Final Project Report
ISYE 4803: Online Learning and Decision Making

Jessica Buzzelli
Soujanya Duggirala
Kiran Gite
Diego Granizo
Annie Ni

12/2/2019

I. Abstract

In financial markets, predicting stocks' future prices is a tall order and attempting to produce such a model would be underwhelming in regards to profit, accuracy, or reproducibility. In this study, the team instead attempted to determine which of four full-information algorithms[1] was best at isolating winning and losing stocks and changing the weight of a stock in its portfolio accordingly. In each trial, the final model aimed to maximize relative return (%) or maximize absolute return ($). In this setup, each portfolio was comprised of individual stocks ("experts") that could be bought based on the weights determined by the algorithm used, with the weight of a stock representing the proportion of total portfolio value invested in a stock. A successful “winning” algorithm would learn to assign high-performing stocks a larger weight in its portfolio and decrease weights of low-performing stocks.

In addition to running each of the two test groups through the four different full information algorithms, the team ran different iterations of each replication using varying timesteps and weight initialization schemes.[2] At each timestep, market conditions were replicated by considering a 1.5% reweighting fee to mimic transaction costs from trading the portfolios' underlying assets. All design permutations are shown in Figure 1 below.

The goal of this study is not to propose a new algorithm for online portfolio selection; the team instead was seeking to analyze the relative suitability of online portfolio selection methods for portfolios with different characteristics.[3] The team considered using Thompson sampling to determine the ultimate “winner” among all setting combinations, but decided against that due to the full-information nature of the market. Instead, each trial was assessed based on visualizations of portfolio values over time.

A primary finding of the study was that algorithms run on high volatility portfolios had much higher returns than those run on low volatility portfolios. Initializing at equal weights was effective for long time periods, but initializing weights based on market capitalization was better for shorter time periods. Furthermore, losses based on absolute return were better than price relative return. Regarding algorithms, MWU and Hedge both produced similar results and both performed better than the RWMA, with exponential gradient having the highest returns.

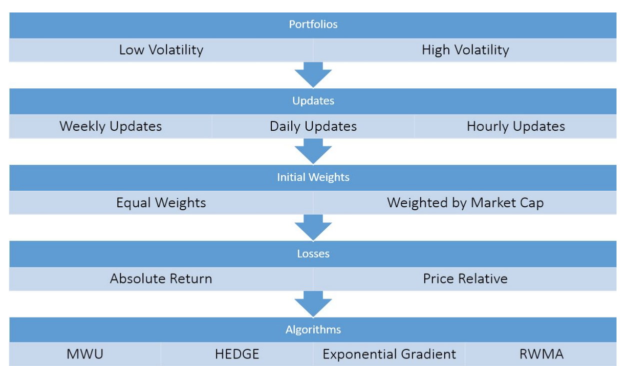


Figure 1
Above: An exhaustive map of the different method used for each component of the final experimental design.

---

[1] Randomized weighted majority, multiplicative weight update, Hedge, and exponential gradient
[2] Timesteps: weekly, daily, and hourly; Weight update schemes: equally distributed or by market cap
[3] Test groups: low volatility (19 stocks), high volatility (18 stocks)

II. Background Research

Problem definition & market considerations

Many studies have been conducted in an attempt to create an ensemble model to predict stocks' future prices, all with varying degrees of failure. As expressed in Burton Makiel's book *A Random Walk Down Wall Street*,[4] stocks' prices are best described as "random walks" where stocks' prices move up or down in random increments at each timestep in a non-formulaic, unpredictable manner due to market conditions, the company and industry's outlook, the state of consumers, and large "market maker" investors' beliefs.

Since stocks' listed prices are calculated as functions of the latest price the stock was bought and sold at in relation to the volume of such trades, the market constantly looks to exploit stocks that are listed at prices investors believe to be below the actual intrinsic value of the asset. This concept of a difference in list price and value ("alpha")[5] fuels how investment firms and individuals alike assess whether stocks are a buy or a sell, a bull or a bear, a winner or loser. While individuals can make money in the market through algorithmic investing, return percentages are typically very low and volatile to a point where it is difficult to say with any certainty that the returns are the result of a winning strategy versus chance.

Since predictions about individual stocks' future prices cannot be expected to have any degree of accreditable accuracy, the team instead opted to isolate high and low performing stocks in a portfolio setting. The team will focus on predicting the likelihood a stock will perform well relative to its cohort based on historical patterns instead of the numerical returns themselves. While stock prices do matter in a portfolio setting, the need to accurately predict the prices to understand if a portfolio will improve or worsen after a composition rebalancing (changing the proportion of funds invested in each stock) is practically eliminated by predictions around the directionality and relative magnitude of each stock's outlook.[6]

Choice of algorithms

The team applied the online learning framework by considering each individual stock to be an "expert," with losses defined as their return over each time period. Four full information algorithms were selected for testing and the team determined appropriate hyperparameters and design attributes for each as previously shown in Figure 1. Based on similar applications of these algorithms[7], the team gathered that online models tend to be most competitive when applied to companies with limited market data since their ability to capture patterns in high volatility scenarios eclipses that of other models such as regression, which rely on the data meeting several core assumptions around normality, collinearity, and/or formulaic nature. This hypothesis is later revisited in Results and Analysis.

