Can We Learn to Beat the Best Stock

Allan Borodin[1] **Ran El-Yaniv**[2] **Vincent Gogan**[1]
Department of Computer Science
University of Toronto[1] Technion - Israel Institute of Technology[2]
{bor,vincent}@cs.toronto.edu rani@cs.technion.ac.il

Abstract

A novel algorithm for actively trading stocks is presented. While traditional universal algorithms (and technical trading heuristics) attempt to predict winners or trends, our approach relies on predictable statistical relations between all pairs of stocks in the market. Our empirical results on historical markets provide strong evidence that this type of technical trading can "beat the market" and moreover, can beat the best stock in the market. In doing so we utilize a new idea for smoothing critical parameters in the context of expert learning.

1 Introduction: The Portfolio Selection Problem

The portfolio selection (PS) problem is a challenging problem for machine learning, online algorithms and, of course, computational finance. As is well known (e.g. see Lugosi [1]) sequence prediction under the log loss measure can be viewed as a special case of portfolio selection, and perhaps more surprisingly, from a certain worst case minimax criterion, portfolio selection is not essentially any harder (than prediction) as shown in [2] (see also [1], Thm. 20 & 21). But there seems to be a qualitative difference between the practical utility of "universal" sequence prediction and universal portfolio selection. Simply stated, universal sequence prediction algorithms under various probabilistic and worst-case models work very well in practice whereas the known universal portfolio selection algorithms do not seem to provide any substantial benefit over a naive investment strategy (see Sec. 4).

A major pragmatic question is whether or not a computer program can consistently outperform the market. A closer inspection of the interesting ideas developed in information theory and online learning suggests that a promising approach is to exploit the natural volatility in the market and in particular to benefit from simple and rather persistent statistical relations between stocks rather than to try to predict stock prices or "winners". We present a non-universal portfolio selection algorithm[1], which does not try to predict winners. The motivation behind our algorithm is the rationale behind *constant rebalancing* algorithms and the worst case study of universal trading introduced by Cover [3]. Not only does our proposed algorithm substantially "beat the market" on historical markets, it also beats the best stock. So why are we presenting this algorithm and not just simply making money? There are, of course some caveats and obstacles to utilizing the algorithm. But for large investors the possibility of a goose laying silver (if not golden) eggs is not impossible.

[1] Any PS algorithm can be modified to be universal by investing any fixed fraction of the initial wealth in a universal algorithm.

Assume a market with m stocks. Let $\mathbf{v}_t = (v_t(1), \ldots, v_t(m))$ be the closing prices of the m stocks for the t^{th} day, where $v_t(j)$ is the price of the jth stock. It is convenient to work with *relative prices* $x_t(j) = v_t(j)/v_{t-1}(j)$ so that an investment of \$$d$ in the jth stock just before the t^{th} period yields $dx_t(j)$ dollars. We let $\mathbf{x}_t = (x_t(1), \ldots, x_t(m))$ denote the *market vector* of relative prices corresponding to the t^{th} day. A *portfolio* $\mathbf{b}$ is an allocation of wealth in the stocks, specified by the proportions $\mathbf{b} = (b(1), \ldots, b(m))$ of current dollar wealth invested in each of the stocks, where $b(j) \geq 0$ and $\sum_j b(j) = 1$. The *daily return* of a portfolio $\mathbf{b}$ w.r.t. a market vector $\mathbf{x}$ is $\mathbf{b} \cdot \mathbf{x} = \sum_j b(j)x(j)$ and the (compound) *total return*, $\mathsf{ret}_X(\mathbf{b}_1, \ldots, \mathbf{b}_n)$, of a sequence of portfolios $\mathbf{b}_1, \ldots, \mathbf{b}_n$ w.r.t. a market sequence $X = \mathbf{x}_1, \ldots, \mathbf{x}_n$ is $\prod_{t=1}^{n} \mathbf{b}_t \cdot \mathbf{x}_t$. A portfolio selection algorithm is any deterministic or randomized rule for specifying a sequence of portfolios.

