Assessing the performance of an "optimally" weighted portfolio against its benchmark

David Nagy
dpn29@cam.ac.uk

October 2020

Contents

The code used to generate the results for this project is available on my GitHub: `https://github.com/dpn29/Portfolio-weighting-project`.

1 Introduction

Stock indices like the SP 500 or Dow Jones Industrial Average (henceforth DJI) must have a rule describing how the components of the index are weighted. For example, in the SP 500, stocks are weighted in proportion to their market capitalization, while the DJI is price-weighted. The common feature is that the weights are determined by factors (almost) irrelevant to the statistical properties and connections between the stock price time series. For example, if we find two tickers which we believe to be negatively correlated while both have positive expected return, we should be able able to increase the Sharpe ratio of our portfolio by holding more of both of them than we would under the index's arbitrary weighting rule.

This idea is extended to an arbitrarily large universe of stocks and its a mathematical formulation is called Modern Portfolio Theory (MPT) or Markowitz Model after the economist who received a Nobel Prize for formalising this concept.

The goal of this project is to compare the performance of two portfolios: the benchmark portfolio is the DJI index and our "strategic" portfolio will be one that attempts to weight the components of the DJI according to MPT. I found that, with my particular parameters, the strategy portfolio had an information ratio of 54%.

2 Modern Portfolio Theory 101

Let's fix a universe of n tickers. A portfolio over this universe is represented by a vector $w \in \mathbb{R}^n$, w_i denoting the weight if the i-th ticker. Similarly, let $\mu \in \mathbb{R}^n$ denote the expected return on each ticker and $\Sigma \in \mathbb{R}^{n \times n}$ be the covariance matrix of the tickers. Then the expected return on our portfolio represented by w can be written as $w^T \mu$ and its risk (as measured by variance) is $w^T \Sigma w$. So the portfolio-optimisation problem, for a fixed target return $\mu^* \in \mathbb{R}$ is written as

$$\begin{aligned} \min_w \ & w^T \Sigma w \\ \text{subject to } & w^t \mu \geq \mu^* \\ & w^T \mathbf{1} = 1 \end{aligned}$$

where $\mathbf{1} = (1, \ldots 1)$. In addition, one can include constraints such as $w_i \geq 0$ (this represents considering long-only portfolios only).

If we vary μ^*, solve the above optimisation for each value, calculate the resulting minimum variance (or equivalently the minimum standard deviation, aka volatility, σ^*) and plot the resulting (σ^*, μ^*) combinations in 2D space, we obtain the so-called efficient frontier. An example image, due to Robert Andrew Martin, is provided below. Each of the dots represent a different portfolio over the 5 stocks whose names are displayed. The efficient frontier traces the return and volatility of those portfolios that we cannot improve upon in the sense that a higher return is only achievable by taking on more risk.

As risk-averse, return-seeking investors, we should be choosing a portfolio corresponding to a point on the efficient frontier. Which point we choose may depend on our risk-preferences, but if we are also able to choose any amount of leverage, we should be aiming for the point with maximal Sharpe ratio.

The portfolio optimisation problem above is in the category of quadratic programming tasks. However, there have been algorithms specifically designed to solve the portfolio optimisation problem as efficiently as possible. A prime example is the Critical Line Algorithm (CLA), an open-source implementation of which is avail-