[4] https://www.exploring-economics.org/en/discover/an-introduction-to-burton-malkiels-a-random-walk-d/
[5] https://www.investopedia.com/articles/financial-theory/08/deeper-look-at-alpha.asp
[6] https://www.academia.edu/32179312/The_Science_of_Algorithmic_Trading_and_Portfolio_Management
[7] https://queue.acm.org/detail.cfm?id=2534976

Choice of asset classes

In this study, the team decided to limit portfolios to only stocks for simplicity as well as in order to keep the underlying use of the models in a format that did not require extensive financial background.

While extremely volatile and likely a very interesting use of online learning models, derivative assets did not align with the scope of this particular study. For example, exploring results from portfolios of derivative securities such as options or futures[8] would require selecting available contracts that expired on the same or similar dates and assessing their worth over time by referencing stock prices. Unlike with stocks, which the team safely assumed would continue being publically traded for the foreseeable future, applying models to expirable securities would tell users nothing about market trends or what to do with investors' money after the training period. Instead, the model would have to be adapted to evaluate portfolios' success at every timestep during the training period while being weighed down by warm-up bias and similar effects.

Additionally, the team did not use exchange traded funds (ETFs) in this study since they are already portfolios in themselves and have predictable, conservative returns. While they could be included in the low volatility portfolio or as a comparison metric, this would be too similar to including a risk free asset, which did not follow this study's scope of optimizing fully risk-inclusive portfolios.

Lastly, the team considered testing the model with portfolios organized by sector, but concluded that this would provide less significant outputs than investigating portfolios grouped by anticipated risk, since investors rarely focus on a single industry due to diversification concerns.

Choice of performance metrics

Initially, the team had discussed using a multi-armed bandit model to sample the different algorithms' returns on the same initial portfolios in order to determine the overall winner for each experimental environment. Due to the full-information nature of the market where investors do not have to put money into assets to know their return, the team determined the partial-information algorithm to be inappropriate.

In lieu of a "final layer" algorithm to process the results, two widely used metrics were selected to assess the algorithms' level of success: relative return (measures profitability) and portfolio volatility (σ, standard deviation of returns, measures risk). Relative return and portfolio volatility were divided to give a portfolio's coefficient of variation so all portfolios could be compared with a single metric.

To maintain an element of "realness" in our experiment, the team included a way to consider if trading portfolios' assets to achieve the weights from a timestep's loss function were

---

[8] http://bit.ly/34R3Mlo: *Stocks, Bonds, Options, Futures, and Portfolio Insurance: A Rose by Any Other Name*

worthwhile given the cost of conducting such trades. This integration is crucial when modeling the success of high-frequency trading (prices evaluated at every hour or quicker), since investors accrue fees with every trade.

III. Data

Stocks

The team chose to test online learning methods on two different portfolios of 18-19 stocks each, one portfolio representing stocks with low volatility and one representing those with high volatility. Determining the anticipated volatility of stocks during the selection process heavily involved the stocks' historical returns and age. The process was made difficult by the need to supply the algorithms with diverse returns to consider. Details on the groups' compositions can be found in Appendix A.

Source

Preselected stocks' open and close prices were obtained at weekly, daily, and hourly frequencies using the AlphaVantage API.[9] Afterwards, the calculated returns and price relatives of each stock were provided to the algorithms used in each replication of the experiment as the training data.

Timeframe

Weekly returns were obtained for the low volatility portfolio beginning at market open 9:30 AM ET on October 13, 2006, giving 685 rows of data. Daily returns were pulled beginning at market open on October 2, 2006 (3308 rows), and hourly returns were pulled beginning at market open September 26, 2019 (226 rows).

For the high volatility portfolio, weekly returns were obtained beginning at market open 9:30 AM ET on December 3, 1999, giving 1043 rows of data.[10] Daily returns were pulled beginning at market open on November 22, 1999 (5032 rows), and hourly returns were pulled beginning at market open August 18, 2019 (418 rows). All returns were obtained through 2:30 PM ET on November 20, 2019.

IV. Preliminary Analysis

Figure 2 shows each portfolio's composition by sector. The low volatility portfolio had more financial services and consumer defensive (essential consumer goods) stocks, which makes sense since these sectors are not prone to drastic price changes in short periods of time. The high volatility portfolio was mainly comprised of technology stocks, which follows from technology companies being relatively newer, harder to evaluate, and more susceptible to major price changes.

[9] https://www.alphavantage.co/

[10] If a company was not publicly traded in the associated time span, the dataset used a $0 open and close price.