The simplest strategy is to "buy-and-hold" stocks using some portfolio $\mathbf{b}$. We denote this strategy by $\text{BAH}_\mathbf{b}$ and let U-BAH denote the uniform buy-and-hold when $\mathbf{b} = (1/m, \ldots, 1/m)$. We say that a portfolio selection algorithm "beats the market" when it outperforms U-BAH on a given market sequence although in practice "the market" can be represented by some non-uniform BAH (e.g. DJIA). Buy-and-hold strategies rely on the tendency of successful markets to grow. Much of modern portfolio theory focuses on how to choose a good $\mathbf{b}$ for the buy-and-hold strategy. The seminal ideas of Markowitz in [4] yield an algorithmic procedure for choosing the weights of the portfolio $\mathbf{b}$ so as to minimize the variance for any feasible expected return. This variance minimization is possible by placing appropriate larger weights on subsets of anti-correlated stocks, an idea which we shall also utilize. We denote the optimal *in hindsight* buy-and-hold strategy (i.e. invest only in the best stock) by BAH^*.

An alternative approach to the static buy-and-hold is to dynamically change the portfolio during the trading period. This approach is often called "active trading". One example of active trading is *constant rebalancing*; namely, fix a portfolio $\mathbf{b}$ and (re)invest your dollars *each day* according to $\mathbf{b}$. We denote this constant rebalancing strategy by $\text{CBAL}_\mathbf{b}$ and let CBAL^* denote the optimal (in hindsight) CBAL. A constant rebalancing strategy can often take advantage of market fluctuations to achieve a return significantly greater than that of BAH^*. CBAL^* is always at least as good as the best stock BAH^* and in some real market sequences a constant rebalancing strategy will take advantage of market fluctuations and significantly outperform the best stock (see Table 1). For now, consider Cover and Gluss' [5] classic (but contrived) example of a market consisting of cash and one stock and the market sequence of price relatives $\binom{1}{1/2}, \binom{1}{2}, \binom{1}{1/2}, \binom{1}{2}, \ldots$ Now consider the $\text{CBAL}_\mathbf{b}$ with $\mathbf{b} = (\frac{1}{2}, \frac{1}{2})$. On each odd day the daily return of $\text{CBAL}_\mathbf{b}$ is $\frac{1}{2}1 + \frac{1}{2}\frac{1}{2} = \frac{3}{4}$ and on each even day, it is $3/2$. The total return over n days is therefore $(9/8)^{n/2}$, illustrating how a constant rebalancing strategy can yield exponential returns in a "no-growth market". Under the assumption that the daily market vectors are observations of identically and independently distributed (i.i.d) random variables, it is shown in [6] that CBAL^* performs at least as good (in the sense of expected total return) as the best online portfolio selection algorithm. However, many studies (see e.g. [7]) argue that stock price sequences do have long term memory and are not i.i.d.

A non-traditional objective (in computational finance) is to develop online trading strategies that are in some sense *always guaranteed* to perform well. Within a line of research pioneered by Cover [5, 3, 2] one attempts to design portfolio selection algorithms that can provably do well (in terms of their total return) with respect to some online or offline benchmark algorithms. Two natural *online* benchmark algorithms are the uniform buy and hold U-BAH, and the uniform constant rebalancing strategy U-CBAL, which is $\text{CBAL}_\mathbf{b}$ with $\mathbf{b} = (\frac{1}{m}, \ldots, \frac{1}{m})$. A natural *offline* benchmark is BAH^* and a more challenging offline benchmark is CBAL^*.