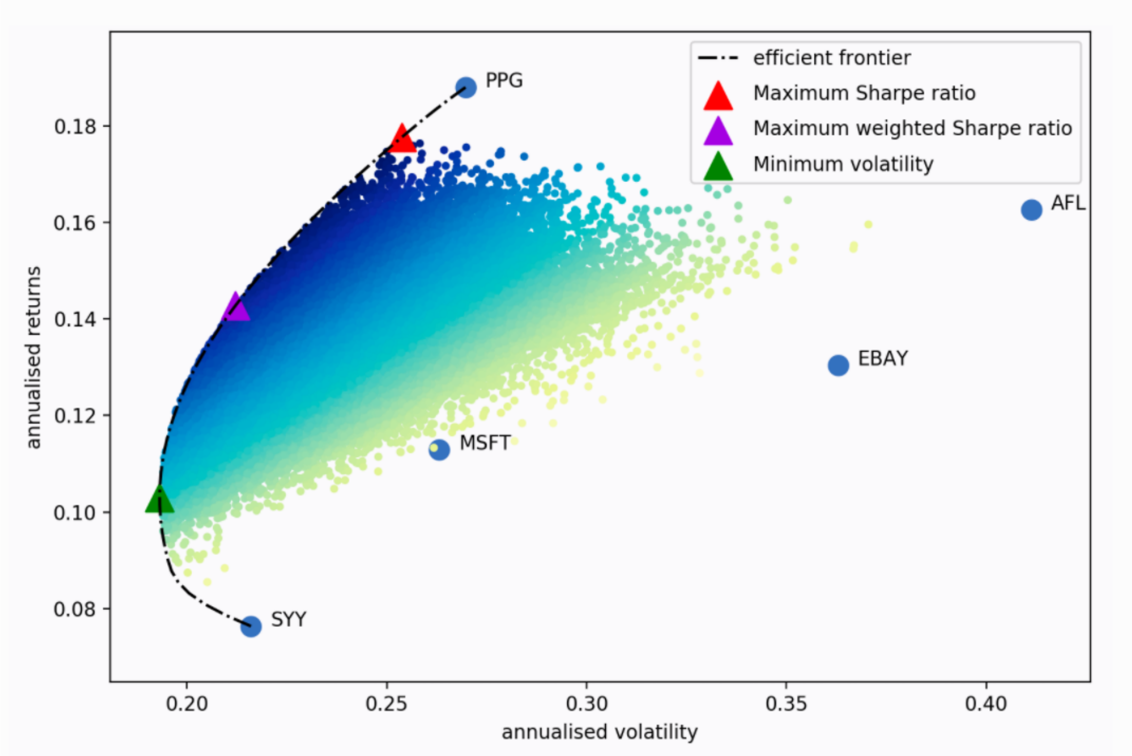

Figure 1: Visualising the efficient frontier

able as part of the PyPortfolioOpt Python package.[1]

3 Parameter choices

Several decisions had to be made about the details of the implementation. A discussion of a subset of these choices follows.

3.1 Choosing the universe of tickers

Choosing the components of a major US stock index as my universe of tickers to optimise over has a number of advantages:

- It is practical to use the index itself as benchmark, rather than coming up with a "benchmark weighting". This way the benchmark is easily interpretable and

[1] PyPortfolioOpt is the best open-source portfolio optimiser available today. It has a well-documented and intuitive API and is built modularly, so that users can define their own models. It was created and is maintained by Robert Andrew Martin, a friend of mine at Cambridge. The CLA algorithm was implemented by Marcos López de Prado and David Bailey. For anyone interested in contributing to a finance-related Python package, I highly recommend checking out PyPortfolioOpt on GitHub.

relatable.

- There exist many ETF tracking these indices, therefore an outperforming strategy would be of direct interest.

- US stocks is the most easily accessible asset class to trade and many funds specialise in this asset class, which hopefully makes this project interesting to more people.

- Data is easily accessible.

First I was trying to use the S&P 500, but I ran into number of problems. Firstly, retrieving price data for historical components did not prove to be easy: some companies that used to be part of the index have gone bankrupt, merged or split and even though my source of data, Thomson Reuters Eikon (courtesy of the Cambridge Judge Business School) retains their data, the name of the ticker changes. Furthermore, data quality was poor in general, with the API not retrieving any, or only partial, data for some of the historical components. Secondly, the composition of the index changes quickly and a company can be selected into the index soon after its public listing, which does not leave enough datapoints to estimate the covariance matrix reliably. Thirdly, the portfolio optimisation is a computationally costly procedure and my laptop started to struggle for a portfolio of 500 stocks. Fourthly, the estimation of the optimal weights will be very unstable for such a large portfolio, as the covariance matrix will contain large (in absolute value) elements which actually only come from statistical noise.

Therefore I changed to the DJI, which resolved all four of the above problems (as it only contains 30 stocks at a time).

