Online Equity Portfolio Allocation

Kevin Wu (kjw2157)
Gregory Johnsen (gwj2108)
John Heintschel (jch2196)
Sasha Beltinova (sab2229)

December 2017

1 Introduction

1.1 Problem Space and Suitability of Online Learning

The stock market presents a natural candidate for applications of online learning, given its sequential nature and immediate and fully observable rewards. In this context, the algorithm is a wealth-maximizing agent and the set of action spaces is a set of available stocks to trade.

In class, we discussed *Hedge*, an online algorithm with experts that minimizes hindsight regret given fully observable rewards across all actions. While *Hedge* maintains a probability distribution over actions that the agent samples from at every round, an investor in the stock market does not need to be constrained to selecting just one action per time period. Instead, the vector of weights may be conceived of as an allocation across a portfolio of multiple stocks, and the investor may simultaneously take many actions in order to realize the weightings implied by the vector.

The foundation for most present-day approaches to portfolio allocation, both in academia and in industry, can be traced back to Harry Markowitz's 1952 paper on portfolio selection [6]. Markowitz's framework, commonly known as Modern Portfolio Theory, seeks to optimize portfolio returns for a given level of risk (mean-variance optimization). While these objectives make sense in many investment contexts, mean-variance optimization is heavily reliant on accurate estimation of its parameters from historical data.

Online learning methods provide an alternative, non-parametric approach to the portfolio allocation problem. Thomas Cover's 1991 "Universal Portfolio" [2] is one of the earliest such model-free methods, which provably achieves a total return close to that of the best best possible result in hindsight. However, Cover's *Universal Portfolio* algorithm is exponential in the number of assets and relies on sampling over the set of available assets with large N. Several years later, Helmbold et al. [4] proposed the *Exponentiated Gradient (EG)*, an alternative method whose complexity is linear in the number of assets, but with similar empirical results as Cover's algorithm. More recently, others such as Das and Banerjee (2011) have expanded on Helmbold's EG, using it as a building block in a meta algorithm (MA) and achieving even stronger empirical and theoretical performance.[3]

1.2 Our Present Study

Our project offers a novel adaptation of Helmbold's canonical EG algorithm, which we term EG with "sophisticated experts." In this setting, EG serves as a meta algorithm itself. Rather than arms corresponding directly with positions in a given stock, each arm corresponds to the outcome of a common technical trading strategy on a particular stock. In the course of this study, we also attempt to replicate Helmbold's original empirical results, but using modern-day financial data dating back to 2000. We find that for more than 2 or 3 stocks, the EG algorithm fails to yield as high overall returns relative to various portfolio allocation benchmarks as was shown by Helmbold. However, we also find that EG with sophisticated experts is able to recover similar levels of performance, in line with benchmark guarantees. We leave the task of analyzing the theoretical underpinnings of these findings for a future report.

2 Preliminaries

Before going into the results of our experiment, we first provide some mathematical details on the portfolio allocation problem in the context of online learning, as well as Helmbold's EG algorithm, which is the basis of our research.

2.1 Portfolio Allocation Setup

The portfolio allocation problem can be formalized as follows.

- Given a set of K stocks, an agent invests wealth according to the non-negative weight vector $\boldsymbol{w} = (w^1, \dots w^K)$ where $\sum_{i=1}^{K} w^i = 1$.
- An agent's total wealth at time t given a set of relative price changes[1] $\boldsymbol{x}_t = (x_t^1, \dots x_t^K)$ is represented by the dot product $\boldsymbol{w} \cdot \boldsymbol{x_t}$.
- Over a fixed time period T, given a sequence of portfolio allocations $\boldsymbol{w}_1, \boldsymbol{w}_2, \dots \boldsymbol{w}_T$ and a sequence of relative price changes $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots \boldsymbol{x}_T$, the final wealth, or the *cumulative reward*, is calculated as:

