star(\{\mathbf{x}^t\}) - LS(\{\mathbf{w}^t\}, \{\mathbf{x}^t\}) \right] = 0 \ .$$

That is, a universal portfolio selection algorithm exhibits the same asymptotic growth rate in normalized logarithmic wealth as the best rebalanced portfolio for *any* sequence of price relatives $\{\mathbf{x}^t\}$.

In Section 3 we adapt a framework developed for supervised learning and give a simple update rule that selects a new portfolio vector from the previous one. We prove that this algorithm is universal in Section 4.

### 2.4 Side information

In reality, the investor might have more information than just the price relatives observed so far. Side information such as prevailing interest rates or consumer-confidence figures can indicate which stocks are likely to outperform the other stocks in the portfolio. Following Cover and Ordentlich [11], we denote the side information by an integer $y$ from a finite set $\{1, 2, \ldots, K\}$. Thus, the behavior of the market including the side information is now denoted by the sequence $\{\mathbf{x}^t, y^t\}$.

Following Cover and Ordentlich [11], we allow the constant-rebalanced portfolio to exploit the side information by expanding the single portfolio into a set of portfolios, one for each possible value of the side information. Thus, a constant-rebalanced portfolio with side information consists of the vectors $\mathbf{w}(1), \mathbf{w}(2), \ldots, \mathbf{w}(K)$ and uses portfolio vector $\mathbf{w}(y^t)$ on day $t$. The wealth and normalized logarithmic wealth resulting from using a set of constant-rebalanced portfolios based on side information are,

$$S_T(\mathbf{w}(\cdot), \{\mathbf{x}^t, y^t\}) \stackrel{\text{def}}{=} \prod_{t=1}^{T} \mathbf{w}(y^t) \cdot \mathbf{x}^t \ , \quad LS_T(\mathbf{w}(\cdot), \{\mathbf{x}^t, y^t\}) \stackrel{\text{def}}{=} \frac{1}{T} \sum_{t=1}^{T} \log\left(\mathbf{w}(y^t) \cdot \mathbf{x}^t\right) \ .$$

Just like the definition of the best constant-rebalanced portfolio, we define the best side information dependent portfolio set $\mathbf{w}^\star(\cdot)$ as the maximizer of $S_T(\mathbf{w}(\cdot), \{\mathbf{x}^t, y^t\})$. Note that the dimension of a side information dependent portfolio selection problem is $K$ times larger than the single portfolio selection problem.

The sequence of side information $\{y^t\}$ could be meaningless random noise, neither a function of the past market nor a predictor of future markets. On the other hand, it might be a perfect indicator of the best investment. Extending the two-investment example given in Section 1, we might have side information $y = 1$ on odd days (when the volatile stock loses half its value) and $y = 2$ on even days (when the volatile stock doubles). This side information can be exploited by the constant-rebalanced portfolio set $\mathbf{w}(1) = (1, 0)$ and $\mathbf{w}(2) = (0, 1)$ to double its wealth every other trading day. However, the only side information communicated to the investor (at the beginning of day $t$) is the single value $y^t$ with no further "explanations," and the sequence $\{y^t\}$ may or may not contain any useful information. Hence, the importance of each side information value must be learned from the performance of the market during previous trading days.

An on-line investment algorithm in this setting has access on day $t$ both to the past history of price relatives (as before) and to the past and current side information values $y^1, \ldots, y^t$. The goal of the algorithm now is to invest in a manner competitive with $S_T(\mathbf{w}^\star(\cdot), \{\mathbf{x}^t, y^t\})$, the wealth of the best constant-rebalanced portfolio with side information. One can easily define a notion of universality analogous to the definition given in Section 2.3.

As noticed by Cover and Ordentlich [11], the investor can partition the trading days based on the side information, and treat each partition separately. Exploiting the side information is therefore no more difficult than running $K$ copies of our algorithm, one for each possible value of the side information. Since the logarithm of the wealth is additive, the logarithm of the wealth on the entire sequence with side information is just the sum of the logarithms of the wealths generated by the $K$ copies of the algorithm.

### 2.5 Related work

Distributional methods are probably the most common approach to adaptive investment strategies for rebalanced portfolios. Kelly [19] assumed the existence of an underlying distribution of the price relatives and used Bayes decision theory to specify the next portfolio vector. Under various conditions, it was demonstrated (e.g. [5, 8, 6, 4, 2]) that with probability one the Bayes decision approach achieves the same growth rate of the wealth as the best rebalanced portfolio. In this approach, the price relative sequences can be drawn from one of a known set of possible distributions. This approach was used by Algoet [1] who considered the set of all ergodic and stationary distributions on infinite sequences, and estimated the underlying distribution in order to choose the next portfolio vector. Cover and Gluss [10] considered the restricted case where the set of price relatives is *finite* and gave an investment scheme with universal properties.

The most closely related previous results are by Cover [9] and Cover and Ordentlich [11]. They prove that certain investment strategies are universal without making any statistical assumptions on the nature of the stock market. Cover [9] proved that the wealth achieved by his universal portfolio algorithm is "almost as large" as the best constant-rebalanced portfolio. His analysis was improved by Cover and Ordentlich [11] who also introduced the notion of side information, and generalized Cover's algorithm to use the Dirichlet$(1/2, \ldots, 1/2)$ and the Dirichlet$(1, \ldots, 1)$ priors over the set of all possible portfolio vectors.

Cover and Ordentlich's investment strategies use an averaging method to pick their portfolio vectors. The portfolio vector used on day $t$ is the weighted average over *all* feasible portfolio vectors

(all $N$-dimensional vectors with non-negative components that sum to 1), where the weight of each possible portfolio vector is determined by its performance in the past. That is,

$$\mathbf{w}^t = \frac{\int \mathbf{w}\, S_{t-1}(\mathbf{w})\, d\mu(\mathbf{w})}{\int S_{t-1}(\mathbf{w})\, d\mu(\mathbf{w})} \quad , \tag{1}$$

where $d\mu$ is one of the Dirichlet distributions mentioned above. Note that the portfolio vectors are weighted according to their past performance, $S_{t-1}(\mathbf{w})$, as well as the prior $\mu(\mathbf{w})$. Discrete approximation [9] or recursive series expansion [11] are used to evaluate the above integrals. In both cases, however, the time and space required for finding the new portfolio vector appears to grow exponentially in the number of stocks. While the bounds achieved by the generalized universal portfolio algorithm of Cover and Ordentlich are stronger than ours, we show that on historical stock data our algorithm performs better while requiring time and space linear in the number of stocks.

# 3 Multiplicative portfolio selection algorithms

Our framework for updating a portfolio vector is analogous to the framework developed by Kivinen and Warmuth [20] for on-line regression. In this on-line framework the portfolio vector itself encapsulates the necessary information from the previous price relatives. Thus, at the start of day $t$, the algorithm computes its new portfolio vector $\mathbf{w}^{t+1}$ as a function of $\mathbf{w}^t$ and the just observed price relatives $\mathbf{x}^t$. In the linear regression setting analyzed by Kivinen and Warmuth, they show that good performance can be achieved by choosing a vector $\mathbf{w}^{t+1}$ that is "close" to $\mathbf{w}^t$. We adapt their method and find a new vector $\mathbf{w}^{t+1}$ that (approximately) maximizes the following function:

$$F(\mathbf{w}^{t+1}) = \eta \log(\mathbf{w}^{t+1} \cdot \mathbf{x}^t) - d(\mathbf{w}^{t+1}, \mathbf{w}^t), \tag{2}$$

where $\eta > 0$ is some parameter called the learning rate and $d$ is a distance measure that serves as a penalty term. This penalty term, $-d(\mathbf{w}^{t+1}, \mathbf{w}^t)$, tends to keep $\mathbf{w}^{t+1}$ close to $\mathbf{w}^t$. The purpose of the first term is to maximize the logarithmic wealth if the current price relative $\mathbf{x}^t$ is repeated. The learning rate $\eta$ controls the relative importance between the two terms. Intuitively, if $\mathbf{w}^t$ is far from the best constant-rebalanced portfolio $\mathbf{w}^\star$ then a small learning rate means that $\mathbf{w}^{t+1}$ will move only slowly toward $\mathbf{w}^\star$. On the other hand, if $\mathbf{w}^t$ is already close to $\mathbf{w}^\star$ then a large learning rate may cause the algorithm to be misled by day-to-day fluctuations.

Different distance functions lead to different update rules. One of the main contributions of this line of work is the use of the relative entropy as a distance function for motivating updates:

$$\mathrm{D}_{RE}(\mathbf{u}||\mathbf{v}) \stackrel{\text{def}}{=} \sum_{i=1}^{N} u_i \log \frac{u_i}{v_i} \quad .$$

Many other on-line algorithms with multiplicative weight updates [21, 3, 20, 14] are also motivated by this distance function and are thus rooted in the minimum relative entropy principle of Kullback [18, 13]. To derive learning rules using relative entropy, we set $d(\mathbf{w}^{t+1}, \mathbf{w}^t) = \mathrm{D}_{RE}(\mathbf{w}^{t+1}||\mathbf{w}^t)$.