Cover and Ordentlich's *Universal Portfolios* algorithm [3, 2], denoted here by UNIVERSAL,

was proven to be *universal* against CBAL*, in the sense that *for every market sequence* X of m stocks over n days, it guarantees a sub-exponential (indeed polynomial) ratio in n,

$$\mathsf{ret}_X(\text{CBAL}^*)/\mathsf{ret}_X(\text{UNIVERSAL}) \leq O\left(n^{\frac{m-1}{2}}\right) \tag{1}$$

From a theoretical perspective this is surprising as the ratio is a polynomial in n (for fixed m) whereas CBAL* is capable of exponential returns. From a practical perspective, while the ratio $n^{\frac{m-1}{2}}$ is not very useful, the motivation that underlies the potential of CBAL algorithms is useful! We follow this motivation and develop a new algorithm which we call ANTICOR. By attempting to systematically follow the constant rebalancing philosophy, ANTICOR is capable of some extraordinary performance in the absence of transaction costs, or even with very small transaction costs.

2 Trying to Learn the Winners

The most direct approach to expert learning and portfolio selection is a "(reward based) weighted average prediction" algorithm which adaptively computes a weighted average of experts by gradually increasing (by some multiplicative or additive update rule) the relative weights of the more successful experts. For example, in the context of the PS problem consider the "exponentiated gradient" EG(η) algorithm proposed by Helmbold *et al.* [8]. The EG(η) algorithm computes the next portfolio to be

$$\mathbf{b}_{t+1}(j) = \frac{\mathbf{b}_t(j)\exp\left\{\eta \mathbf{x}_t(j)/(\mathbf{b}_t \cdot \mathbf{x}_t)\right\}}{\sum_{j=1}^{m} \mathbf{b}_t(j)\exp\left\{\eta \mathbf{x}_t(j)/(\mathbf{b}_t \cdot \mathbf{x}_t)\right\}}$$

where η is a "learning rate" parameter. EG was designed to greedily choose the best portfolio for yesterday's market $\mathbf{x}_t$ while at the same time paying a penalty from moving far from yesterday's portfolio. For a universal bound on EG, Helmbold *et al.* set $\eta = 2x_{\min}\sqrt{2(\log m)/n}$ where $x_{\min}$ is a lower bound on any price relative.[2] It is easy to see that as n increases, η decreases to 0 so that we can think of η as being very small in order to achieve universality. When $\eta = 0$, the algorithm EG(η) degenerates to the uniform CBAL which is not a universal algorithm. It is also the case that if each day the price relatives for all stocks were identical, then EG (as well as other PS algorithms) will converge to the uniform CBAL. Combining a small learning rate with a "reasonably balanced" market we expect the performance of EG to be similar to that of the uniform CBAL and this is confirmed by our experiments (see Table1).[3]

Cover's universal algorithms adaptively learn each day's portfolio by increasing the weights of successful CBALs. The update rule for these universal algorithms is

$$\mathbf{b}_{t+1} = \frac{\int \mathbf{b} \cdot \mathsf{ret}_t(\text{CBAL}_{\mathbf{b}})d\mu(\mathbf{b})}{\int \mathsf{ret}_t(\text{CBAL}_{\mathbf{b}})d\mu(\mathbf{b})},$$

where $\mu(\cdot)$ is some prior distribution over portfolios. Thus, the weight of a possible portfolio is proportional to its total return $\mathsf{ret}_t(\mathbf{b})$ thus far times its prior. The particular universal algorithm we consider in our experiments uses the Dirichlet prior (with parameters $(\frac{1}{2},\ldots,\frac{1}{2})$) [2]. Within a constant factor, this algorithm attains the optimal ratio (1) with respect to CBAL*.[4] The algorithm is equivalent to a particular static distribution over the

[2] Helmbold *et al.* show how to eliminate the need to know $x_{\min}$ and n. While EG can be made universal, its performance ratio is only sub-exponential (and not polynomial) in n.

[3] Following Helmbold *et al.* we fix $\eta = 0.01$ in our experiments.

[4] Experimentally (on our datasets) there is a negligible difference between the uniform universal algorithm in [3] and the above Dirichlet universal algorithm.

class of all CBALs. This equivalence helps to demystify the universality result and also shows that the algorithm can never outperform CBAL*.