$$S(\cdot) = \prod_{t=1}^{T} (\boldsymbol{w}_t \cdot \boldsymbol{x}_t)$$

2.2 Constantly Rebalanced Portfolio, Best CRP

An important benchmark in the online portfolio allocation literature is the constantly rebalanced portfolio (CRP)—an allocation method in which $\boldsymbol{w}_1 = \boldsymbol{w}_2 = \dots \boldsymbol{w}_T$.[2] The *best* Constantly Rebalanced Portfolio (BCRP), $\boldsymbol{w^*}$, is defined *in hindsight* by:

$$\boldsymbol{w^*} = \underset{w}{\operatorname{argmax}}\, S(\cdot) = \underset{w}{\operatorname{argmax}} \prod_{t=1}^{T} (\boldsymbol{w} \cdot \boldsymbol{x}_t)$$

[1] Current day's price divided by the previous day's price.

[2] Note that this algorithm is not the same as a passive "buy-and-hold" strategy, as maintaining the same $\boldsymbol{w}$ at each round requires selling the stocks which have outperformed in the previous period and buying those that have underperformed.

and serves as an important benchmark of performance throughout the online portfolio allocation literature. In fact, Cover and Thomas are able to show in 1991 that the BCRP is an optimal strategy in an i.i.d. market (an assumption which itself is the subject of a good deal of literature). [1]

2.3 Exponentiated Gradient

At each round, the Exponentiated Gradient (EG) method picks a weight vector that seeks to maximize the overall reward in the next round while penalizing large updates to $\boldsymbol{w}$ [4], as measured by the K-L divergence between $\boldsymbol{w}_{t+1}$ and $\boldsymbol{w}_t$.

$$\underset{\boldsymbol{w}_{t+1}}{\operatorname{argmax}}\, \eta \log(\boldsymbol{w}_{t+1} \cdot \boldsymbol{x_t}) - D_{KL}(\boldsymbol{w}_{t+1} || \boldsymbol{w}_t)$$

An approximate solution to the above optimization results in the following weight update formula:

$$w^i_{t+1} = g(\boldsymbol{w}_t, \boldsymbol{x}_t) = \frac{w^i_t \exp(\eta x^i_t / \boldsymbol{w}_t \cdot \boldsymbol{x}_t)}{\sum_{j=1}^{N} w^j_t \exp(\eta x^j_t / \boldsymbol{w}_t \cdot \boldsymbol{x}_t)} \tag{1}$$

2.3.1 Connections to Hedge

The weight update formula in EG provides a multiplicative update procedure that is linear in the number of arms, similar to Hedge. While Hedge maintains a probability distribution over actions, EG maintains a weight vector whose components sum to 1. The difference in weight update procedures between the two algorithms is due to the way actions are taken simultaneously in the portfolio allocation problem, rather than sampled from a distribution. Finally, EG seeks to maximize multiplicative rather than additive rewards.

3 Body

3.1 Experimental Setup

Provided with some fixed amount of initial wealth, simulations were run of algorithms which were able to buy and sell stocks on each trading day from January 1, 2000 to August 31, 2017.[3] Each algorithm was only able to trade with current stocks from the Dow Jones Industrial Average. [4] In all of our experiments, we considered a maximum of 28 different possible stocks in which to invest [5].

To facilitate the testing of different portfolio allocation algorithms in a consistent and user-friendly way, we set up a custom simulation environment in Python to run our trials.[6] This

[3] Historical end-of-day stock price data was obtained from https://www.quandl.com/.

[4] Current DJIA components: AAPL, AXP, BA, CAT, CSCO, CVX, DD, DIS, GE, GS, HD, IBM, INTC, JNJ, JPM, KO, MCD, MMM, MRK, MSFT, NKE, PFE, PG, UNH, UTX, V, VZ, WMT, XOM.

[5] Although DJIA is comprised of 30 stocks, the historical data for two stocks, V (Visa) and TRV (Travelers Companies Inc), had significant amount of missing data and these stocks were thus excluded from the experiment.

[6] https://github.com/kevjwu/coms6998-project

environment is initialized with a given algorithm, or Agent, and a set of Experts corresponding to the different possible stocks. Different Agents may use different strategies to determine which Expert's recommendations should be followed, and different Experts may use different criterion for making a recommendation. At each round of the simulation, each Expert updates its state with the observed rewards of its corresponding stock; these rewards accrue to the Agent's portfolio; the Agent calculates an update to its weight vector based on the observed rewards of its Experts; and the position is reallocated before moving on to the next round.