It is hard to maximize $F$ since both terms depend non-linearly on $\mathbf{w}^{t+1}$. One possible approach is to use an iterative optimization algorithm, such as gradient projection, to find the maximum vector $\mathbf{w}^{t+1}$ that maximizes $F$ under the constraint $\sum_{i=1}^{N} w_i^{t+1} = 1$ (see, for instance, Fletcher [12]). This approach is time consuming as it requires solving a different non-linear equation on each trading period. Furthermore, as we later demonstrate in Section 5, the portfolio algorithm that finds the exact solution to Equation (2) in practice does not yield better results than the algorithm we

now present which is based on the following efficient approximation. Instead of finding the exact maximizer of $F$, we replace the first term with its first-order Taylor polynomial around $\mathbf{w}^{t+1} = \mathbf{w}^t$. This approximation is reasonable if $F$ satisfies a Lipschitz condition and the vector $\mathbf{w}^{t+1}$ is relatively close to $\mathbf{w}^t$. We also use a Lagrange multiplier to handle the constraint that the components of $\mathbf{w}^{t+1}$ must sum to one. This leads us to maximize $\hat{F}$ instead of $F$:

$$\hat{F}(\mathbf{w}^{t+1}, \gamma) \;=\; \eta \left( \log(\mathbf{w}^t \cdot \mathbf{x}^t) + \frac{\mathbf{x}^t \cdot (\mathbf{w}^{t+1} - \mathbf{w}^t)}{\mathbf{w}^t \cdot \mathbf{x}^t} \right) \;-\; d(\mathbf{w}^{t+1}, \mathbf{w}^t) + \gamma \left( \sum_{i=1}^{N} w_i^{t+1} - 1 \right) .$$

This is done by setting the $N$ partial derivatives to zero (for $1 \leq i \leq N$):

$$\frac{\partial \hat{F}(\mathbf{w}^{t+1}, \gamma)}{\partial w_i^{t+1}} \;=\; \eta \frac{x_i^t}{\mathbf{w}^t \cdot \mathbf{x}^t} - \frac{\partial d(\mathbf{w}^{t+1}, \mathbf{w}^t)}{\partial w_i^{t+1}} + \gamma = 0 \;\; . \tag{3}$$

If the relative entropy is used as the distance function then Equation (3) becomes

$$\eta \frac{x_i^t}{\mathbf{w}^t \cdot \mathbf{x}^t} - (\log \frac{w_i^{t+1}}{w_i^t} + 1) + \gamma = 0 \quad \text{or} \quad w_i^{t+1} = w_i^t \exp\left( \eta \frac{x_i^t}{\mathbf{w}^t \cdot \mathbf{x}^t} + \gamma - 1 \right) \;\; .$$

Enforcing the additional constraint $\sum_{i=1}^{N} w_i^{t+1} = 1$ gives a portfolio update which we call the *exponentiated gradient* (EG($\eta$)) update:

$$w_i^{t+1} = \frac{w_i^t \exp\left(\eta x_i^t / \mathbf{w}^t \cdot \mathbf{x}^t\right)}{\sum_{j=1}^{N} w_j^t \exp\left(\eta x_j^t / \mathbf{w}^t \cdot \mathbf{x}^t\right)} \;\; . \tag{4}$$

A similar update for the case of linear regression was first given by Kivinen and Warmuth [20].

In addition to the updates, we also need to choose an initial portfolio vector $\mathbf{w}^1$. When no prior information is given, a reasonable choice would be to start with an equal weight assigned to each of the stocks in the portfolio, that is, $\mathbf{w}^1 = (1/N, \ldots, 1/N)$. When side information is presented, we employ a set of portfolio vectors. We use the EG($\eta$) update to change the portfolio vector indexed by the side information. Hence, the problem of portfolio selection with side information simply reduces to a parallel selection of $K$ different portfolios. If the side information is indeed informative, the set of portfolios will achieve larger wealth than a sequence of portfolio vectors resulting from the entire sequence. We demonstrate this in the experimental section that follows.

In the next section we analyze our $EG(\eta)$-update based portfolio selection algorithm. We compare the performance of the EG($\eta$)-update as well as the Exact EG($\eta$)-update (which maximizes the $F$ in Equation 2 rather than the approximation $\hat{F}$) with other on-line portfolio selection algorithms for different settings in Section 5.

# 4 Analysis

In this section, we analyze the logarithmic wealth obtained by the EG($\eta$) portfolio update rule. We prove worst-case bounds on the update which imply that the EG($\eta$) update is almost as good as the best constant-rebalanced portfolio when certain assumptions hold on the relative volatility of the stocks in the portfolio. We also present a variant of EG($\eta$) which requires no such assumptions.

Although the analysis is presented for a single portfolio vector, it can be generalized to the multiple vectors kept when side information is present by partitioning the trading days based on the side information and treating each partition separately.

Since $x_i^t$ represents price relatives, we have that $x_i^t \geq 0$ for all $i$ and $t$. Furthermore, we assume that $\max_i x_i^t = 1$ for all $t$. We can make this assumption without loss of generality since multiplying the price relatives $\mathbf{x}^t$ by a constant $c$ simply adds $\log c$ to the logarithmic wealth, leaving the difference between the logarithmic wealth achieved by the EG($\eta$)-update and the best achieved logarithmic wealth $LS^\star$ unchanged. Put another way, the assumed lower bound $r$ on $x_i^t$ used in Theorem 1 (below) can be viewed as a lower bound on the ratio of the worst to best price relatives for trading day $t$.

To remind the reader, a portfolio vector is a vector of non-negative numbers that sum to 1. The EG($\eta$) portfolio update algorithm uses the following rule:

$$w_i^{t+1} = \frac{w_i^t \exp\left(\eta \frac{x_i^t}{\mathbf{w}^t \cdot \mathbf{x}^t}\right)}{Z_t}$$

where $\eta > 0$ is the learning rate, and $Z_t$ is the normalization

$$Z_t = \sum_{1 \leq i \leq N} w_i^t \exp\left(\eta \frac{x_i^t}{\mathbf{w}^t \cdot \mathbf{x}^t}\right) \ .$$

The following theorem characterizes a general property of the EG($\eta$)-update.

**Theorem 1** *Let $\mathbf{u} \in \mathbb{R}^N$ be a portfolio vector, and let $\mathbf{x}^1, \ldots, \mathbf{x}^T$ be a sequence of price relatives with $x_i^t \geq r > 0$ for all $i, t$ and $\max_i x_i^t = 1$ for all $t$. For $\eta > 0$ the logarithmic wealth due to the portfolio vectors produced by the EG($\eta$)-update is bounded from below as follows:*

$$\sum_{t=1}^{T} \log(\mathbf{w}^t \cdot \mathbf{x}^t) \geq \sum_{t=1}^{T} \log(\mathbf{u} \cdot \mathbf{x}^t) - \frac{D_{RE}(\mathbf{u}||\mathbf{w}^1)}{\eta} - \frac{\eta T}{8r^2}.$$

*Furthermore, if $\mathbf{w}^1$ is chosen to be the uniform proportion vector, and we set $\eta = 2r\sqrt{2\log N/T}$ then we have*

$$\sum_{t=1}^{T} \log(\mathbf{w}^t \cdot \mathbf{x}^t) \geq \sum_{t=1}^{T} \log(\mathbf{u} \cdot \mathbf{x}^t) - \frac{\sqrt{2T\log N}}{2r}.$$

**Proof** Let $\Delta_t = \mathrm{D}_{RE}(\mathbf{u}||\mathbf{w}^{t+1}) - \mathrm{D}_{RE}(\mathbf{u}||\mathbf{w}^t)$. Then

$$\begin{aligned}
\Delta_t &= -\sum_i u_i \log(w_i^{t+1}/w_i^t) \\
&= -\sum_i u_i (\eta x_i^t / \mathbf{w}^t \cdot \mathbf{x}^t - \log Z_t) \\
&= -\eta \frac{\mathbf{u} \cdot \mathbf{x}^t}{\mathbf{w}^t \cdot \mathbf{x}^t} + \log Z_t. \qquad (5)
\end{aligned}$$

To bound $\log Z_t$, we use the fact that for all $\alpha \in [0,1]$ and $x \in \mathbb{R}$,

$$\log(1 - \alpha(1 - e^x)) \leq \alpha x + x^2/8. \qquad (6)$$

This bound was verified by examining the first two derivatives of $f(x) = \alpha x + x^2/8 - \ln(1 - \alpha(1 - e^x))$ (see [15]).

Since $x_i^t \in [0,1]$ and since $\beta^x \leq 1-(1-\beta)x$ for $\beta > 0$ and $x \in [0,1]$, we have:

$$\begin{aligned} Z_t &= \sum_i w_i^t e^{\eta x_i^t / \mathbf{w}^t \cdot \mathbf{x}^t} \\ &\leq \sum_i w_i^t (1 - (1 - e^{\eta / \mathbf{w}^t \cdot \mathbf{x}^t}) x_i^t) \\ &= 1 - (1 - e^{\eta / \mathbf{w}^t \cdot \mathbf{x}^t}) \mathbf{w}^t \cdot \mathbf{x}^t. \end{aligned}$$

Now, applying inequality 6, we have

$$\log Z_t \leq \eta + \frac{\eta^2}{8(\mathbf{w}^t \cdot \mathbf{x}^t)^2}.$$

Combining with Equation (5) gives:

$$\begin{aligned} \Delta_t &\leq \eta \left(1 - \frac{\mathbf{u} \cdot \mathbf{x}^t}{\mathbf{w}^t \cdot \mathbf{x}^t}\right) + \frac{\eta^2}{8(\mathbf{w}^t \cdot \mathbf{x}^t)^2} \\ &\leq -\eta \log(\mathbf{u} \cdot \mathbf{x}^t / \mathbf{w}^t \cdot \mathbf{x}^t) + \frac{\eta^2}{8(\mathbf{w}^t \cdot \mathbf{x}^t)^2} \end{aligned}$$

since $1 - e^x \leq -x$ for all $x$.