A different type of "winner learning" algorithm can be obtained from any sequence prediction strategy. For each stock, a (soft) sequence prediction algorithm provides a probability $p(j)$ that the next symbol will be $j \in \{1, \ldots, m\}$. We view this as a prediction that stock j will have the best price relative for the next day and set $\mathbf{b}_{t+1}(j) = p_j$. We consider predictions made using the prediction component of the well-known Lempel-Ziv (LZ) lossless compression algorithm [9]. This prediction component is nicely described in Langdon [10] and in Feder [11]. As a prediction algorithm, LZ is provably powerful in various senses. First it can be shown that it is asymptotically optimal with respect to any stationary and ergodic finite order Markov source (Rissanen [12]). Moreover, Feder shows that LZ is also universal in a worst case sense with respect to the (offline) benchmark class of all finite state prediction machines. To summarize, the common approach to devising PS algorithms has been to attempt and learn winners using winner learning schemes.

3 The Anticor Algorithm

We propose a different approach, motivated by the CBAL "philosophy". How can we interpret the success of the uniform CBAL on the Cover and Gluss example of Sec. 1? Clearly, the uniform CBAL here is taking advantage of price fluctuation by constantly transferring wealth from the high performing stock to the anti-correlated low performing stock. Even in a less contrived market, we should be able to take advantage when a stock is currently outperforming other stocks especially if this strong performance is anti-correlated with the performance of these other stocks. Our ANTICOR$_w$ algorithm considers a short market history (consisting of two consecutive "windows", each of w trading days) so as to model statistical relations between each pair of stocks. Let

$$\mathsf{LX}_1 = \log(\mathbf{x}_{t-2w+1}), \ldots, \log(\mathbf{x}_{t-w})^T \text{ and } \mathsf{LX}_2 = \log(\mathbf{x}_{t-w+1}), \ldots, \log(\mathbf{x}_t)^T,$$

where $\log(\mathbf{x}_k)$ denotes $(\log(x_k(1)), \ldots, \log(x_k(m)))$. Thus, LX_1 and LX_2 are the two vector sequences (equivalently, two $w \times m$ matrices) constructed by taking the logarithm over the market subsequences corresponding to the time windows $[t-2w+1, t-w]$ and $[t-w+1, t]$, respectively. We denote the jth column of LX_k by $\mathsf{LX}_k(j)$. Let $\mu_k = (\mu_k(1), \ldots, \mu_k(m))$, be the vectors of averages of columns of LX_k (that is, $\mu_k(j) = E\{LX_k(j)\}$). Similarly, let σ_k, be the vector of standard deviations of columns of LX_k. The cross-correlation matrix (and its normalization) between column vectors in LX_1 and LX_2 are defined as:

$$M_{cov}(i,j) = (\mathsf{LX}_1(i) - \mu_1(i))^T (\mathsf{LX}_2(j) - \mu_2(j));$$

$$M_{cor}(i,j) \begin{cases} \frac{M_{cov}(i,j)}{\sigma_1(i)\sigma_2(j)} & \sigma_1(i), \sigma_2(j) \neq 0; \\ 0 & \text{otherwise.} \end{cases}$$