3.2 Algorithms Used

3.2.1 Agents

We begin with the original EG algorithm (Algorithm 1), which for 2 or 3 stocks, is shown to have low hindsight regret compared to the BCRP[4]. Wealth is allocated between stocks in proportionality to the weights. In our simulation environment, we call such an expert—one that always recommends to buy a given stock—a "Dummy" expert.

Algorithm 1: Exponentiated Gradient

```
Parameter: η
Given K stocks and initial wealth S_0, initialize weight vector w = (1/K, ...1/K).
for t ∈ {1, ..., T} do
  for i ∈ {1, ..., K} do
    Observe reward x_i^t for stock
    Update weight w_i^{t+1} = g(η, w_i^t, x_i^t). (See Equation 1)
  end for
  Update wealth. S_t = w^t · x^t.
  Reallocate wealth according to w_{t+1}.
end for
```

3.2.2 Experts

Our extension of Helmbold's EG algorithm is to allocate to Experts which have discretion over recommending a given stock (Algorithm 2), rather than allocating directly to the stocks. Each expert is actually a subroutine which maintains some state based on its available historical price data.

In this experiment, we tried two different moving-average based "sophisticated experts", which we call Mean Reversion and Momentum (Algorithm 3). These experts are based on generic technical trading strategies, and both seek to capitalize on price trends in the market, but using opposite approaches.

1. The Momentum expert seeks to find stocks that are performing well, in the hopes that they will continue to perform well. The expert buys when the recent price surpasses some threshold multiple above a moving average.

2. The Mean Reversion expert seeks to find stocks which aren't performing well, in the hopes that in the long run they will make up their losses and revert toward a mean price level. The expert buys when the recent price falls below some threshold multiple of its moving average.

Since these expert subroutines may either choose to buy or not to buy at any given time period, any "unallocated" money is reinvested among experts that are buying in the current time period, proportional to their weights.

Algorithm 2: Exponentiated Gradient with Sophisticated Experts

Parameters: η
Given K experts and initial wealth S_0, initialize weight vector $\boldsymbol{w} = (1/K, ...1/K)$.
for $t \in \{1, \ldots, T\}$ **do**
  for $i \in \{1, \ldots, K\}$ **do**
    Observe reward x_i^t for expert.
    Update weight for expert $w_i^{t+1} = g(\eta, w_i^t, x_i^t)$. (See Equation 1)
    Expert calls subroutine $EXP(x_i)$ to determine whether to BUY or NOT BUY.
  end for
  Update wealth. $S_t = \boldsymbol{w^t} \cdot \boldsymbol{x^t}$.
  Reallocate wealth according to $\boldsymbol{w_{t+1}}$.
end for

Algorithm 3: Moving Average Expert Subroutine: Mean Reversion

Parameters: n (window size), τ (moving average threshold)
Initialize expert for a given stock i.
Initialize a time-ordered list of at most n prices for i, $\boldsymbol{x'} = \emptyset$.
for $t \in \{1, \ldots, T\}$ **do**
  if $t < n$ **then**
    Add x_t^i to list of most recent prices $\boldsymbol{x'}$.
    return NO BUY
  end if
  Remove x'_{t-n} from list of most recent prices $\boldsymbol{x'}$.
  Add x_t^i to list of most recent prices $\boldsymbol{x'}$.
  Estimate the mean μ, and variance σ of last n returns.
  if $x_t^i < \mu - \tau * \sigma$ **then**
    return BUY
  else
    return NO BUY
  end if
end for

3.2.3 Benchmarks

Finally, we ran three common benchmark agents in our simulation framework in order to evaluate the performance of EG and its variations against performance guarantees in the online portfolio allocation literature.[5]

1. Buy and Hold: An agent assigns weights equally to each stock. This is a naive benchmark

in the sense that comparing our algorithms with "buy and hold" will tell us whether a given strategy is better than random portfolio allocation with no learning.