Since $x_i^t \geq r$, and summing over all $t$, we have

$$\begin{aligned} -\mathrm{D}_{RE}(\mathbf{u}||\mathbf{w}^1) &\leq \mathrm{D}_{RE}(\mathbf{u}||\mathbf{w}^{T+1}) - \mathrm{D}_{RE}(\mathbf{u}||\mathbf{w}^1) \\ &\leq \eta \sum_{t=1}^{T} (\log(\mathbf{w}^t \cdot \mathbf{x}^t) - \log(\mathbf{u} \cdot \mathbf{x}^t)) + \frac{\eta^2 T}{8r^2}, \end{aligned}$$

which implies the first bound stated in the theorem. The second bound of the theorem follows by straightforward algebra, noting that $\mathrm{D}_{RE}(\mathbf{u}||\mathbf{w}^1) \leq \log N$ when $\mathbf{w}^1$ is the uniform probability vector. ■

Since

$$LS_T = \frac{1}{T} \sum_{t=1}^{T} \log\left(\mathbf{w}^t \cdot \mathbf{x}^t\right),$$

Theorem 1 immediately gives $LS_T^\star - LS_T \leq \sqrt{\log N/(2r^2T)}$ (under the conditions of Theorem 1). Thus, for an appropriate choice of $\eta$, when the number of days $T$ becomes large, the difference between the logarithmic wealth achieved by EG($\eta$) is guaranteed to converge to the logarithmic wealth of the best constant-rebalanced portfolio. However, Theorem 1 is not strong enough to show that EG($\eta$) is a universal portfolio algorithm. This is because choosing the proper $\eta$ requires knowledge of both the number of trading days and the ratio $r$ in advance. We will deal with both of these difficulties, starting with the dependence of $\eta$ on $r$.

When no lower bound $r$ on $x_i^t$ is known, we can use the following portfolio update algorithm which is parameterized by a real number $\alpha \in [0,1]$. Let

$$\tilde{\mathbf{x}}^t = (1 - \alpha/N)\mathbf{x}^t + (\alpha/N)\mathbf{1}$$

where $\mathbf{1}$ is the all 1's vector. As before, we maintain a portfolio vector $\mathbf{w}^t$ which is updated using $\tilde{\mathbf{x}}^t$ rather than $\mathbf{x}^t$:

$$w_i^{t+1} = \frac{w_i^t \exp(\eta \tilde{x}_i^t / \mathbf{w}^t \cdot \tilde{\mathbf{x}}^t)}{\sum_i w_i^t \exp(\eta \tilde{x}_i^t / \mathbf{w}^t \cdot \tilde{\mathbf{x}}^t)}.$$

Further, the portfolio vector that we invest with is also slightly modified. Specifically, the algorithm uses the portfolio vector

$$\tilde{\mathbf{w}}^t = (1-\alpha)\mathbf{w}^t + (\alpha/N)\mathbf{1}$$

and so the logarithmic wealth achieved is $\log(\tilde{\mathbf{w}}^t \cdot \mathbf{x}^t)$.

We call this modified algorithm $\widetilde{\mathrm{EG}}(\alpha, \eta)$.

**Theorem 2** *Let $\mathbf{u} \in \mathbb{R}^N$ be a portfolio vector, and let $\mathbf{x}^1, \ldots, \mathbf{x}^T$ be a sequence of price relatives with $x_i^t \geq 0$ for all $i, t$ and $\max_i x_i^t = 1$ for all $t$. For $\alpha \in (0, 1/2]$ and $\eta > 0$, the logarithmic wealth due to the portfolio vectors produced by the $\widetilde{EG}(\alpha, \eta)$-update is bounded from below as follows:*

$$\sum_{t=1}^{T} \log(\tilde{\mathbf{w}}^t \cdot \mathbf{x}^t) \geq \sum_{t=1}^{T} \log(\mathbf{u} \cdot \mathbf{x}^t) - 2\alpha T - \frac{D_{RE}(\mathbf{u}||\mathbf{w}^1)}{\eta} - \frac{\eta T}{8(\alpha/N)^2}.$$

*Furthermore, if $\mathbf{w}^1$ is chosen to be the uniform proportion vector, $T \geq 2N^2 \log N$, and we set $\alpha = \left(N^2 \log N/(8T)\right)^{1/4}$ and $\eta = \sqrt{8\alpha^2 \log N/(N^2 T)}$ then we have*

$$\sum_{t=1}^{T} \log(\tilde{\mathbf{w}}^t \cdot \mathbf{x}^t) \geq \sum_{t=1}^{T} \log(\mathbf{u} \cdot \mathbf{x}^t) - 2(2N^2 \log N)^{1/4} \cdot T^{3/4}. \tag{7}$$

**Proof** From our assumption that $\max_i x_i^t = 1$, we have

$$\frac{\tilde{\mathbf{w}}^t \cdot \mathbf{x}^t}{\mathbf{w}^t \cdot \tilde{\mathbf{x}}^t} \geq \frac{(1-\alpha)\mathbf{w}^t \cdot \mathbf{x}^t + \alpha/N}{(1-\alpha/N)\mathbf{w}^t \cdot \mathbf{x}^t + \alpha/N}.$$

The right hand side of this inequality is decreasing as a function of $\mathbf{w}^t \cdot \mathbf{x}^t$ and so is minimized when $\mathbf{w}^t \cdot \mathbf{x}^t = 1$. Thus,

$$\frac{\tilde{\mathbf{w}}^t \cdot \mathbf{x}^t}{\mathbf{w}^t \cdot \tilde{\mathbf{x}}^t} \geq (1-\alpha) + \alpha/N,$$

or equivalently,

$$\begin{aligned} \log(\tilde{\mathbf{w}}^t \cdot \mathbf{x}^t) &\geq \log(\mathbf{w}^t \cdot \tilde{\mathbf{x}}^t) + \log(1 - \alpha + \alpha/N) \\ &\geq \log(\mathbf{w}^t \cdot \tilde{\mathbf{x}}^t) - 2\alpha. \end{aligned} \tag{8}$$

The last inequality uses $\log(1-\alpha+\alpha/N) \geq \log(1-\alpha) \geq -2\alpha$ for $\alpha \in (0, 1/2]$.

From Theorem 1 applied to the price relative instances $\tilde{\mathbf{x}}^t$, we have that

$$\sum_{t=1}^{T} \log(\mathbf{w}^t \cdot \tilde{\mathbf{x}}^t) \geq \sum_{t=1}^{T} \log(\mathbf{u} \cdot \tilde{\mathbf{x}}^t) - \frac{\mathrm{D}_{RE}(\mathbf{u}||\mathbf{w}^1)}{\eta} - \frac{\eta T}{8(\alpha/N)^2} \tag{9}$$

where we used the fact that $\tilde{x}_i^t \geq \alpha/N$.

Note that $\mathbf{u} \cdot \tilde{\mathbf{x}}^t = (1-\alpha/N)\mathbf{u} \cdot \mathbf{x}^t + \alpha/N \geq \mathbf{u} \cdot \mathbf{x}^t$. Combined with Equations (8) and (9), and summing over all $t$, this gives the first bound of the theorem. The second bound of the theorem follows from the fact that $\mathrm{D}_{RE}(\mathbf{u}||\mathbf{w}^1) \leq \log N$ when $\mathbf{w}^1$ is the uniform probability vector. ■

Dividing inequality (7) of Theorem 2 by the number of trading days $T$ shows that the logarithmic wealth achieved by the $\widetilde{\mathrm{EG}}(\alpha, \eta)$-update converges to that of the best constant-rebalanced portfolio (for an appropriate choice of $\eta$ dependent on $T$). However, we still have the issue that the learning

rate must be chosen in advance as a function of $T$. The following algorithm and corollary shows how a doubling trick can be used to obtain a universal portfolio algorithm.

The staged $\widetilde{\mathrm{EG}}(\alpha,\eta)$-update runs in stages which are numbered from 0. The number of days in stage 0 is $2N^2\log N$, and the number of days in each stage $i > 0$ is $2^iN^2\log N$. Thus if $T > 2N^2\log N$ is the total number of days, the last stage entered is numbered $\lceil\log_2(\frac{T}{2N^2\log N})\rceil$. At the start of each stage the portfolio vector is re-initialized to the uniform proportion vector and $\alpha$ and $\eta$ are set as in Theorem 2 using the number of days in the stage as the value for $T$.