$M_{cor}(i,j) \in [-1,1]$ measures the correlation between log-relative prices of stock i over the first window and stock j over the second window. For each pair of stocks i and j we compute $\mathsf{claim}_{i \to j}$, the extent to which we want to shift our investment from stock i to stock j. Namely, there is such a claim iff $\mu_2(i) > \mu_2(j)$ and $M_{cor}(i,j) > 0$ in which case $\mathsf{claim}_{i \to j} = M_{cor}(i,j) + A(i) + A(j)$ where $A(h) = |M_{cor}(h,h)|$ if $M_{cor}(h,h) < 0$, else 0. Following our interpretation for the success of a CBAL, $M_{cor}(i,j) > 0$ is used to predict that stocks i and j will be correlated in consecutive windows (i.e. the current window and the next window based on the evidence for the last two windows) and $M_{cor}(h,h) < 0$ predicts that stock h will be anti-correlated with itself over consecutive windows. Finally, $\mathbf{b}_{t+1}(i) = \tilde{\mathbf{b}}_t(i) + \sum_{j \neq i} [\mathsf{transfer}_{j \to i} - \mathsf{transfer}_{i \to j}]$ where $\mathsf{transfer}_{i \to j} = \tilde{\mathbf{b}}_t(i) \cdot \mathsf{claim}_{i \to j} / \sum_j \mathsf{claim}_{i \to j}$ and $\tilde{\mathbf{b}}_t$ is the resulting portfolio just after market closing (on day t).

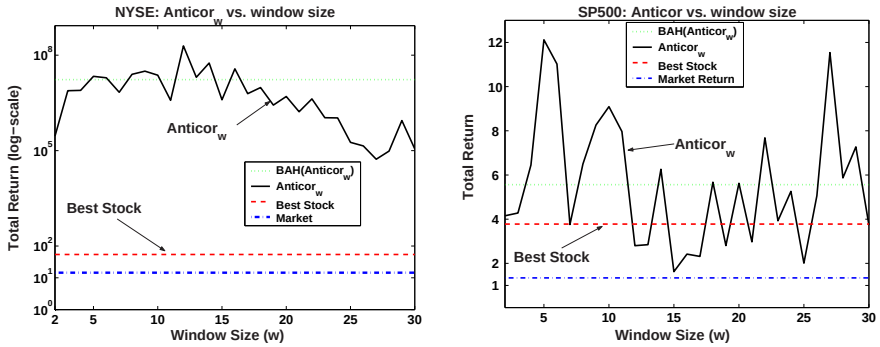

Figure 1: ANTICOR$_w$'s total return (per \$1 investment) vs. window size $2 \leq w \leq 30$ for NYSE (left) and SP500 (right).

Our ANTICOR$_w$ algorithm has one critical parameter, the window size w. In Figure 1 we depict the total return of ANTICOR$_w$ on two historical datasets as a function of the window size $w = 2, \ldots, 30$. As we might expect, the performance of ANTICOR$_w$ depends significantly on the window size. However, for all w, ANTICOR$_w$ beats the uniform market and, moreover, it beats the best stock using most window sizes. Of course, in online trading we cannot choose w in hindsight. Viewing the ANTICOR$_w$ algorithms as experts, we can try to learn the best expert. But the windows, like individual stocks, induce a rather volatile set of experts and standard expert combination algorithms [13] tend to fail.

Alternatively, we can adaptively learn and invest in some weighted average of all ANTICOR$_w$ algorithms with w less than some maximum W. The simplest case is a uniform investment on all the windows; that is, a uniform buy-and-hold investment on the algorithms ANTICOR$_w$, $w \in [2, W]$, denoted by BAH$_W$(ANTICOR). Figure 2 (left) graphs the total return of BAH$_W$(ANTICOR) as a function of W for all values of $2 \leq W \leq 50$ with respect to the NYSE dataset (see details below). Similar graphs for the other datasets we consider appear qualitatively the same and the choice $W = 30$ is clearly not optimal. However, for all $W \geq 3$, BAH$_W$(ANTICOR) beats the best stock in all our experiments.