2. Best Fixed Stock: This is a special case of buy and hold, in which all the weight is assigned to the single best stock *in hindsight*.

3. Best Constantly Rebalanced Portfolio (BCRP): This agent constantly updates to maintain weights which have been determined to maximize cumulative returns *in hindsight*. Maximum Likelihood Estimation was used to derive the BCRP weights [2]. [7]

3.3 Results

Overall, we were able to achieve strong empirical performance with our strategy of Exponentiated Gradient using a Mean Reversion expert, achieving returns very close to the BCRP benchmark suggested by Cover and Thomas to be optimal in i.i.d. markets. [1] As mentioned before, we were surprised to find that in our simulations, Helmbold's original EG algorithm performed similarly to the naive Buy and Hold benchmark. We conducted several subexperiments to further analyze these findings.

3.3.1 Simulations Run

For our implementation of Helmbold's original EG algorithm, we report results with a learning rate η of 0.05, which is the same setting used in Helmbold [4]. However, after doing a grid search over possible parameter settings, we found that returns in the presence of outliers was monotonically increasing in the learning rate. We felt that this result was indicative of a potential shortcoming in the EG algorithm in the sense that with high enough η, the algorithm was simply trying to catch up to one or two "winners" in the portfolio, rather than making incremental updates to its allocation. Thus, we ran several subexperiments to better understand this behavior.

In total, we consider results from four different sets of trials:

1. Trading on all 28 stocks

2. Trading on a set of stocks that exclude the two best performing stocks over the time period in question (AAPL and UNH, which we call outliers)[8]

3. Trading on the ten most volatile stocks, which were calculated in hindsight and included the outliers AAPL and UNH

4. Trading on the top ten volatile stocks with AAPL and UNH excluded.

3.3.2 Findings

The EG Agent with the Mean Reversion Expert (EGMRE) achieved an annualized return of 23.8% in our full 28-stock simulation, which approaches the returns of the Best Constant Rebalanced Portfolio, which achieved 26.8% annualized returns over the same period. Dropping the hindsight outliers of AAPL and UNH, EG with Mean Reversion was still able to achieve superior performance, realizing an annualized return of 15.37%, virtually on par with the BCRP return of 15.43%.

[7] https://github.com/Marigold/universal-portfolios/blob/master/universal/algos/bcrp.py

[8] Over the given time period, AAPL and UNH returned 4.5x and 3x over the next best stock, MSFT.

Table 1: Trading Strategies on All Stocks

Strategy:	Wealth	w/o Outliers	APY	APY w/o Outliers
Buy and Hold	6.21	3.68	10.91%	7.67%
Best Fixed Stock	45.41	10.98	24.15%	14.55%
Best CRP	66.01	12.62	26.77%	15.43%
EG - Dummy	6.18	4.53	10.89%	8.65%
EG - Momentum	4.31	1.54	8.54%	2.46%
EG - Mean Reversion	42.79	12.38	23.80%	15.37%

Table 2: Trading Strategies on the 10 Most Volatile Stocks

Strategy:	Wealth	w/o Outliers	APY	APY w/o Outliers
Buy and Hold	10.17	2.86	14.06%	6.14 %
Best Fixed Stock	10.98	10.98	14.55%	14.55%
Best CRP	66.01	7.48	26.77%	12.07%
EG - Dummy	9.38	3.69	13.53%	7.68%
EG - Momentum	3.44	0.55	7.27%	-0.03%
EG - Mean Reversion	45.80	21.62	24.25%	19.06%

EG with Dummy experts achieved a final wealth of 6.18, which translates to an annualized return, or annualized percentage yield (APY), of 10.89% on the full data set. Disappointingly, this was only in line with the results achieved by a naive Buy and Hold strategy which achieves returns of 10.91% over the simulation period. Furthermore, EG trailed the best CRP benchmark by a large margin. This finding was persistent across the four simulations we tried; whether removing high performing outliers or looking only at volatile stocks, EG was generally close to the Buy and Hold benchmark. We would like to conduct further research to understand this divergence from Helmbold's original findings. We can hypothesize that it could be due to structural changes in the financial markets, or a potential "pricing in" of the EG strategy by the markets.