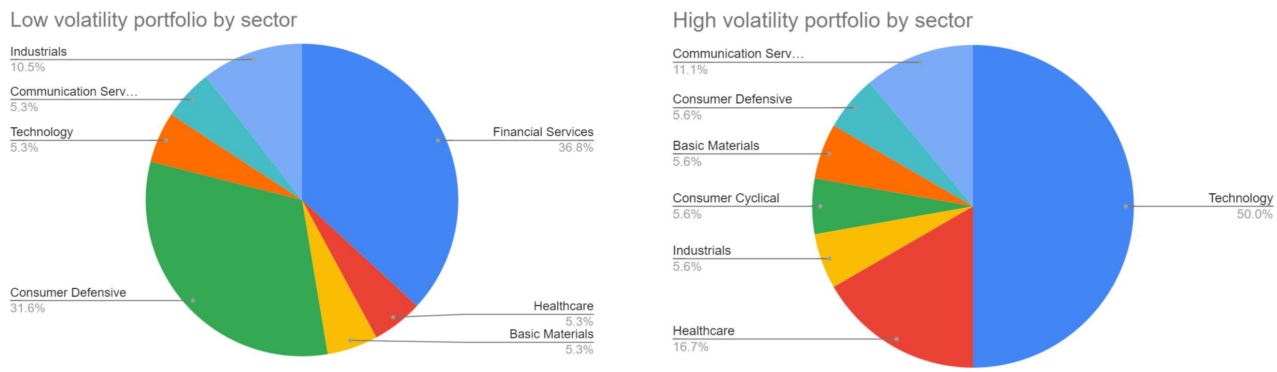


Figure 2

On the left, Consumer Defensive dominated the low volatility portfolio; on the right, a majority of the younger, higher volatility stocks were recent Technology IPOs.

Examining the portfolios by stock volatility and distribution of returns also confirmed the differences between the low and high volatility portfolios. As shown in Figure 3 on the left, the high volatility portfolio had a higher central volatility, while the majority of the low volatility portfolio's stocks had volatilities between 0 and 2.5. On the right, the high volatility portfolio's stock returns were shown to have a much larger spread than the low volatility portfolio's stocks. The team wanted to ensure that the overall returns of the low volatility and high volatility portfolios were similar to avoid bias in results. Average returns of the low and high portfolios were 82.3% and 85.6%, respectively, and a hypothesis test revealed that these returns were not significantly different when considering the volatility of each portfolio.

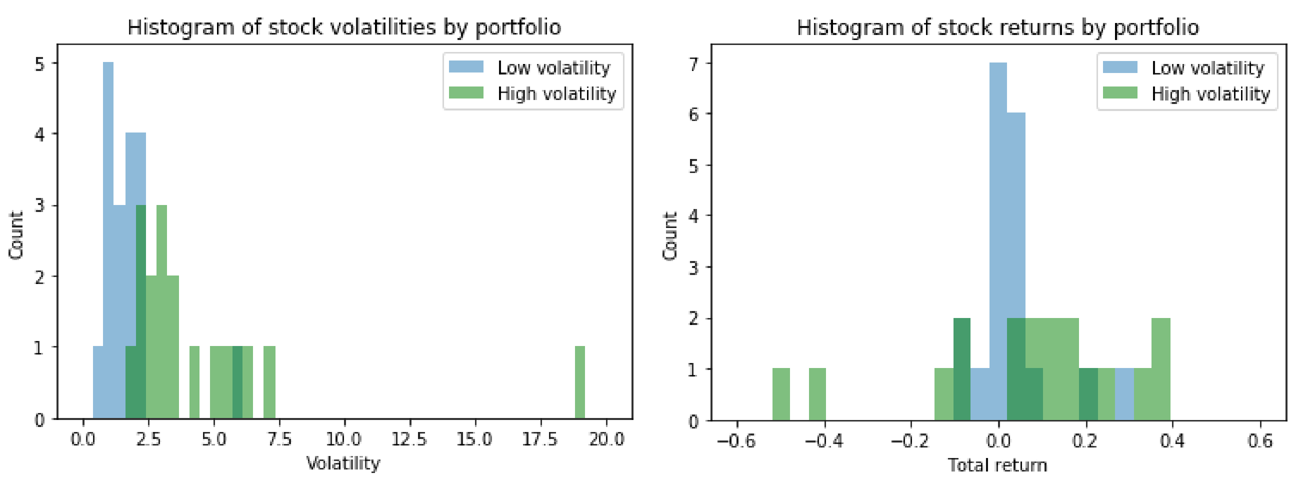


Figure 3

As desired, stock selections for each portfolio represented the correct relative volatilities (left). Lower volatility stocks typically had a smaller return spread (right); this is because low volatility companies tend to be more mature.

Full time series plots of all portfolios over time can be found in Appendix C. It is clear that the high-volatility portfolio stock prices tend to fluctuate more than low-volatility portfolio stock prices.

V. Methods

Objective

In order to determine which online trading strategies are best for finding high-performing stocks within different types of portfolios, the team trained different algorithms and used variations of different inputs to these algorithms in order to determine the optimal combination of parameters for each portfolio. A starting portfolio size of $10 million was chosen because it was large enough to minimize the likelihood of trading fractions of stocks, which is not realistic.