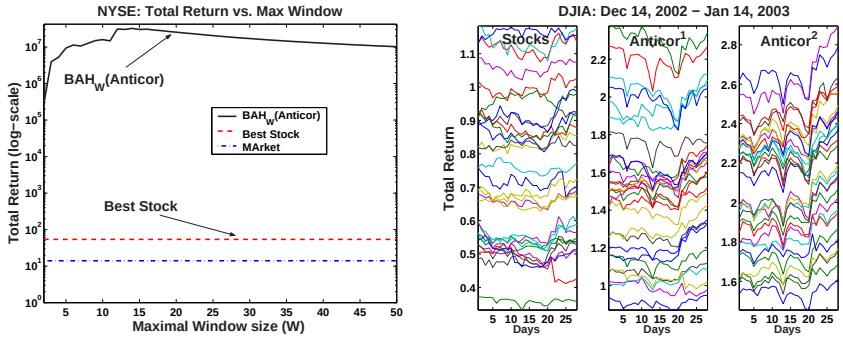

Figure 2: **Left:** BAH$_W$(ANTICOR)'s total return (per \$1 investment) as a function of the maximal window W. **Right:** Cumulative returns for last month of the DJIA dataset: stocks (left panel); ANTICOR$_w$ algorithms trading the stocks (denoted ANTICOR^1, middle panel); ANTICOR$_w$ algorithms trading the ANTICOR algorithms (right panel).

Since we now consider the various algorithms as stocks (whose prices are determined by

the cumulative returns of the algorithms), we are back to our original portfolio selection problem and if the ANTICOR algorithm performs well on stocks it may also perform well on algorithms. We thus consider active investment in the various ANTICOR_w algorithms using ANTICOR. We again consider all windows $w \leq W$. Of course, we can continue to compound the algorithm any number of times. Here we compound twice and then use a buy-and-hold investment. The resulting algorithm is denoted $\text{BAH}_W(\text{ANTICOR}(\text{ANTICOR}))$. One impact of this compounding, depicted in Figure 2 (right), is to smooth out the anti-correlations exhibited in the stocks. It is evident that after compounding twice the returns become almost completely correlated thus diminishing the possibility that additional compounding will substantially help.[5] This idea for eliminating critical parameters may be applicable in other learning applications. The challenge is to understand the conditions and applications in which the process of compounding algorithms will have this smoothing effect!

4 Experimental Results

We present an experimental study of the the ANTICOR algorithm and the three online learning algorithms described in Sec. 2. We focus on $\text{BAH}_{30}(\text{ANTICOR})$, abbreviated by ANTI^1 and $\text{BAH}_{30}(\text{ANTICOR}(\text{ANTICOR}))$, abbreviated by ANTI^2. Four historical datasets are used. The first NYSE dataset, is the one used in [3, 2, 8, 14]. This dataset contains 5651 daily prices for 36 stocks in the New York Stock Exchange (NYSE) for the twenty two year period July 3^{rd}, 1962 to Dec 31^{st}, 1984. The second TSE dataset consists of 88 stocks from the Toronto Stock Exchange (TSE), for the five year period Jan 4^{th}, 1994 to Dec 31^{st}, 1998. The third dataset consists of the 25 stocks from SP500 which (as of Apr. 2003) had the largest market capitalization. This set spans 1276 trading days for the period Jan 2^{nd}, 1998 to Jan 31^{st}, 2003. The fourth dataset consists of the thirty stocks composing the Dow Jones Industrial Average (DJIA) for the two year period (507 days) from Jan 14^{th}, 2001 to Jan 14^{th}, 2003.[6]

These four datasets are quite different in nature (the market returns for these datasets appear in the first row of Table 1). While every stock in the NYSE increased in value, 32 of the 88 stocks in the TSE lost money, 7 of the 25 stocks in the SP500 lost money and 25 of the 30 stocks in the "negative market" DJIA lost money. All these sets include only highly liquid stocks with huge market capitalizations. In order to maximize the utility of these datasets and yet present rather different markets, we also ran each market in reverse. This is simply done by reversing the order and inverting the relative prices. The reverse datasets are denoted by a '-1' superscript. Some of the reverse markets are particularly challenging. For example, *all* of the NYSE^{-1} stocks are going down. Note that the forward and reverse markets (i.e. U-BAH) for the TSE are both increasing but that the TSE^{-1} is also a challenging market since so many stocks (56 of 88) are declining.