Finally, the EG Agent with the Momentum Expert performed the worst of all strategies, with an annualized return of 8.54%, worse than any benchmark.

In simulations run without the extreme high-performing outliers of AAPL and UNH, all strategies unsurprisingly perform worse; however, the relative rankings of strategies was mostly the same, with the exception of EGMRE, which outperformed Best Fixed Stock with outliers removed.

3.4 Qualifications and Practical Considerations

While our simulation was run in an environment similar to those in the literature, it is worth discussing some inherent limitations of the arrangement. The first arises from the notion of slippage, which refers to any discrepancy between the price at which one anticipates executing a trade and the price at which it is fulfilled. Slippage may occur due to latency (time difference between signals) or the price impact of a trade (a large trade of an illiquid asset may actually drive the asset price). Though the impact of slippage can be nontrivial, we found it to be beyond the scope of this project.[9]

[9] It is also worth noting that slippage is less likely to be significant in a mean reversion strategy, where positions are counter to the prevailing price momentum.

Wealth (Full, 28-Stock Action Space)

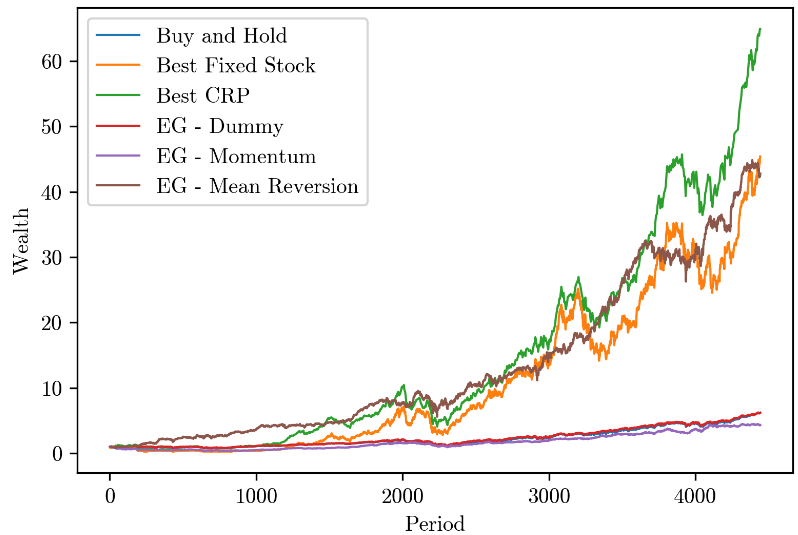

Figure 1: Our EG algorithm with a Mean Reversion expert achieves performance in line with the BCRP and Best Fixed Stock benchmarks. Helmbold's original EG is in line with the Buy and Hold benchmark.

Wealth (Reduced Action Space Without Outlier Performers)

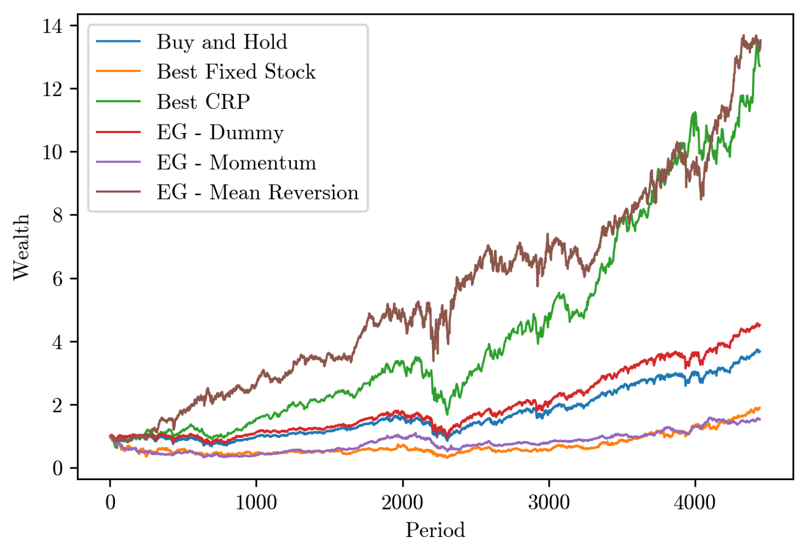

Figure 2: Even with strong performers removed, EG Mean Reversion keeps pace with the BCRP benchmark. Helmbold's original EG performs slightly above Buy and Hold in this setting.