Different times used

One parameter variation was using different time periods to update weights. The team hypothesized that using weekly, daily, or hourly returns would all give different results when comparing the performance of the high- and low-volatility portfolios. For example, the rapidly changing prices in the high-volatility portfolio could possibly be captured more effectively with hourly updates compared to weekly updates.

One issue encountered was that the newer stocks in the high volatility portfolio did not have prices for some of the early weekly and daily time intervals. The team decided to handle this by setting the weights of these stocks to 0 when there was no data for the stocks, and re-initializing the weights to 1 or the market capitalization of the stock after the stock's IPO. This was done for all algorithms except for the exponential gradient algorithm, discussed later on.

Weight initializations

Another parameter that was modified was the way that weights are initialized. The first method used was the standard initialization, setting the weight of each stock equal to 1. The second method was setting the weight of each stock to a value proportional to its market capitalization, defined as the total value of a company's shares of stock[11]. The team hypothesized that given enough data, both weight initializations would eventually converge to the same portfolio weights. However, with the team's limited data, the differences in weight initializations would have more impact. Market capitalization-based initial weights would give larger companies an advantage in the portfolio, while equal weight initialization would not.

Loss types and scaling

Different methods of calculating losses were also tested. The first loss calculation involved the absolute return of a stock in a time period (closing price - opening price in that period). Further scaling operations and calculations were then performed to convert the loss into a format usable for different types of algorithms. The second loss method was based on price relative (opening price in period t ÷ opening price in period t-1), also transformed into the correct format for different algorithms.

These different methods were chosen due to the different stock characteristics they capture; absolute return does not scale price changes by initial magnitude while price relative

[11] https://www.fidelity.com/learning-center/trading-investing/fundamental-analysis/understanding-market-capitalization

does, meaning that if a small stock and a large stock experienced the same price drop, price relative would give a much higher loss to the small stock.

Evaluating transaction costs

An innovation in the algorithm implementations was the inclusion of transaction costs as a factor in decision-making. The team used a fee of 1.5% of the total transaction value since this is a standard fee that is applied currently to high frequency trading. Transaction value was calculated in each time step by adding up the magnitudes of proportion changes of stocks and multiplying by total portfolio value.

If 1.5% of this total “money moved” value was greater than the change in portfolio value from the previous time period to the current period, the algorithm would decide that the reweighting was not “worth it” and no weight updates would be made. If the portfolio made enough money in the previous time period to cover this transaction cost (1.5%*money moved), however, the transaction would be considered “worth it” and stock weights would be updated. Transaction costs incurred were kept track of throughout the algorithm run, and the sum of transaction fees was subtracted from the total portfolio value at the end of the run in order to accurately represent portfolio return.

Algorithms

1. Randomized Weighted Majority Algorithm

The randomized weighted majority algorithm requires a loss between 0 and 1. The team chose not to account for the magnitude of returns in this algorithm, instead setting the loss of a stock to 0 if a return was positive and 1 if return was negative in a time period. Price relative was not used as a basis for loss calculations in this algorithm.

2. Multiplicative Weight Update/Hedge Algorithms

Price relatives and absolute returns were both used as a basis for loss calculations in these algorithms, with good results (price relative > 1 or absolute return > 0) incurring negative losses and bad results incurring positive losses. Since the multiplicative weight update and Hedge algorithms require losses in [-1,1], losses needed to be divided by a big M value. Initially, the largest-magnitude return present across the whole dataset was designated as the big M. However, upon further discussion, the team realized that in an online learning setting, the true maximum of the data would not be known. Instead, the largest magnitude return in a time period was used as the big M for that round.

3. Exponential Gradient - EG(η)

The team found this algorithm in current literature on online portfolio selection. A brief background on the algorithm and its implementation: Following Helmbold, Schapire, Singer, & Warmuth[12] (1998), the exponential gradient algorithm takes in a price relative vector x^t . This vector contains the value of the next day's opening price over the current day’s opening price. The algorithm works similarly to gradient descent by updating the weights of each stock

[12] https://www.cis.upenn.edu/~mkearns/finread/portfolio.pdf

according to a learning rate η . Instead of using root mean squared losses, it utilizes relative entropy. Additionally, this algorithm circumvents the need to bound the losses providing a way to parametrize the weights while only adding a constant term to the bound on portfolio wealth. The weight update simplifies to the following equation which ensures that the weight across each stock is normalized so as to keep the total weight of the portfolio to 1.

$$w_i^{t+1} = \frac{w_i^t \exp(\eta x_i^t / \mathbf{w^t} \cdot \mathbf{x^t})}{\sum_j w_j^t \exp(\eta x_i^t / \mathbf{w^t} \cdot \mathbf{x^t})}$$

VI. Results and Analysis

Return and volatility results of the two portfolios under the RWMA, MWU, and Hedge algorithms with different settings can be found in Appendices A-E. The return of each portfolio type (volatility and time period) was compared to the market return in that time period so the actual advantage of using an algorithm over investing in a market index could be determined.