Table 1 reports on the total returns of the various algorithms for all eight datasets. We see that prediction algorithms such as LZ can do quite well but the more aggressive ANTI^1 and ANTI^2 have excellent and sometimes fantastic returns. Note that these active strategies beat the best stock and even CBAL^* in all markets with the exception of the TSE^{-1} in which they still significantly outperform the market. The reader may well be distrustful of what appears to be such unbelievable returns for ANTI^1 and ANTI^2 especially when applied to the NYSE dataset. However, recall that the NYSE dataset consists of $n = 5651$ trading days and the y such that y^n = the total NYSE return is approximately 1.0029511 for ANTI^1 (respectively, 1.0074539 for ANTI^2); that is, the average daily increase is less than .3%

[5] This smoothing effect also allows for the use of simple prediction algorithms such as "expert advice" algorithms [13], which can now better predict a good window size. We have not explored this direction.

[6] The four datasets, including their sources and individual stock compositions can be downloaded from http://www.cs.technion.ac.il/∼rani/portfolios.

(respectively, .75%). Thus a transaction cost of 1% can present a significant challenge to such active trading strategies (see also Sec. 5). We observe that UNIVERSAL and EG have no substantial advantage over U-CBAL. Some previous expositions of these algorithms highlighted particular combinations of stocks where the returns significantly outperformed UNIVERSAL and the best stock. But the same can be said for U-CBAL.

Algorithm	NYSE	TSE	SP500	DJIA	NYSE^{-1}	TSE^{-1}	SP500^{-1}	DJIA^{-1}
MARKET (U-BAH)	14.49	1.61	1.34	0.76	0.11	1.67	0.87	1.43
BEST STOCK	54.14	6.27	3.77	1.18	0.32	**37.64**	1.65	2.77
CBAL*	250.59	6.77	4.06	1.23	2.86	**58.61**	1.91	2.97
U-CBAL	27.07	1.59	1.64	0.81	0.22	1.18	1.09	1.53
ANTI^1	**17,059,811.56**	**26.77**	**5.56**	**1.59**	**246.22**	7.12	**6.61**	**3.67**
ANTI^2	**238,820,058.10**	**39.07**	**5.88**	**2.28**	**1383.78**	7.27	**9.69**	**4.60**
LZ	79.78	1.32	1.67	0.89	5.41	4.80	1.20	1.83
EG	27.08	1.59	1.64	0.81	0.22	1.19	1.09	1.53
UNIVERSAL	26.99	1.59	1.62	0.80	0.22	1.19	1.07	1.53

Table 1: Monetary returns in dollars (per $1 investment) of various algorithms for four different datasets and their reversed versions. The winner and runner-up for each market appear in boldface. All figures are truncated to two decimals.

5 Concluding Remarks

When handling a portfolio of m stocks our algorithm may perform up to m transactions per day. A major concern is therefore the commissions it will incur. Within the *proportional commission* model (see e.g. [14] and [15], Sec. 14.5.4) there exists a fraction $\gamma \in (0,1)$ such that an investor pays at a rate of $\gamma/2$ for each buy and for each sell. Therefore, the return of a sequence $\mathbf{b}_1, \ldots, \mathbf{b}_n$ of portfolios with respect to a market sequence $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is $\prod_t \left(\mathbf{b}_t \cdot \mathbf{x}_t (1 - \sum_j \frac{\gamma}{2} |\mathbf{b}_t(j) - \tilde{\mathbf{b}}_t(j)|) \right)$, where $\tilde{\mathbf{b}}_t = \frac{1}{\mathbf{b}_t \cdot \mathbf{x}_t} (\mathbf{b}_t(1)\mathbf{x}_t(1), \ldots, \mathbf{b}_t(m)\mathbf{x}_t(m))$. Our investment algorithm in its simplest form can tolerate very small proportional commission rates and still beat the best stock.[7] We note that Blum and Kalai [14] showed that the performance guarantee of UNIVERSAL still holds (and gracefully degrades) in the case of proportional commissions. Many current online brokers only charge a small per share commission rate. A related problem that one must face when actually trading is the difference between bid and ask prices. These bid-ask *spreads* (and the availability of stocks for both buying and selling) are typically functions of stock liquidity and are typically smaller for large market capitalization stocks. We consider here only very large market cap stocks. As a final caveat, we note that we assume that any one portfolio selection algorithm has no impact on the market! But just like any goose laying golden eggs, widespread use will soon lead to the end of the goose; that is, the market will quickly react.