**Corollary 3** *The staged $\widetilde{EG}(\alpha,\eta)$-update is a universal portfolio selection algorithm.*

**Proof** We first bound the difference $\sum_{t=1}^{T}\log(\mathbf{u}\cdot\mathbf{x}^t) - \sum_{t=1}^{T}\log(\tilde{\mathbf{w}}^t\cdot\mathbf{x}^t)$ for any $\mathbf{u}$, any sequence of price relatives $\{\mathbf{x}^t\}$ and the $\tilde{\mathbf{w}}^t$ computed by the staged $\widetilde{\mathrm{EG}}(\alpha,\eta)$-update. Let $b = \lceil\log_2(\frac{T}{2N^2\log N})\rceil$ be a bound on the last stage number. From Theorem 2 we obtain

$$\begin{aligned}
\sum_{t=1}^{T}\log(\mathbf{u}\cdot\mathbf{x}^t) - \sum_{t=1}^{T}\log(\tilde{\mathbf{w}}^t\cdot\mathbf{x}^t) &\leq 4N^2\log N + \sum_{i=1}^{b} 2^{1/4}(2^i)^{3/4}2N^2\log N \\
&\leq 2^{5/4}N^2\log N(1+\sum_{i=0}^{b}(2^{3/4})^i) \\
&= 2^{5/4}N^2\log N(1+\frac{(2^{3/4})^{b+1}-1}{2^{3/4}-1}) \\
&\leq 6N^2\log N(1+(2^{3/4})^b) \\
&\leq 6N^2\log N(1+(\frac{T}{2N^2\log N})^{3/4}) \ .
\end{aligned}$$

Now, setting $\mathbf{u}$ to the best constant-rebalanced portfolio and dividing by $T$ allows us to rewrite the previous line in terms of the normalized logarithms of the wealth achieved

$$LS^\star(\{\mathbf{x}^t\}) - LS(\{\tilde{\mathbf{w}}^t\},\{\mathbf{x}^t\}) \leq \frac{6N^2\log N(1+(\frac{T}{2N^2\log N})^{3/4})}{T} \ .$$

As $T\to\infty$, the above bound goes to 0, completing the proof. ■

In sum, the difference between the average daily logarithmic increase in wealth of the $\widetilde{\mathrm{EG}}(\alpha,\eta)$-update and the best constant-rebalanced portfolio drops to zero at the rate $O(((N^2\log N)/T)^{1/4})$ for $T \geq 2N^2\log N$. When the ratio between the best and worst stock on each day is bounded and relatively small (as can often be expected in practice), the $\mathrm{EG}(\eta)$-update can be used instead giving a convergence rate to zero of $O(\sqrt{(\log N)/T})$. In comparison, the bounds proved by Cover and Ordentlich [11] for their algorithm converge to zero at the rate $O((N\log T)/T)$. In terms of the number of trading days $T$, their bounds are much superior, especially compared to our bound for $\widetilde{\mathrm{EG}}(\alpha,\eta)$. The only case in which our bounds have an advantage is when the number of stocks $N$ included in the portfolio is relatively large and the market has bounded relative volatility so that $\mathrm{EG}(\eta)$ can be used. Despite the comparative inferiority of our theoretical bounds, in our experiments, we found that our algorithm did better, even though the number of trading days $T$ was large (over 5,000) and the portfolios included only a few stocks.

# 5 Experiments with NYSE data

We tested our update rules on historical stock market data from the New York Stock Exchange accumulated over a 22-year period. For each experiment, we restricted our attention to a subset

of the 36 stocks for which we have data,[2] and compared the EG($\eta$)-update with the best single stock in the subset as well as the best constant-rebalanced portfolio for the subset. We found the best constant-rebalanced portfolio by applying a batch maximum-likelihood mixture estimation procedure as described in our earlier paper [15]. After determining the best constant-rebalanced portfolio, we then computed its performance on the price relative sequence. We also compared the performance of our update rules to that of Cover's universal portfolio algorithm and to an update that is based on exact maximization of Equation (2). We term the latter method the Exact EG($\eta$)-update. We compared the results for all subsets of stocks considered by Cover [9] in his experiments.

Summarizing our results, we found that, perhaps surprisingly, the wealth achieved by the EG($\eta$)-update was larger than the wealth achieved by the universal portfolio algorithm. This outcome is contrary to the superior worst-case bounds proved for the universal portfolio algorithm. The difference in performance was largest for portfolios composed of volatile stocks.

Furthermore, the universal portfolio strategy using the Dirichlet$(1,\ldots,1)$ prior outperformed the Dirichlet$(1/2,\ldots,1/2)$ prior, despite the better bounds proved by Cover and Ordentlich [11] for the Dirichlet$(1/2,\ldots,1/2)$ prior. We therefore used the Dirichlet$(1,\ldots,1)$ prior when generating all of the universal portfolio results reported in this section.

We did not find any significant difference in the performance achieved by the EG($\eta$)-update algorithm and the Exact EG($\eta$) algorithm while the latter was much slower.

We have two possible explanations why those algorithms with better analytical bounds performed worse in our experiments. First, the analytical bounds involve approximations, and a refined analysis might yield better bounds on some (or all) of the algorithms. Second, the analytical bounds are guarantees that hold for all sequences of price relatives. Therefore, the algorithms with better bounds might be hedging against unusual sequences of price relatives at the expense of their performance on the sequences of price relatives occurring in the historical market data.

The first example given by Cover is a portfolio based on Iroquois Brands Ltd. and Kin Ark Corp., two NYSE stocks chosen for their volatility. During the 22-year period ending in 1985, Iroquois increased in price by a factor of 8.92, while Kin Ark increased in price by a factor of 4.13. The best constant-rebalanced portfolio achieves a factor of 73.70 and the universal portfolio a wealth of 39.97. Using the EG($\eta$)-update with $\eta = 0.05$ yields a factor of 70.85, which is almost as good as the best constant-rebalanced portfolio. The Exact EG($\eta$)-update yields a similar factor of 70.91.

The results of the wealth achieved over the 22 years are given for this pair of stocks, as well as other pairs, in Table 1. We observed qualitatively similar results for the different portfolios considered by Cover [9]: Commercial Metals and Kin Ark (see also Figure 1), Commercial Metals and Meicco Corp., and IBM and Coca Cola (Figure 2). In all cases, the wealth achieved by EG($\eta$) and the Exact EG($\eta$) is larger than the wealth of the universal portfolio algorithm. Moreover, in several cases the wealth of the EG($\eta$)-update is almost as good as the wealth of the best constant-rebalanced portfolio. In all the experiments we performed, the yields of the EG($\eta$)-update and the Exact EG($\eta$)-update were very similar, and always within one percent of one another.

Three examples comparing the daily wealth achieved by EG($\eta$)-update and the best constant rebalanced portfolio are depicted in Figures 1–3. Note that when the stocks considered are not volatile and show a lock-step performance, as in the case of IBM and Coca Cola (see the volatility

[2]The set of stocks from which we built the various portfolios consisted of the following stocks: AHP, Alcoa, American Brands, Arco, Coca Cola, Commercial Metals, Dow Chemicals, Dupont, Espey Manufacturing, Exxon, Fischbach, Ford, GE, GM, GTE, Gulf, HP, IBM, Ingersoll, Iroquois, JNJ, Kimberly-Clark, Kin Ark, Kodak, Lukens, Meicco, Merck, MMM, Mobil, Morris Mining, P&G, Pillsbury, Schlum, Sears, Sherman Williams, and Texaco.

| Stocks | | Iroquois & Kin Ark | Comm. Met. & Kin Ark | Comm. Met. & Meicco | IBM & Coca Cola |
|---|---|---|---|---|---|
| Volatility – Stock 1 | | 0.034 | 0.025 | 0.025 | 0.013 |
| Volatility – Stock 2 | | 0.050 | 0.050 | 0.031 | 0.014 |
| Correlation of return | | 0.041 | 0.064 | 0.067 | 0.388 |
| Best Stock | - wealth | 8.9 | 52.0 | 52.0 | 13.4 |
| | - APY | 10.4 | 19.7 | 19.7 | 12.5 |
| BCRP | - wealth | 73.7 | 144.0 | 103.0 | 15.1 |
| | - APY | 21.6 | 25.3 | 23.4 | 13.1 |
| EG($\eta$) | - wealth | **70.9** | **110.2** | 97.9 | 14.9 |
| | - APY | 21.4 | 23.8 | 23.2 | 13.1 |
| | - /BCRP | 0.96 | 0.77 | 0.95 | 0.99 |
| Exact EG($\eta$) | - wealth | 70.4 | 110.1 | **98.4** | **15.0** |
| | - APY | 21.3 | 23.8 | 23.2 | 13.1 |
| | - /BCRP | 0.96 | 0.76 | 0.95 | 0.99 |
| Universal | - wealth | 40.0 | 78.4 | 74.0 | 14.2 |
| | - APY | 18.3 | 21.9 | 21.6 | 12.8 |
| | - /BCRP | 0.54 | 0.56 | 0.72 | 0.94 |

Table 1: Comparison of the wealth achieved by the EG($\eta$)-update, the Exact EG($\eta$)-update, and the universal portfolio algorithm. For all the portfolios considered, we give both the total wealth and the average annual percent yield (APY) for each portfolio-selection algorithm as well as for the best constituent stock in the portfolio and the best constant-rebalanced portfolio (BCRP) computed in hindsight from the entire price relatives sequence. We also report wealth as a fraction of that achieved by the BCRP (denoted "/BCRP" in the table). The first rows of the table report the volatility of the constituent stocks and the correlation between their returns.

and correlation information in Table 1), the wealth achieved by the universal portfolios and the EG($\eta$)-update as well as the best constant-rebalanced portfolio barely outperform the individual stocks.