3.2 Portfolio optimisation parameters

I simply estimated μ and Σ by method of moments based on historical data. It is common practice to use some kind of shrinkage on the covariance matrix to avoid statistical noise having too much of an impact on the portfolio weights. I estimated 30×30 covariance matrices based on one trading year (about 252 datapoints) and did not use shrinkage. Arguably it would also make sense to use some kind of recency-weighted average instead of simple method of moments, but I neglected these improvements due to time constraints.

I included the additional constraints $0 \leq w_i \leq 0.1$ for all i. This enforces that we are only considering long-only portfolios with at least 10 components.

After the optimiser estimates the efficient frontier, weights corresponding to the portfolio with maximal Sharpe ratio are chosen as the allocation to hold for the next period.

3.3 Rebalancing dates

The more frequently the portfolio is rebalanced, the better results we should get in theory, since the allocation responds quickly to changes in μ and Σ (subject to estimating the parameters on sufficient data). On the other hand, a more frequent rebalancing would imply higher trading costs. Since trading costs are not modelled, I was conservative and decided to choose relatively infrequent rebalancing.

The rebalancing dates were the dates when the composition of the DJI changed, of which there were 10 since the year 2000. So, if d_1 is a rebalancing date and the next one is d_2, the algorithm checks which stocks were in the index between d_1 and d_2 and estimates μ and Σ based on one calendar year of data *before* d_1 for these stocks. Then it comes up with the optimal allocation w, which is the allocation that will be held in the time period between d_1 and d_2. When we reach date d_2, the process is repeated and so on.

3.4 Choice of performance metric

As the goal of active portfolio managers is to maximise their Sharpe ratio, it seems natural to calculate the Sharpe ratio of our strategy and compare it to the Sharpe ratio of the index. The difference between the two Sharpe ratios can be attributed as the alpha of the strategy. However, since the benchmark is very clearly defined and this project is mainly interested in whether the strategy outperforms the benchmark (not the absolute performance of the strategy) it makes more sense to use the *annualised* information ratio, defined as $IR = \frac{\mathbb{E}(R_s - R_b)}{\sqrt{\text{Var}(R_s - R_b)}}$ instead, where R_b and R_s are random variables representing the *annual* (log) returns of the strategy and the benchmark, respectively.

Since my data is in daily resolution, I can estimate the daily information ratio by method of moments as

$$\frac{\bar{r}_a}{\sqrt{\frac{\sum_{i=1}^{n}(r_{si} - r_{bi} - \bar{r}_a)^2}{n}}},$$

where n is the number of trading days in the holding period, r_{si} and r_{bi} are the realised daily log returns on day i for the strategy and benchmark respectively and $\bar{r}_a = \frac{1}{n}\sum_{i=1}^{n}(r_{si} - r_{bi})$ is the realised average active log return.[2] Since the expected total log return over a period scales linearly in n and volatility scales linearly in $\sqrt{n}$,[3] my estimate for the strategy's annual information ratio is

$$\widehat{IR} = \frac{\bar{r}_a\sqrt{n}}{\sqrt{\frac{\sum_{i=1}^{n}(r_{si} - r_{bi} - \bar{r}_a)^2}{n}}}.$$

[2]Note the usage of n, rather than $n-1$, in the denominator for the estimator of variance. Using $n-1$ would give an unbiased estimator, while using n gives the MoM estimator, which is also the maximum likelihood estimator under the assumption that returns are lognormally distributed. The difference is negligible since I have at least 150 datapoints in each period.

[3]This is true under the assumption that portfolio returns follow a Wiener process, which is disputed but a good approximation.

4 A word on parameter selection

With any quantitative trading strategy, we are ultimately interested in out-of-sample performance. When a model has many parameters we optimise over, the training performance will be a bad (too optimistic) indicator of out-of-sample performance. When there is sufficient data, this issue can be tackled by separating our data into a training, a cross-validation and a test set: train the models on the training data, and select a model based on the cross-validation data. The test data is only used to access the generalisation, i.e. out-of-sample performance of the model.

I have not implemented such cross-validation, the reason being that I decided the implementation details reported in Section 3 before seeing any results. In other words, I did not optimise over any hyperparameters, so the results reported in Section 5 can be taken as point estimates of out-of-sample performance, coming from an *unbiased* estimator.