Any report of abnormal returns using historical markets should be suspected of "data snooping". In particular, when a dataset is excessively mined by testing many strategies there is a substantial chance that one of the strategies will be successful by simple overfitting. Another data snooping hazard is stock selection. For example, the 36 stocks selected for the NYSE dataset were all known to have survived for 22 years. Our ANTICOR algorithms were fully developed using only the NYSE and TSE datasets. The DJIA and SP500 sets were obtained (from public domain sources) after the algorithms were fixed. Finally, our algorithm has one parameter (the maximal window size W). Our experiments indicate that the algorithm's performance is robust with respect to W (see Figure 2).

[7] For example, with $\gamma = 0.1\%$ we can typically beat the best stock. These results will be presented in the full paper.

A number of well-respected works report on statistically robust "abnormal" returns for simple "technical analysis" heuristics, which slightly beat the market. For example, the landmark study of Brock *et al.* [16] apply 26 simple trading heuristics to the DJIA index from 1897 to 1986 and provide strong support for technical analysis heuristics. While *consistently* beating the market is considered a great (if not impossible) challenge, our approach to portfolio selection indicates that beating the best stock is an achievable goal. What is missing at this point of time is an analytical model which better explains why our active trading strategies are so successful. In this regard, we are investigating various "statistical adversary" models along the lines suggested by [17, 18]. Namely, we would like to show that an algorithm performs well (relative to some benchmark) for any market sequence that satisfies certain constraints on its empirical statistics.

References

[1] G. Lugosi. Lectures on prediction of individual sequences. URL:http://www.econ.upf.es/∼lugosi/ihp.ps, 2001.

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

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

[4] H. Markowitz. *Portfolio Selection: Efficient Diversification of Investments*. John Wiley and Sons, 1959.

[5] T.M. Cover and D.H. Gluss. Empirical bayes stock market portfolios. *Advances in Applied Mathematics*, 7:170–181, 1986.

[6] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[7] A. Lo and C. MacKinlay. *A Non-Random Walk Down Wall Street*. Princeton University Press, 1999.

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

[9] J. Ziv and A. Lempel. Compression of individual sequences via variable rate coding. *IEEE Transactions on Information Theory*, 24:530–536, 1978.

[10] G.G. Langdon. A note on the lempel-ziv model for compressing individual sequences. *IEEE Transactions on Information Theory*, 29:284–287, 1983.

[11] M. Feder. Gambling using a finite state machine. *IEEE Transactions on Information Theory*, 37:1459–1465, 1991.

[12] J. Rissanen. A universal data compression system. *IEEE Transactions on Information Theory*, 29:656–664, 1983.

[13] N. Cesa-Bianchi, Y. Freund, D. Haussler, D.P. Helmbold, R.E. Schapire, and M.K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, May 1997.

[14] A. Blum and A. Kalai. Universal portfolios with and without transaction costs. *Machine Learning*, 30(1):23–30, 1998.

[15] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[16] L. Brock, J. Lakonishok, and B. LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance*, 47:1731–1764, 1992.

[17] P. Raghavan. A statistical adversary for on-line algorithms. DIMACS *Series in Discrete Mathematics and Theoretical Computer Science*, 7:79–83, 1992.

[18] A. Chou, J.R. Cooperstock, R. El-Yaniv, M. Klugerman, and T. Leighton. The statistical adversary allows optimal money-making trading strategies. In *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1995.