The Gulf, HP and Schlum three-stock portfolio plotted in Figure 3 exhibits some interesting behavior. Schlum skyrockets between days 4000 and 4500, enabling it to outperform the 22-year BCRP at that point. Schlum later declines in value, and both the 22-year BCRP and the EG($\eta$)-update outperform Schlum over the entire period. If the experiment ended around day 4500, then the EG($\eta$)-update would yield less wealth than simply investing in Schlum. Indeed, the BCRP for 4500 trading days is invested wholly in Schlum, while the BRCP for the entire 22-year period invest 44% in Gulf, 34% in HP and 22% in Schlum.

For the Iroquois/Kin Ark portfolio, Figure 4 shows the percentage of the wealth invested in Iroquois for both the EG($\eta$)-update and the universal portfolio algorithm. Although both algorithms tended to keep similar portfolios, the universal portfolio algorithm changed its investment proportions more aggressively when Kin Ark rose around days 300–500.

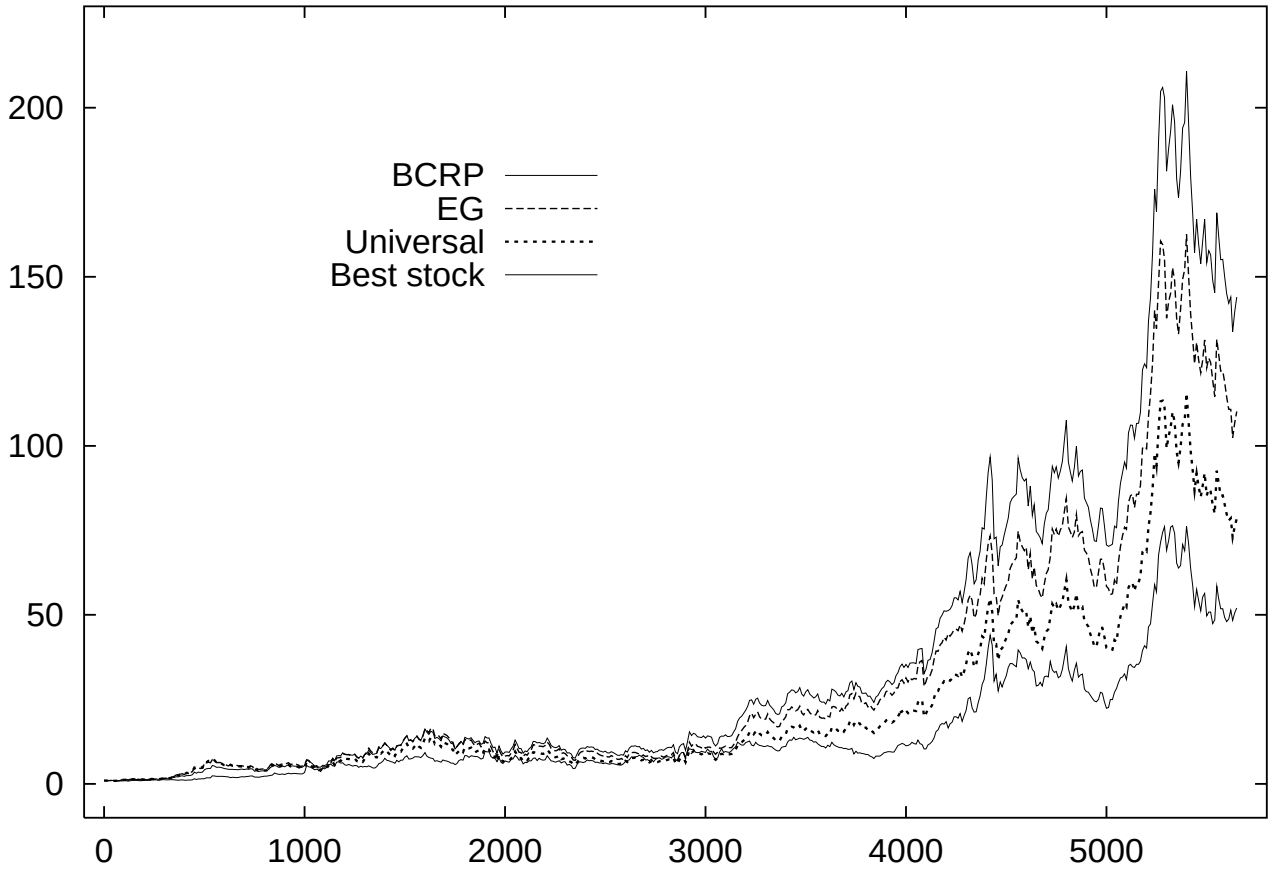


Figure 1: Comparison of wealths achieved by the best constant-rebalanced portfolio, the EG($\eta$)-update, the universal portfolio algorithm and the best stock for a portfolio consisting of Commercial Metals and Kin Ark.

### 5.1 Margin loans

Following Cover [9], we also tested the case where the portfolio can buy stocks on margin. This case can be modeled by adding an additional "margin component" for each stock to the vector of price relatives. We assumed that all margin purchases were made 50% down and with a 50% loan. Thus, the margin price relative for a stock $i$ on day $t$ is $2x_i^t - 1 - c$ where $c$ is the daily interest rate (recall that $x_i^t$ is the price relative of stock $i$). We tested this case with $c = 0.000233$ which corresponds to an annual interest rate of 6%. The results are given in Table 2. It is clear from the table that the four-investment portfolio containing the same two stocks plus "buying on the margin" results in a greater wealth. The efficiency of our update rule enables us to test our updates on more than two stocks. Moreover, as shown by the analysis, the wealth "lost" by our algorithms compared to the best constant-rebalanced portfolio scales like $O(\sqrt{N})$, whereas for the bounds on Cover's universal portfolio algorithms, the loss in wealth is linear in the number of investment options $N$. Thus, our algorithm is more likely to tolerate additional investment options, such as buying on margin.

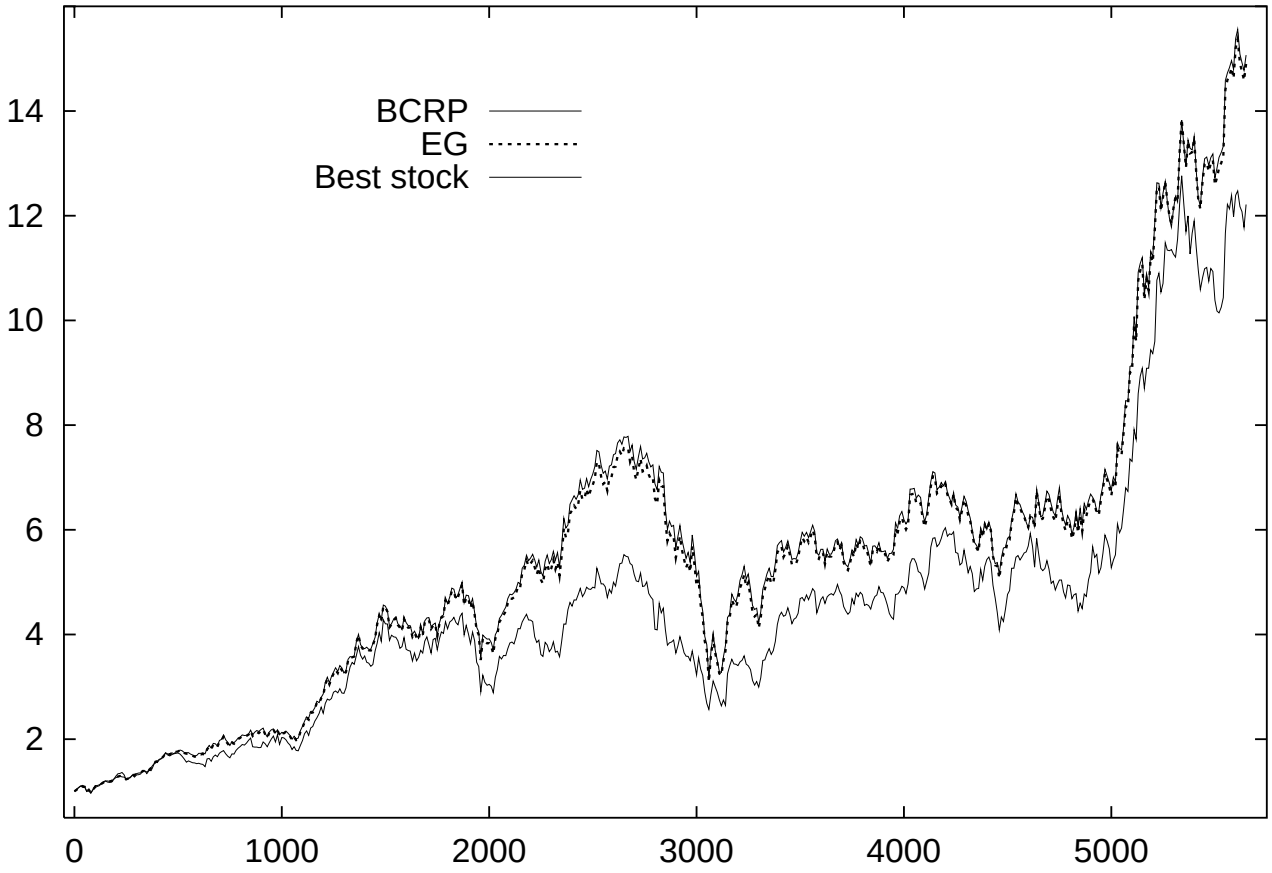


Figure 2: Comparison of wealths achieved by the best constant-rebalanced portfolio, the EG($\eta$)-update and the best stock for a portfolio consisting of IBM and Coca Cola.