A second limitation is in commission or exchange costs associated with executing a trade. In practice these costs vary greatly, so we do not attempt to include them in our simulation, but suffice it to say that it would be an important consideration before any real-world implementation of a trading strategy, and one which plagues other recent improvements on the EG algorithm. [3]

Interestingly, we did note a common behavior in our EG Mean Reversion simulations, whereby positions in many equities would be closed out and held at 0 for several periods at a time—that is to say, our portfolio allocation was often *sparse*. This is actually held as a desirable feature by Das [3] and others for its ability to minimize transaction costs. We would like to conduct further empirical research to understand the prevalence of this behavior in EG with Sophisticated Experts. If it is a common outcome, it could be an important practical benefit of our algorithm.

4 Conclusions

We are pleased by the strong empirical performance of our EG algorithm using Mean Reversion experts, and also interested in its potential to mitigate transaction costs, as this has been a challenge for more recent EG-like models. At the same time, we would like to do further exploration to understand why Helmbold's original EG algorithm without sophisticated experts seems to have broken down.

Overall, we find online portfolio allocation to be a rich problem space, and the possible extensions to our work are many. To briefly mention a few avenues for further exposition:

- *Changing market regimes:* Singer's "Switching Portfolios" paper seeks to capitalize on changing market regimes by minimizing a shifting regret bound (a best dynamically rebalanced portfolio, rather than a best CRP) [7].

- *Side information:* Helmbold et al. show that running EG with side information yielded even greater returns than regular EG, a result which we did not have the chance to run on our modern-day dataset. [4]

- *Meta Optimization:* Das and Banerjee show quite strong results feeding EG into a higher meta-algorithm, but also raise the need to find sparse strategies to mitigate transaction costs. [3]

- *Online Variance Minimization:* Warmuth and Kuzmin show interesting theoretical results from a special case of EG designed to manage variance. They are able to prove bounds on the variance incurred by EG relative to the best offline alternative. They point out the usefulness of manging variance in the online portfolio allocation setting, but do not undertake an empirical study. [8]

References

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley-Interscience, New York, 1991.

[2] Thomas M. Cover. "Universal Portfolios". In: *Mathematical Finance* 1.1 (1991), pp. 1–29. DOI: `http://www-isl.stanford.edu/~cover/papers/paper93.pdf`.

[3] Puja Das and Arindam Banerjee. "Meta Optimization and its Application to Portfolio Selection". In: *Proceedings of the 17th ACM SIGKDD* (2011), pp. 1163–1171. URL: `https://dl.acm.org/citation.cfm?id=2020588`.

[4] David P. Helmbold et al. "On-Line Portfolio Selection Using Multiplicative Updates". In: *Mathematical Finance* 8.4 (1998), pp. 325–347. DOI: `http://www.cis.upenn.edu/~mkearns/finread/helmbold98line.pdf`.

[5] Bin Li and Steven C.H. Hoi. "Online Portfolio Selection: A Survey". In: *ACM Comput. Surv.* 46.3 (2014).

[6] Harry Markowitz. "Portfolio Selection". In: *The Journal Of Finance* 7.1 (1952), pp. 77–91. DOI: `https://www.math.ust.hk/~maykwok/courses/ma362/07F/markowitz_JF.pdf`.

[7] Yoram Singer. "Switching Portfolios". In: *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (1998), pp. 488–495. DOI: `https://arxiv.org/pdf/1301.7413.pdf`.

[8] Manfred K. Warmuth and Dima Kuzmin. "Online variance minimization". In: *Machine Learning* 87 (2012). DOI: `10.1007/s10994-011-5269-0`.