Best weight initializations and loss methods

One interesting finding is that the equal weight initialization setting outperforms the market cap weight initialization setting in the MWU, RWMA, and Hedge algorithms. For the hourly dataset, though, market cap weight initialization vastly outperforms equal weights in the MWU, RWMA, and Hedge algorithms. Another finding is that within the equal weight initialization setting, absolute return-based losses give better performance than price relative-based losses. However, within the market cap weight initialization setting, price relative gives better performance than absolute return. In addition, the coefficients of variation of the MWU and Hedge algorithms were very similar with MWU having a very slight edge. MWU and Hedge performed better than RWMA. High-volatility portfolios had overwhelmingly better returns than low-volatility portfolios, although some low-volatility portfolios with weekly and daily updates did have positive returns.

Overall best performance

Daily update decisions using the Hedge algorithm with equal initial weights and absolute return-based losses gave the best ratio of return to risk, with a coefficient of variation of 332.67. For weekly and daily updates, five portfolios each stood out by yielding a very high return, visualized in Appendix C.

For both weekly and daily updates, the MWU and Hedge algorithms performed very similarly, but with MWU beating Hedge for all scenarios except for high volatility daily returns. The combination of equal initial weights and absolute return-based losses proved to be the optimal one for the weekly and daily MWU and Hedge algorithms, with all of the top portfolios having this setting. MWU with price relative-based losses and RWMA with absolute return losses also make up the 10 highlighted portfolios shown above. As seen in Figure 5, daily portfolios had similar or higher returns than weekly ones while having nearly half the risk, indicating that daily updates were optimal given this particular data.

For the hourly update setting, results were very different; the MWU algorithm run on the high-volatility portfolio with market capitalization-based initial weights was the only one that gave a positive return. For this particular setting, both absolute return-based and price relative-based losses gave a positive return.

One explanation for these results is the comparatively smaller amount of data present for the hourly update setting, containing hundreds of rows instead of thousands. It is possible that there was not enough training data for the algorithm to learn the patterns in stock returns.

Visualizing Algorithm Performance

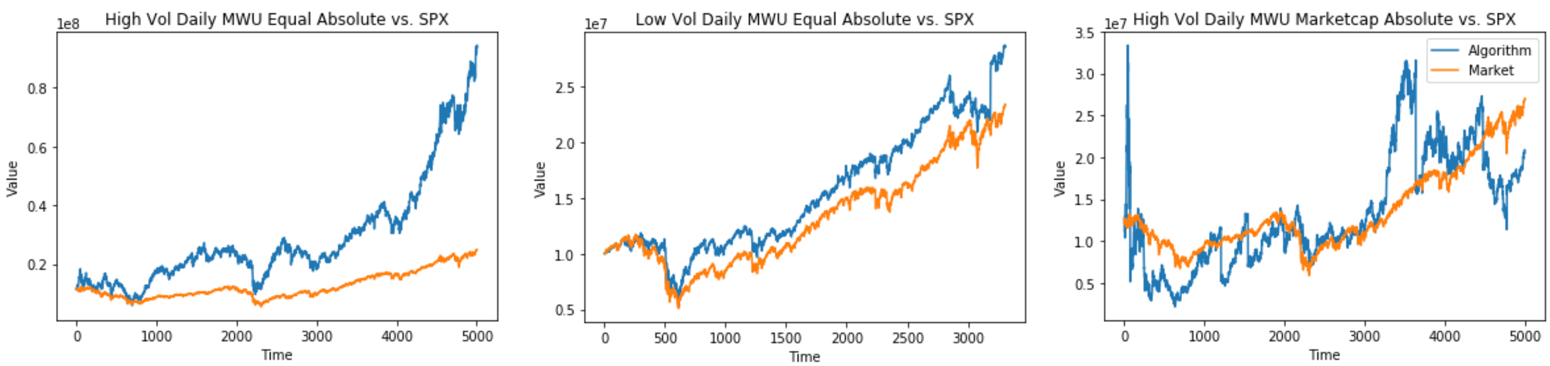


Figure 7
On the left: daily update algorithm on high-volatility portfolio compared to SPX market index; center: daily update algorithm on low-volatility portfolio, right: daily update algorithm with negative returns

The high volatility equal weight portfolio dipped in response to SPX going down, but was able to pull away from the market after time ≈ 2000. The low volatility portfolio stayed close to the market. Part of the difference between the high and low volatility portfolios could also be due to the different amounts of training data for the two algorithms: the high volatility portfolio received > 5000 rows of training data, while the low volatility portfolio had > 3000. This training data effect is also seen when comparing daily and weekly returns to hourly update portfolios. The hourly update algorithms did seem to be learning, but with only around 400 rows of training data, it is possible that there is not enough data to fully learn patterns in hourly returns. A final finding is that high volatility portfolios had the highest positive returns among all tested portfolios and settings, but also produced the lowest negative returns, while the low volatility portfolios had less extreme returns.