### 5.2 Learning rate

We found that learning rates around $\eta = 0.05$ are good choices for the EG($\eta$)-update. If the starting portfolio is far from the best constant-rebalanced portfolio and EG uses a very low learning rate, then it will perform poorly because it does not move away from the original portfolio fast enough. On the other hand, a high learning rate can also be bad as it causes EG to be misled by day-to-day fluctuations. For the 2-stock example given in the introduction, the EG($\eta$)-update actually loses money when the learning rate is above one.

Within some middle range, it turns out that the performance of EG is not overly sensitive to the particular choice of learning rate $\eta$. Learning rates from 0.01 to 0.15 all achieved great wealth, greater than the wealth achieved by the universal portfolio algorithm and in many cases comparable to the wealth achieved by the constant-rebalanced portfolio. The wealth achieved for different learning rates for the four-investment portfolio discussed above (two stocks plus margin) are given in Table 3.

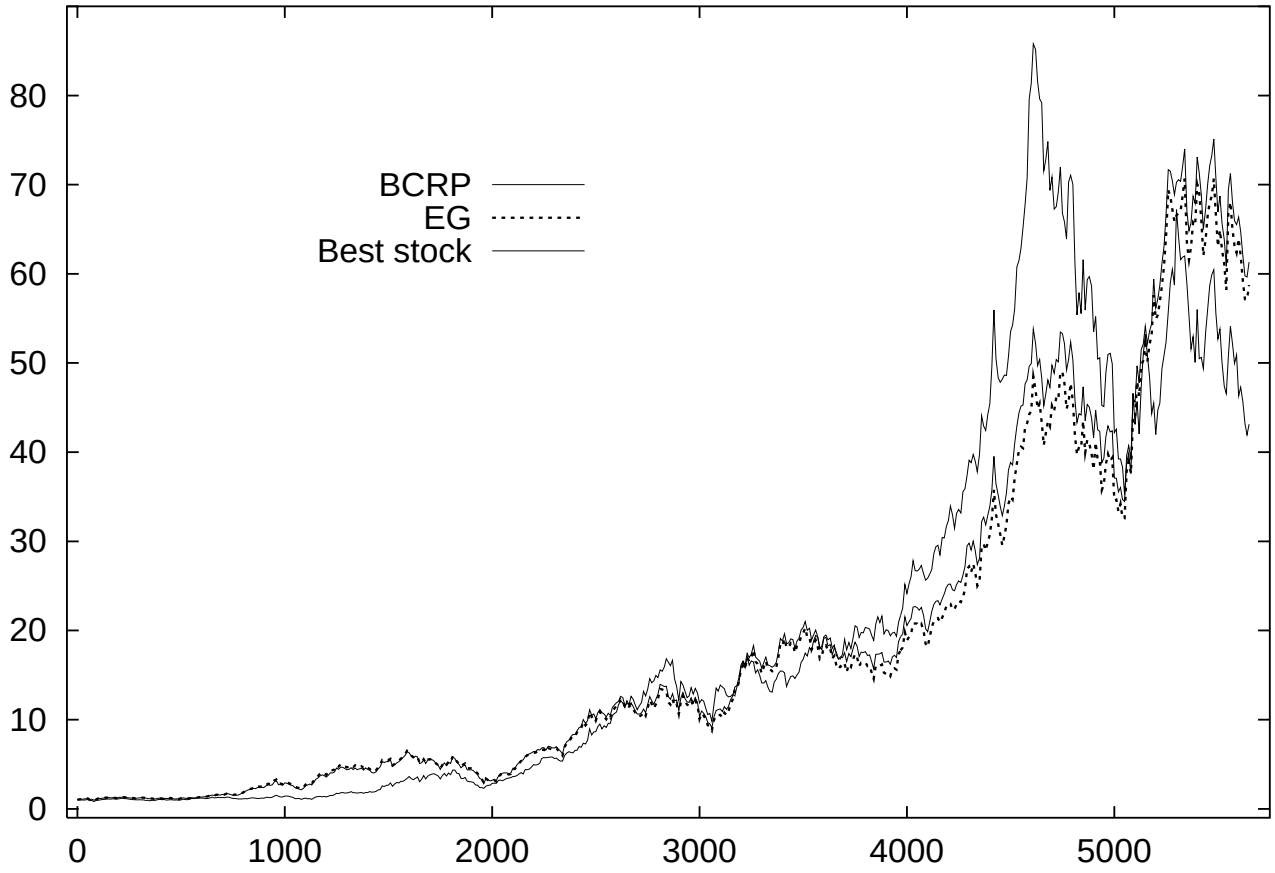


Figure 3: Comparison of wealths achieved by the best constant-rebalanced portfolio, the EG($\eta$)-update, and the best stock for a portfolio consisting of Gulf, HP and Schlum.

### 5.3 Side information

We also tested the performance of our portfolio update algorithm when side information is presented. For these experiments, the side information was utilized using the method described in Section 2.4 of partitioning the sequence into subsequences based on the value of the side information. There are many possible forms of side information on which these algorithms might be tested. In our experiments, we chose to define the side information value to be the index of the stock with the best growth of wealth on the last 100 trading days — information that would certainly be available to a investor in a real trading situation. Thus, the possible set of values for the side information is $\{1, \ldots, K\}$ where $K = N$.

The results are summarized in Table 4. It is evident from the examples given in the table that using the side information (i.e., keeping $N$ portfolio vectors) results in a significant improvement in the wealth achieved, even when using such simple and readily available side information. However, the gap between the best side information dependent constant-rebalanced portfolio and the wealth achieved by the EG($\eta$)-update with side information is now much larger. One of the reasons is that we used the same learning rate regardless of the side information value. Large learning rates cause the update algorithms to quickly approach the best constant-rebalanced portfolio, but make it difficult for the algorithm to reach this portfolio exactly. On the other hand, small learning