5 Results

The backtest of the strategy produced the estimate $\widehat{IR} = 54\%$. This is based on data from Sept 2000 - Sept 2020, with the first component change of the index (so the first position of the strategy) occurring in April 2004. So the Great Recession and the Covid crisis are included in the sample, but the dot-com bubble is not.

The result is definitely encouraging. Information ratios of 40-60% are considered good from active portfolio managers. The statistical significance of the result is quite hard to gauge however, as constructing confidence intervals requires statistical techniques that are beyond the current scope. However, breaking the sample period into smaller subperiods (e.g. each holding period) and looking at the statistic calculated for those subperiods increases my confidence in the positive result.

	Days in period	Strat return	Strat vol	DJI return	DJI vol	Information ratio
2004-04-08	971.00	15.78	17.73	4.42	11.99	76.68
2008-02-19	150.00	-19.78	20.26	-17.46	23.18	-13.92
2008-09-22	178.00	-41.46	50.84	-30.98	46.01	-39.05
2009-06-08	832.00	30.50	19.74	14.13	16.87	86.02
2012-09-24	249.00	22.31	15.83	13.59	10.71	58.09
2013-09-23	374.00	43.16	16.80	10.67	11.38	198.71
2015-03-19	620.00	17.55	17.74	8.29	12.44	61.66
2017-09-01	204.00	46.82	22.50	13.30	14.61	148.90
2018-06-26	192.00	31.60	27.28	10.55	16.50	85.33
2019-04-02	357.00	6.45	30.00	5.77	31.06	2.20

Figure 2: Annualised statistics in percentage for each holding period

The information ratio was only negative for the two periods that overlap with the 2008 financial crisis. It was close to zero for the period containing the Covid crisis. It seems like the strategy can be expected to do very well in normal times, with larger drops than the benchmark in times of crisis. This might be explained by the fact that the strategy favours riskier (growth) stocks as these are likely to have had higher returns, leading to higher estimated returns and higher weights in the allocation. But when a crisis hits, the increased uncertainty has more of an effect on these riskier stocks. The outperformance of the strategy comes from the fact that it makes use of correlations efficiently, to be able to invest in riskier stocks without taking on too much risk.

Finally, I produced the graphs in Figure 3 and Figure 4 for visual comparison.

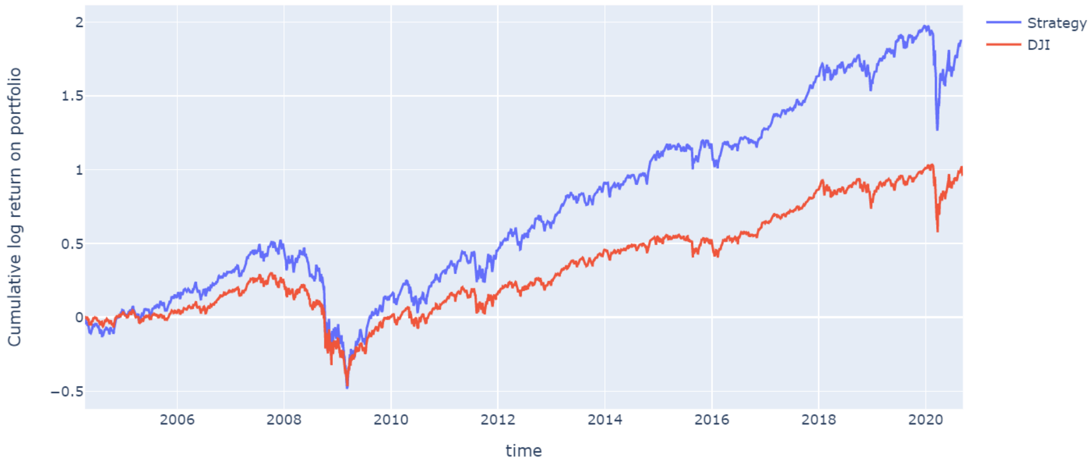

Figure 3: Cumulative log return for the strategy and benchmark. In each holding period, the volatility of the strategy portfolio has been normalised to match the volatility of the benchmark. Therefore the two returns are on portfolios of the same risk level (where risk level is measured by standard deviation of log returns)