Outside of the original algorithms the exponential gradient algorithm also performed very well, as shown above in Figure 8. In both cases out performing the market and the “best” stock in hindsight when using longer periods for each update. Interestingly, on shorter periods the algorithm performed well for the high volatility portfolio, outperforming the market but not the best stock. However, it underperformed with the low volatility portfolio towards the end of the experiment. Past literature on this algorithm showed much closer performance of the algorithm to the best stock however that study limited portfolios to only two stocks. As the portfolio becomes more diversified returns also increased and since in a typical market wealth grows exponentially it can take much greater advantage of these gains.

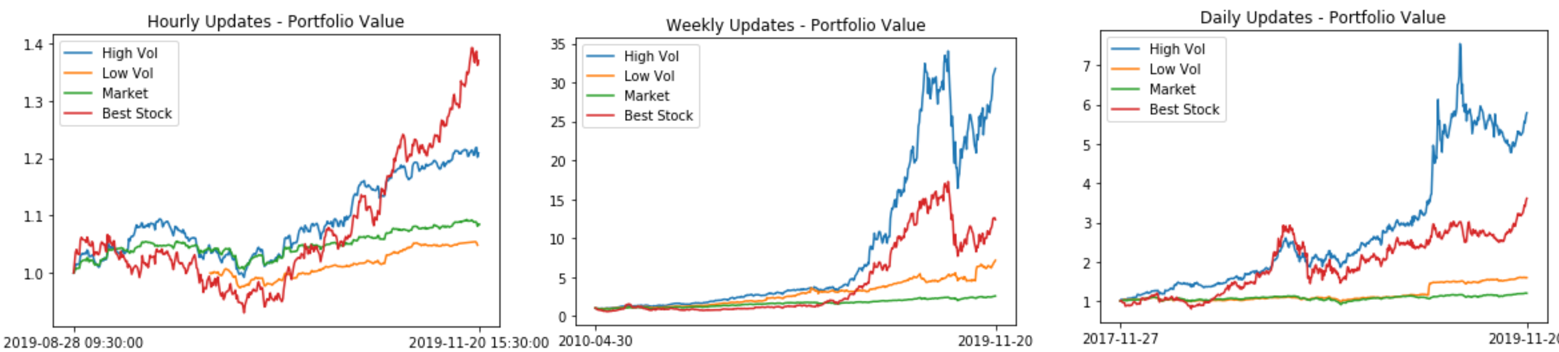


Figure 8

Portfolio values using EG(η) compared to the SPX market index and the “Best” performing stock in hindsight.

Another extension in our study involved outside influences to the portfolio. As mentioned past studies saw limited number of stocks over a period from 1960-1980. Our experiments account for the introduction of new stocks to the portfolio and a more severe stress test in the form of the 2008 financial crisis. On both these fronts we can the algorithm performed well, still maintaining above market and best stock performance seen in the weekly updates in figure 8 and also taking advantage of the high volatility of new IPO seen spike for both portfolio values in the daily updates. One criticism of this algorithm is the run time. When computing the new weights the operation varies exponentially with the number of stocks in the portfolio while our other update schemes grow linearly.

VII. Conclusion

An extension to this analysis is to use portfolio beta (risk level relative to the market's volatility) as a loss value to determine the final diversification of the models. Beta could also be interpreted as a measurement likely to be inversely related to portfolios' return that could reflect the likelihood of a risk-neutral entity to invest in the portfolio. It would also be simple to replicate the experiment to use sector-focused portfolios and then combine to create an optimal portfolio. This is a similar strategy pursued by large investment firms.

One interesting extension would be to formulate a model which could learn the best times to check prices as humans do not make these decisions at regular intervals (once per time period). Alternatively, news and sentiment can drive these human decisions. How could one account for how this sentiment drives price fluctuations?

While working with the training data, the team acknowledged that there may have needed to be more data to fully learn certain patterns. The team had also discussed using live trading data, but there was not enough time to fully explore this idea. Certainly, this analysis must be replicated on different types of portfolios to truly confirm findings.

VIII. Appendix

Appendix A: Test groups of preselected stocks

Lower anticipated volatility (19 total): Bank of New York Mellon Corp (BK), Cigna (CI), J.P. Morgan (JPM), Dow Chemical Company (DD), Colgate-Palmolive (CL), Hartford Financial Services Group (HIG), Aflac Inc. (AFL), Andocs Ltd. (DOX), BCE Inc. (BCE), Berkshire Hathaway Inc. Class B (BRK-B), Coca-Cola Company (KO), Honeywell International Inc. (HON), Loews Corporation (L), PepsiCo Inc. (PEP), Republic Services Inc. (RSG), Proctor & Gamble (PG), Walmart (WMT), Sysco (SYY), Western Union (WU)

Higher anticipated volatility (18 total): AMD (AMD), Celgene (CLEG), ResMed (RMD), Arconic (ARNC), Macy's (M), Nvidia (NVDA), Xilinx (XLNX), Freeport-McMoRan (FCX), Nektar Therapeutics(NKTR), Lam Research (LRCX), Lyft(LYFT), Uber (UBER), Stitchfix (SFIX), Salesforce (CRM), Square (SQ), Beyond Meat (BYND), Pinterest (PINS), Spotify (SPOT)