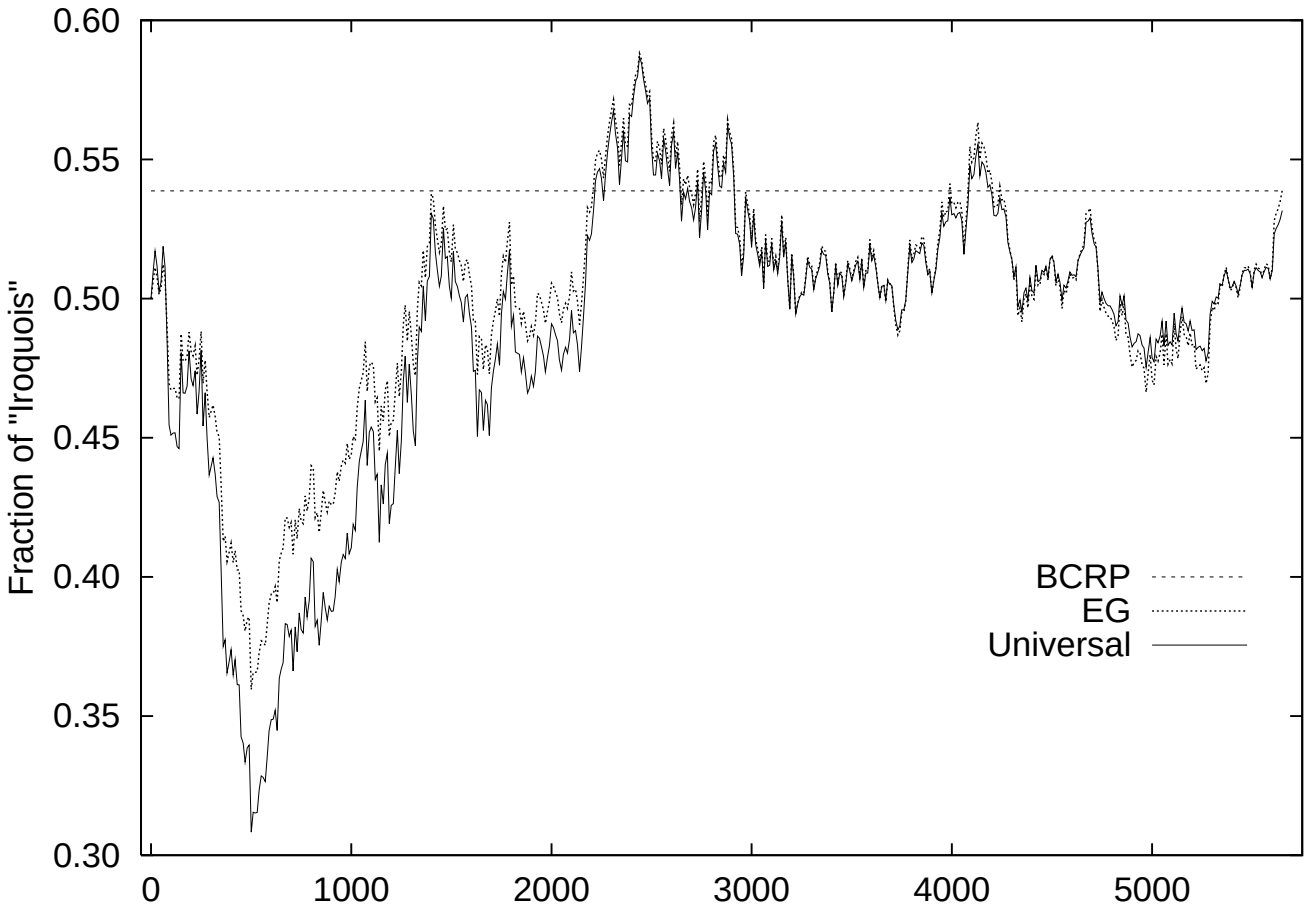


Figure 4: The fraction of the wealth invested in Iroquois over time by the universal portfolio algorithm and the EG($\eta$)-update for a portfolio consisting of Iroquois and Kin Ark.

rates aid convergence to the best constant-rebalanced portfolio, but may cause the algorithm to spend a long time far away from this value. Therefore, when the side information splits the number of trading days unevenly, different learning rates for the different side information values may be required.

### 5.4 Portfolios with more than two stocks

The theoretical bound on the wealth achieved by the EG($\eta$)-update suggests that the EG($\eta$)-update algorithm would scale better than Cover's universal portfolio algorithm as the number of stocks or assets in the portfolio grows. Furthermore, the time required by the EG($\eta$)-update to modify the portfolio after each trading day is constant per stock while the time complexity of the universal portfolio algorithm grows exponentially fast with the number of stocks in the portfolio. These two key factors suggest that the EG($\eta$)-update might be a viable alternative as a provably competitive portfolio selection algorithm.

To empirically verify that the above advantages hold in practice, we performed experiments with portfolios of size $N = 2$ to $N = 24$ from the New York Stock Exchange. We compared the wealth and the total time required to update the portfolios through the entire 22-year period.

| | Without loans | With margin loans |
|---|---|---|
| Commercial Metals | 52.0 | 19.7 |
| Kin Ark | 4.1 | 0.0 |
| BCRP | 144.0 | 262.4 |
| Universal portfolio | 78.5 | 98.4 |
| EG($\eta$) | 110.2 | 121.9 |
| Exact EG($\eta$) | 110.1 | 122.0 |

Table 2: Comparison of the portfolio selection algorithms when margin loans for each stock are available.

| | | EG | | | | | |
|---|---|---|---|---|---|---|---|
| BCRP | Universal | $\eta = 0.01$ | $\eta = 0.02$ | $\eta = 0.05$ | $\eta = 0.10$ | $\eta = 0.15$ | $\eta = 0.20$ |
| 262.4 | 98.4 | 119.9 | 121.4 | 121.9 | 113.3 | 103.1 | 91.3 |

Table 3: Comparison of the wealth achieved by the EG($\eta$)-update and for various learning rates and the universal portfolio algorithm, for the stocks considered in Table 2.

To compute updates for the universal portfolio algorithm, we used a sampling technique as described by Cover [9]. For portfolios of size $N \leq 9$, we approximated Equation (1) by a finite sum taken over all possible portfolio vectors $\mathbf{w}$ of the form $(i_1/r, i_2/r, \ldots, i_N/r)$ where each $i_j$ is a nonnegative integer and $i_1 + \cdots + i_N = r$. In our experiments, we used $r = 10$. This sampling technique quickly becomes infeasible: for a portfolio consisting of 9 stocks, it takes about 9.5 hours to calculate the universal portfolio updates over the 22-year trading period. The rapid infeasibility of the universal portfolio algorithm can also be seen in Figure 5 which shows the time required to compute the universal algorithm compared to EG($\eta$).

To handle this difficulty, for portfolios consisting of more than 9 stocks, we instead approximated Equation (1) using a large number ($10^8$) of randomly chosen vectors $\mathbf{w}$, each selected independently and uniformly from the space of all probability vectors. Because this is only an approximate method, the yields of the universal portfolio might be underestimated due to sparse sampling artifacts which are unavoidable given the time complexity of the algorithm.

Figure 5 also gives computation times for the Exact EG($\eta$)-update. Clearly, as noted earlier, this method is significantly slower than EG($\eta$) while achieving almost identical returns.

To test performance of the portfolio selection algorithm on larger portfolios, we first created five portfolios consisting of 12 or 24 stocks each which were chosen according to volatility. To be specific, we ranked the 36 stocks based on their volatility as measured by the standard deviation of the logarithm of the price relatives (see, for instance, Hull [17]). We then divided the 36 stocks into three portfolios: the first consisted of the 12 stocks with lowest volatility; the next consisted of the 12 stocks with highest volatility; and the third consisted of the remaining 12 stocks of medium volatility. We also tested on two larger portfolios consisting of the 24 stocks of lowest volatility and the 24 stocks of highest volatility.

The results for these five portfolios are given in Table 5. This table also shows the volatility of each of the portfolio selection algorithms. These results indicate that these methods produce more wealth when more volatile stocks are used. At the same time, EG($\eta$) and the universal portfolio

| Stocks | Without Side Information | | | With Side Information | | |
|---|---|---|---|---|---|---|
| | BCRP | EG($\eta$) | Univ. | BCRP | EG($\eta$) | Univ. |
| Iroq. & Kin Ark | 73.7 | 70.9 | 40.0 | 307.9 | 99.4 | 86.6 |
| Com. & Kin Ark | 144.0 | 110.2 | 80.5 | 451.3 | 257.2 | 115.7 |
| Com. & Meicco | 103.0 | 97.9 | 74.1 | 436.2 | 186.1 | 110.9 |
| IBM & Coke | 15.1 | 14.9 | 14.2 | 118.5 | 89.9 | 21.1 |

Table 4: Comparison of the wealth achieved by the best constant-rebalanced portfolio (BCRP) and the EG($\eta$)-update when no side information is provided and when side information about the best stock in the last 100 trading days is presented. We have used the same learning rate ($\eta = 0.05$) for both cases.

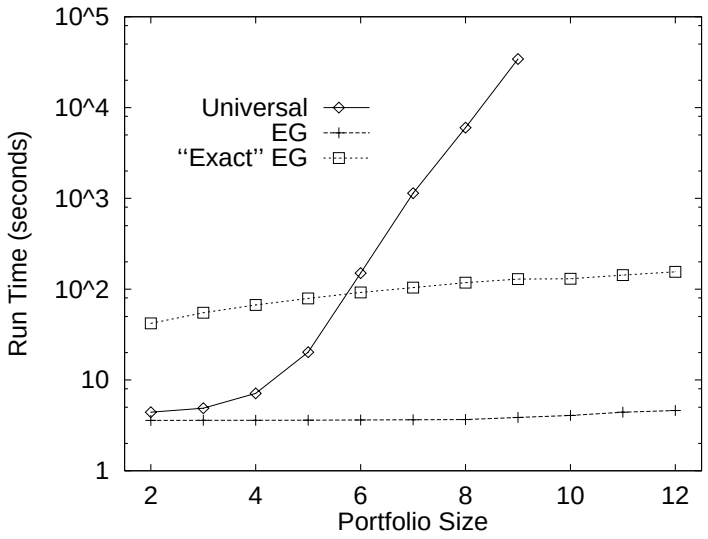


Figure 5: The running times of the universal, EG($\eta$), and Exact EG($\eta$) portfolio selection algorithms as a function of the number of stocks in the portfolio. All experiments were conducted on an SGI MIPS R10000 running at 195 MHz.

algorithms are less volatile than any of the constituent stocks.

Next, we tested the performance of the algorithms as a function of the number of stocks in a portfolio consisting of randomly selected stocks. For each portfolio size $N$, we randomly picked 100 different subsets (out of the $\binom{36}{N}$ possible subsets) and ran the universal portfolio algorithm and the EG($\eta$) update. We then calculated the geometric mean of the wealths obtained over the 100 different subsets of size $N$ and plotted the results in Figure 6 in annual percent yields. Also, for each subset of stocks, we calculated the wealth achieved by the portfolio selection algorithms as a fraction of the wealth attained by the best constant rebalanced portfolio. The geometric average of these wealth ratios is plotted in Figure 7.