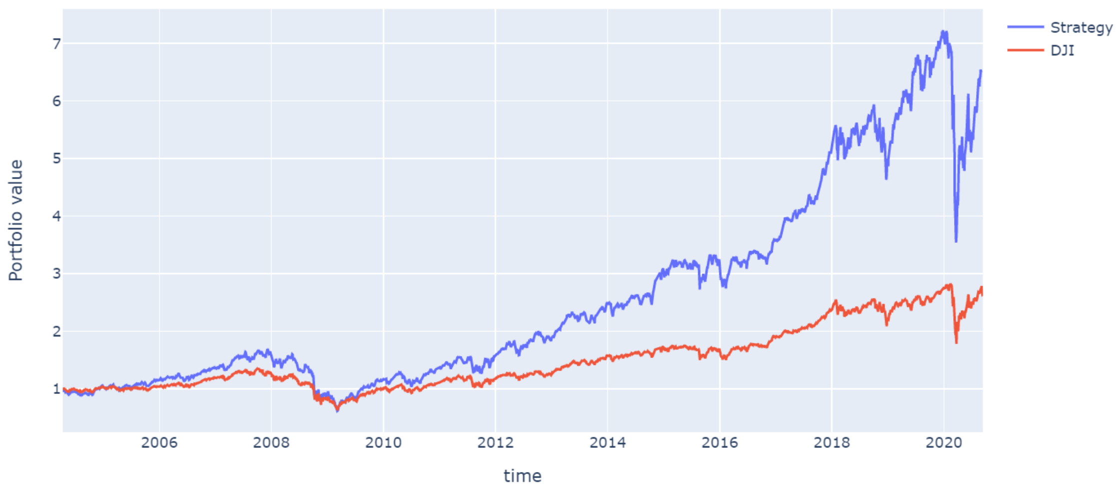

Figure 4: Showing the portfolio values over time for an initial investment of 1. The graph was generated by turning Figure 3's log returns into actual returns

6 Limitations and directions for extension

I consider the following extensions as meaningful improvements to this project. Due to other commitments I do not have to implement them, but an suggestions or queries are welcome at my email, dpn29@cam.ac.uk.

- Extend the analysis to multiple stock indices or asset classes.

- Experiment with different weighting methods for the estimation of the expected returns and the covariance matrix and with different constraints for the optimisation.
- Identify the statistical significance of the results.
- Model trading costs.

7 Conclusion

The aim of this project was to assess the performance of a portfolio selected by mean-variance optimisation against its "arbitrarily" weighted benchmark. For the DJI index, the optimised portfolio had an estimated information ratio of 54%.

This result is equivalent to saying that, if two portfolios were started with the same capital in 2004, one tracking the DJI index and the other following the strategy described above with leverage so as to match the volatilities of the two portfolios, the strategy portfolio would have increased in value more than 6-fold, while the benchmark portfolio would have increased only 3-fold, as of mid-2020. This is substantial outperformance.

While the statistical significance of this result is hard to assess, the time period studied spans 16 years and two major stock market crashes, suggesting robustness. Modelling trading costs would decrease the performance as the strategy portfolio requires more transactions than the benchmark. However, this is unlikely to be excessive, as the strategy portfolio is reweighted on only 10 occasions during the 16 years.

In conclusion, it is suggested that portfolio weighting optimisation has substantial benefits.

8 Disclaimer

This content is being provided solely for information purposes and is not intended to be a solicitation or offer, recommendation or advice to buy or sell interests in any security or investment product mentioned in it, to effect any transaction, or to conclude any transaction of any kind whatsoever, in particular to any recipient who is not a qualified, accredited, eligible or/and professional investor. Whilst we shall use reasonable efforts to obtain information from sources which we believe to be reliable, we assume no liability regarding this content and give no warranty as to the accuracy, completeness or reliability of any mentioned data and thus assume no liability for losses arising from the use of this content. The information, opinions and assessments contained in the present document shall apply at the time of publication and may be revoked or changed without prior notice.