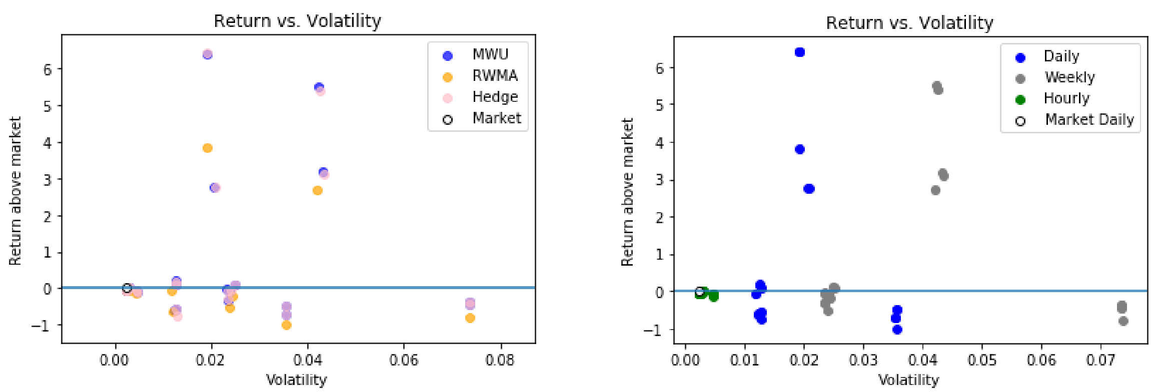
Appendix B: Coefficients of Variation for Top 6 Algorithm Settings

Low volatility, weekly updates		High volatility, weekly updates	
Algorithm and setting	CoV	Algorithm and setting	CoV
MWU Equal weights, absolute return	4.1680	MWU Equal weights, absolute return	129.7217
Hedge Equal weights, absolute return	3.9482	Hedge Equal weights, absolute return	126.8988
MWU Market cap weights, absolute return	-0.9103	MWU Equal weights, price relative	73.1386
Hedge Equal weights, price relative	-2.8091	Hedge Equal weights, price relative	71.0046
Hedge Market cap weights, absolute return	-2.8136	RWMA Equal weights, absolute return	64.1449
MWU Equal weights, price relative	-3.9205	Hedge Market cap weights, price relative	-5.0910

Low volatility, daily updates		High volatility, daily updates	
Algorithm and setting	CoV	Algorithm and setting	CoV
MWU Equal weights, absolute return	15.7619	Hedge Equal weights, absolute return	332.6736
Hedge Equal weights, absolute return	13.8413	MWU Equal weights, absolute return	331.1606
Hedge Equal weights, price relative	7.8828	RWMA Equal weights, absolute return	198.0777
MWU Equal weights, price relative	6.1417	MWU Equal weights, price relative	133.9324
RWMA Equal weights, absolute return	-6.4237	Hedge Equal weights, price relative	133.2163
Hedge Market cap weights, price relative	-44.0079	Hedge Market cap weights, price relative	-13.8455

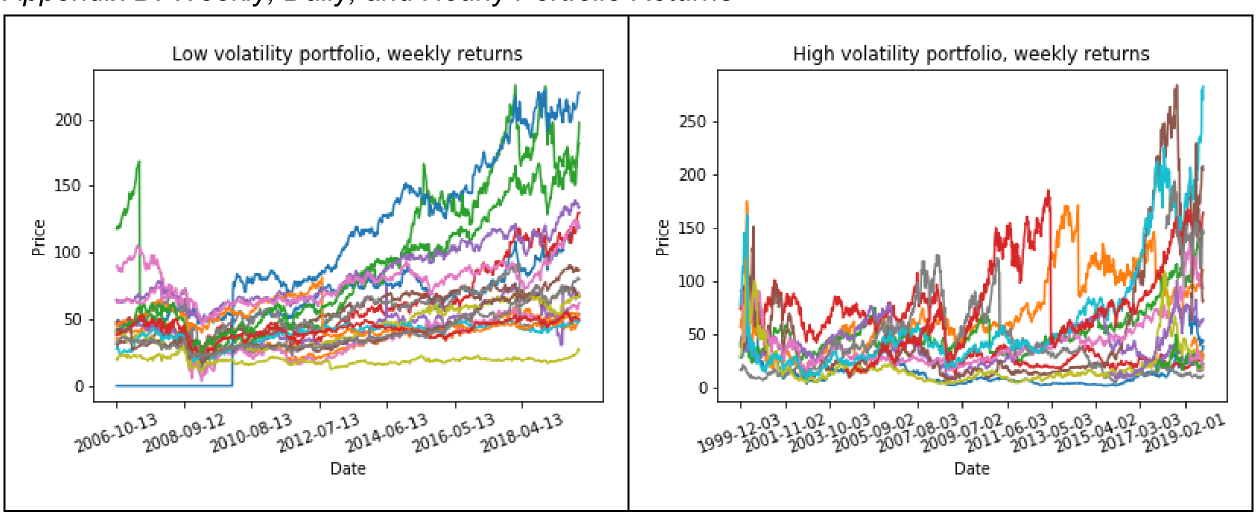
Low volatility, hourly updates		High volatility, hourly updates	
Algorithm and setting	CoV	Algorithm and setting	CoV
Hedge Equal weights, price relative	-19.04	MWU Market cap weights, absolute return	1.4
MWU Equal weights, absolute return	-19.40	Hedge Market cap weights, absolute return	1.3
MWU Equal weights, price relative	-19.44	Hedge Market cap weights, price relative	1.3
Hedge Equal weights, absolute return	-19.84	MWU Market cap weights, price relative	1.3
Hedge Market cap weights, price relative	-22.25	RWMA Market cap weights, absolute return	-13.53
RWMA Equal weights, absolute return	-22.2692	MWU Equal weights, absolute return	-14.54