It is clear from the figures that the wealth achieved by the EG($\eta$)-update is consistently higher than that of the universal portfolio algorithm, and the discrepancy grows as the size of the portfolio increases. However, as noted earlier, the significant degradation in performance of the universal portfolio algorithm when $N$ gets larger might be influenced by the sampling-based approximation technique which was used for $N > 9$; in fact, this effect seemed so great for $N > 12$, that we did

| Portfolio | Stock volatility min | avg | max | BCRP wealth | EG($\eta$) wealth | vol. | Universal wealth | vol. |
|---|---|---|---|---|---|---|---|---|
| Low 12 | 0.0115 | 0.0132 | 0.0141 | 16.6 | 15.6 | 0.0049 | 12.4 | 0.0083 |
| Med 12 | 0.0142 | 0.0153 | 0.0170 | 54.1 | 31.6 | 0.0061 | 16.9 | 0.0088 |
| High 12 | 0.0171 | 0.0260 | 0.0498 | 174.3 | 150.2 | 0.0103 | 81.2 | 0.0117 |
| Low 24 | 0.0115 | 0.0143 | 0.0170 | 54.1 | 17.1 | 0.0051 | | |
| High 24 | 0.0142 | 0.0206 | 0.0498 | 250.6 | 156.1 | 0.0090 | | |

Table 5: Comparison of the wealth achieved by the best constant-rebalanced portfolio (BCRP), the universal portfolio and the EG($\eta$)-update for portfolios of varying volatilities consisting of 12 or 24 stocks each. Minimum, average and maximum volatilities are given for the stocks in each portfolio, as well as the volatilities of universal and EG($\eta$) portfolios.

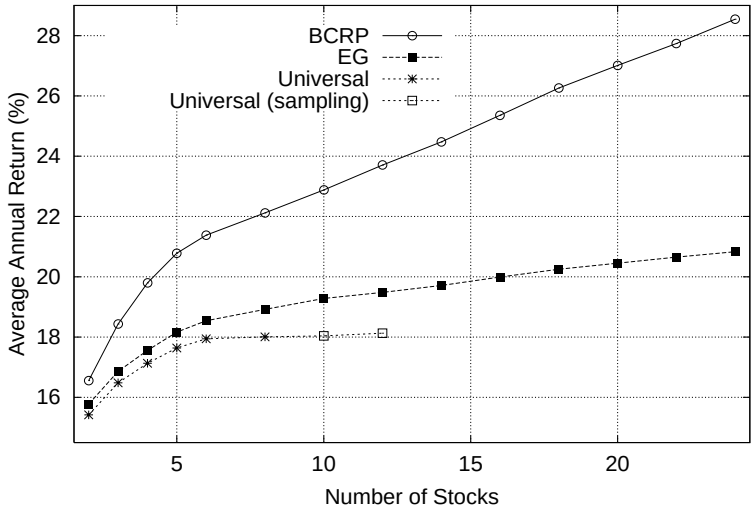


Figure 6: The average annual percent yield of BCRP, the universal portfolio algorithm and the EG($\eta$)-update for random subsets of stocks from size 2 to 24.

not plot any results in this higher range. On this data, it seems clear that the good performance and very low computation time required by the EG($\eta$)-update provides a viable alternative to the universal portfolio algorithm with provable competitiveness bounds.

As theoretical analysis implies, the gap between the average yield of the EG($\eta$)-update and the BCRP increases with the size of the portfolio. However, it is interesting to note that the average annual return of the EG($\eta$)-update also increases as the portfolio includes more stocks.

# 6 Discussion and future research

Although the experimental results presented in this paper are encouraging, we have ignored one important aspect of a real market: trading costs. Typically, there are two types of commissions imposed in a real market. In the first case, the investor needs to pay a percentage of the transaction to a broker. In this case, we can still write down a closed form expression for the wealth achieved

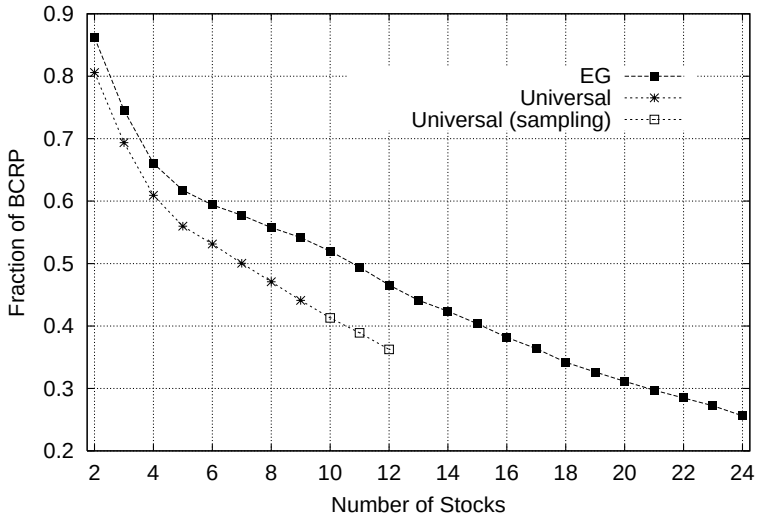


Figure 7: Comparison of the average wealth of the universal portfolio algorithm and the EG($\eta$)-update for random subsets of stocks from size 2 to 24. The results are given as fractions of the wealth of the BCRP.

at each time step while taking the trading costs into account. However, the wealth function we are trying to maximize becomes highly non-linear, and it is hard to derive an update rule. Cover's universal portfolio algorithm was recently extended and analyzed in this manner by Blum and Kalai [7].

The second type of commission is to pay a fixed amount per transaction, that is, per purchase or sale of a stock. Therefore, there might be days for which the wealth will be larger if no trading is performed, especially if the portfolio vector after the new trading day is close to the desired portfolio vector. We can define a semi-constant-rebalanced portfolio which is rebalanced only on a subset of the possible trading days. Now, in addition to the best constant-rebalanced portfolio, we need also to find the best subset of the sequence that results in the maximal wealth. We suspect that finding the best subset is computationally hard. Still, it is not clear whether finding a competitive approximation is hard as well.

This paper and most other work on investment strategies employ a tacit assumption that the market is stationary and seek a strategy that successfully competes against the best *single* constant-rebalanced portfolio. However, this assumption is far from being realistic. This suggests applying techniques developed for tracking a drifting concept [3, 16] to on-line portfolio selection in a changing market. This approach was recently explored by Singer [22].

There is also more theoretical work to be done in order to understand why EG($\eta$) seems to perform better than Cover's algorithm despite the clear theoretical superiority of his algorithm.

## Acknowledgments

Thanks to Tom Cover and Erik Ordentlich for providing us with the stock market data (originally generated by Hal Stern) used in our experiments. We are also grateful to Erik Ordentlich for a careful reading and helpful comments on an earlier draft.

## References


[1] P. H. Algoet. Universal schemes for prediction, gambling, and portfolio selection. *Annals of Probability*, 20:901–941, 1992.

[2] P. H. Algoet and T. M. Cover. Asymptotic optimality and asymptotic equipartition properties of log-optimum investment. *Annals of Probability*, 16(2):876–898, 1988.

[3] P. Auer and M. K. Warmuth. Tracking the best disjunction. In *36th Annual Symposium on Foundations of Computer Science*, 1995.

[4] A. Barron and T. M. Cover. A bound on the financial value of information. *IEEE Transactions on Information Theory*, 34:1097–1100, 1988.

[5] R. Bell and T. M. Cover. Competitive optimality of logarithmic investment. *Mathematics of Operations Research*, 5:161–166, 1980.

[6] R. Bell and T. M. Cover. Game-theoretic optimal portfolios. *Managment Science*, 34:724–733, 1988.

[7] Avrim Blum and Adam Kalai. Universal portfolios with and without transaction costs. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, pages 309–313. ACM Press, 1997.

[8] T. M. Cover. An algorithm for maximizing expected log investment return. *IEEE Transactions on Information Theory*, 30:369–373, 1984.

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

[10] T. M. Cover and D. Gluss. Empirical Bayes stock market portfolios. *Advances in Applied Mathematics*, 7, 1986.

[11] T.M. Cover and E. Ordentlich. Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2), 1996.

[12] R. Fletcher. *Practical methods of optimization.* Wiley, 1987.

[13] D. Haussler, N. Littlestone, and M. K. Warmuth. Predicting $\{0,1\}$-functions on randomly drawn points. *Information and Computation*, 115(2):284–293, 1994.

[14] D. P. Helmbold, J. Kivinen, and M. K. Warmuth. Worst-case loss bounds for sigmoided linear neurons. In *Advances in Neural Information Processing Systems 8*, 1996.

[15] D. P. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth. A comparison of new and old algorithms for a mixture estimation problem. *Machine Learning*, 27(1):97–119, 1997.

[16] M. Herbster and M. K. Warmuth. Tracking the best expert. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 286–294, 1995.

[17] J. C. Hull. *Options, futures, and other derivatives.* Prentice Hall, 1997.

[18] G. Jumarie. *Relative information.* Springer-Verlag, 1990.

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

[20] Jyrki Kivinen and Manfred K. Warmuth. Additive versus exponentiated gradient updates for linear prediction. *Information and Computation*, 132(1):1–64, January 1997.

[21] N. Littlestone. Redundant noisy attributes, attribute errors, and linear threshold learning using Winnow. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 147–156, 1991.

[22] Yoram Singer. Switching portfolios. *International Journal of Neural Systems*, to appear.