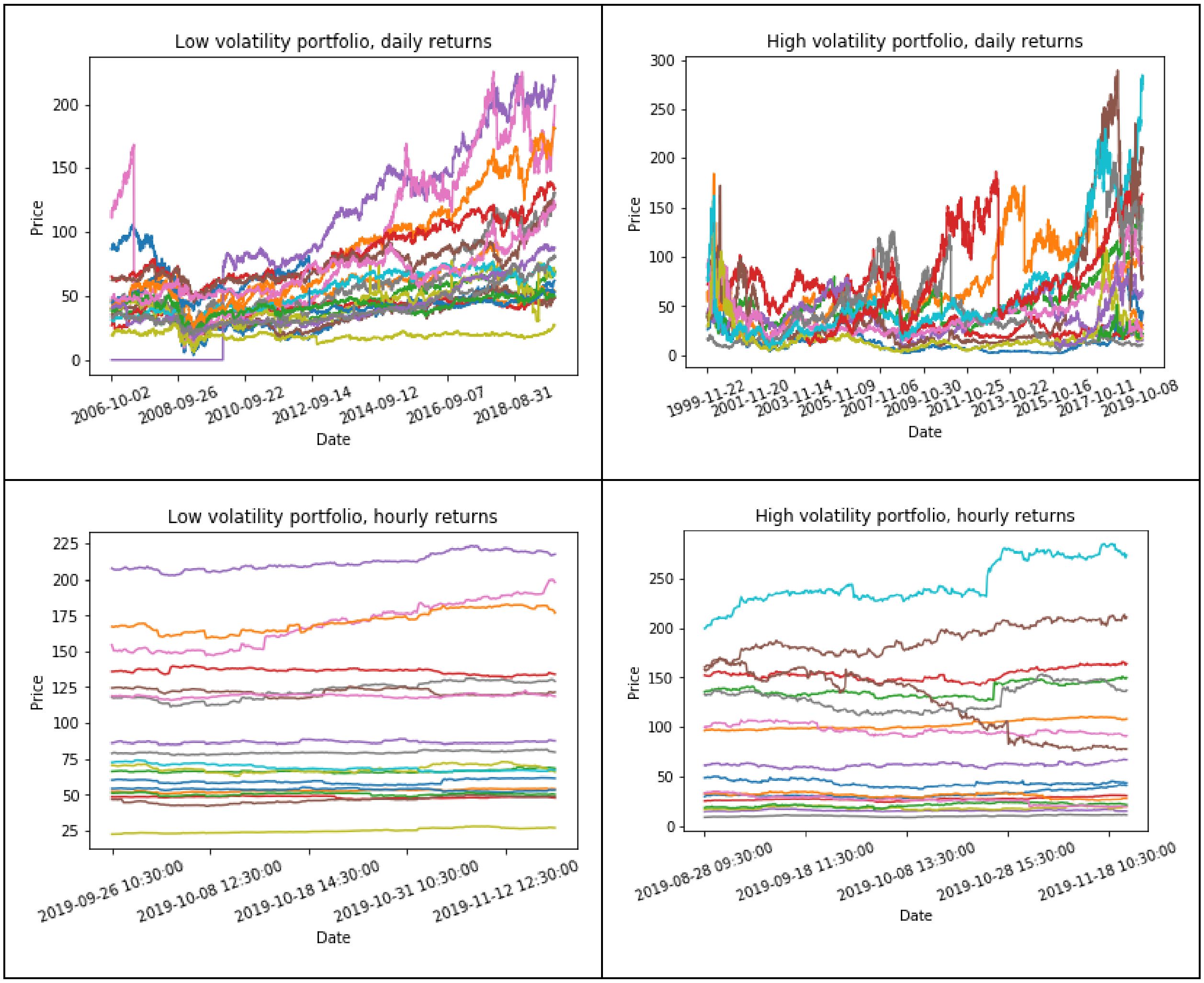
Appendix C:



All algorithms and settings, plotted as return above market vs. volatility
5 portfolios each are highlighted for daily and weekly updates on the right: some points overlap

Appendix D: Weekly, Daily, and Hourly Portfolio Returns





Appendix E: Github repository

https://github.com/sduggirala3/OnlineLearningStockPortfolios