Trading Algorithm Project

Tharun Polamarasetty

AUGUST 6, 2020

Under the guidance of:

Dr. Tao Pang

Graham (QMS Capital)

Contents

1. Introduction

I have created an algorithm that outperforms the S&P 500 consistently and to test this, we used a backtest from 2004 to 2019. The algorithm is based on the idea presented in the two research papers, and then I enhanced the algorithm for better returns. The backtest results from 2005 to 2019 give 373% returns with positive skew, Sharpe ratio of 2.3, alpha is 0.24 and beta is 0.3 which means the algorithm is much stable with low volatility but better returns. The returns are just for 8 months a year, but if we move the free cash to the benchmark, it shows a return of 489% returns, whereas the benchmark during this period returns 345%. The reported results are post commissions and slippage.

2. Algorithm Idea

The preliminary idea is to replicate the paper Haigang Zhou and John Qi Zhu. "Jump on the Post-Earnings Announcement Drift." by using a non-parametric jump model using the 16-day bipolar variation to offset volatility and pass through the 1% significance level. Next, we check the log-returns if it's positive, the algorithm takes Long position, else if the returns are negative, the algorithm takes a short position. Then, we pass through 700 companies sorted by market cap, with sufficient liquidity, daily volume, A-class share price for each security, and few other factors to ensure that the security is tradable with enough capital without too much slippage while taking or exiting the position.

2.a Non- Parametric Jump model

L Statistic

$$\mathcal{L}(i) \equiv \frac{\log S(t_i)/S(t_{i-1})}{\widehat{\sigma(t_i)}},$$

where

$$\widehat{\sigma(t_i)}^2 \equiv \frac{1}{K-2} \sum_{j=i-K+2}^{i-1} |\log S(t_j)/S(t_{j-1})||\log S(t_{j-1})/S(t_{j-2})|.$$

Here S(t, i) is the price at time t. This algorithm takes the daily prices, and the value of K is 16. So, we can decide the Jump is not in the usual volatility jumps of the security. K, the window size within which the corresponding local movements of the price process are considered, is chosen to eliminate the effect of jumps on the volatility estimation. Under the null hypothesis of no jumps at time t. Under the alternative hypothesis of having jumped at time t, where sqrt(2π) ~ $N(0, 1)$ as L tends to infinity. To accommodate our return data at a daily frequency, we set K equal to 16 and specified the rejection region at the significance level of 1%, following Lee and Mykland (2008)

Selection of rejection region:

$$\frac{\max_{i \in \bar{A}_n} |\mathcal{L}(i)| - C_n}{S_n} \to \xi,$$

where ξ *has a cumulative distribution function* $P(\xi \leq x) = \exp(-e^{-x})$,

$$C_n = \frac{(2 \log n)^{1/2}}{c} - \frac{\log \pi + \log(\log n)}{2c(2 \log n)^{1/2}} \quad \textit{and} \quad S_n = \frac{1}{c(2 \log n)^{1/2}},$$

where n is the number of observations.

$(|L(i)|-Cn)/Sn$ is $\beta*$, and the cumulative distribution function $P(\xi \leq \beta*) = \exp(-e-\beta*) = 0.99$. Since the significance level is 1%, $\beta* = -\log(-\log(0.99)) = 4.6001$. Therefore, if $\beta* > 4.6001$, then we reject the hypothesis of no jump at *ti*.

2.b Normalized Factor

However, the returns were like 3% so, I modified the algorithm by adding a new filter with normalized differences of consensus estimates and the actual quarterly reports. This classifies into 20 deciles and across numerous years, we have backtested and gave a fixed number for the normalized factor. After the original quarterly report, if the number is larger than a fixed upper limit, the algorithm enters into a long position, similarly with a lower limit and short position.

3. Quantopian

I used the Quantopian platform to do both the research and write the code is Algorithm IDE where both platforms are constructed using python

3.a. Research notebook

The Quantopian comes with a plethora of inbuilt functions, most of the research is used based on this Quantopian's notebook. We received the fundamentals data from Morningstar and FactSet data. However, we are restricted to use the data up until March 2019. Due to this limitation and memory constraints, we had to use 700 US securities sorted through market cap and passed through 6 other filters, So that the securities are tradable liquid, dividend-adjusted, A-class shares.

3.b. Algorithm IDE

The inbuilt functions are constructed on top of python after the limitations from the research notebook are successful, the IDE doesn’t have any memory issues, but it has a whole new level of inbuilt functions, we scheduled the functions to trigger whenever we get the earnings release date. Then we meticulously assign weights and rebalance portfolios, since the earnings reports occur at different dates for each company and they are dynamic. So our portfolio is made dynamic.

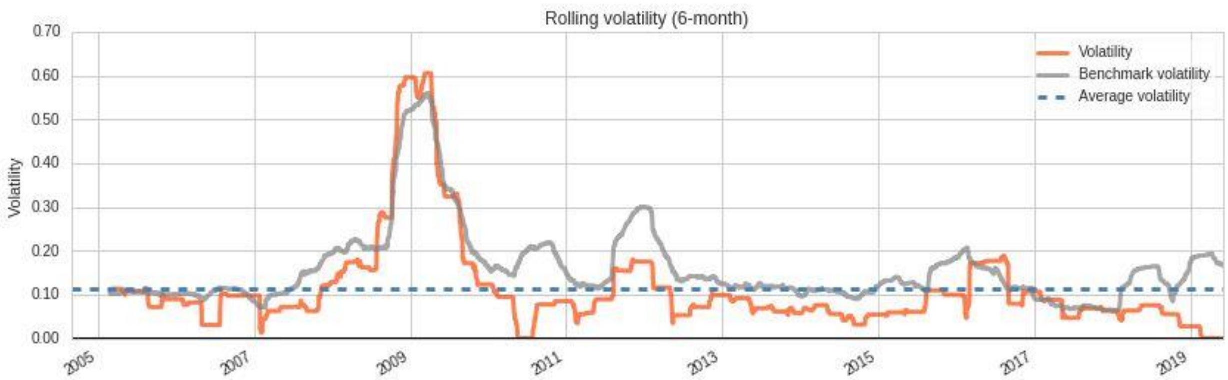
4. Strategy Performance

4.a. Cumulative returns

The algorithm performs 379% returns over the past 15 years while the S&P 500 gives a return of 342%. The best part with the eyeball test we can see that it consistently outperforms the benchmark index with time. We can also see that it depends on the point of entry time. The algorithm is stable with a decent Sharpe ratio of 2.4 and a positive alpha, which signifies the merit of the algorithm.

Since we used Quatopian's algorithm IDE to backtest, we had data until 2019 first quarter, however since the algorithm is based on the shocks in the quarterly reports, testing due to COVID-19 would have been an interesting watch but, we couldn't get the current data so we couldn't test. Further details are discussed in the performance in crisis situations.

4.b. Rolling volatility



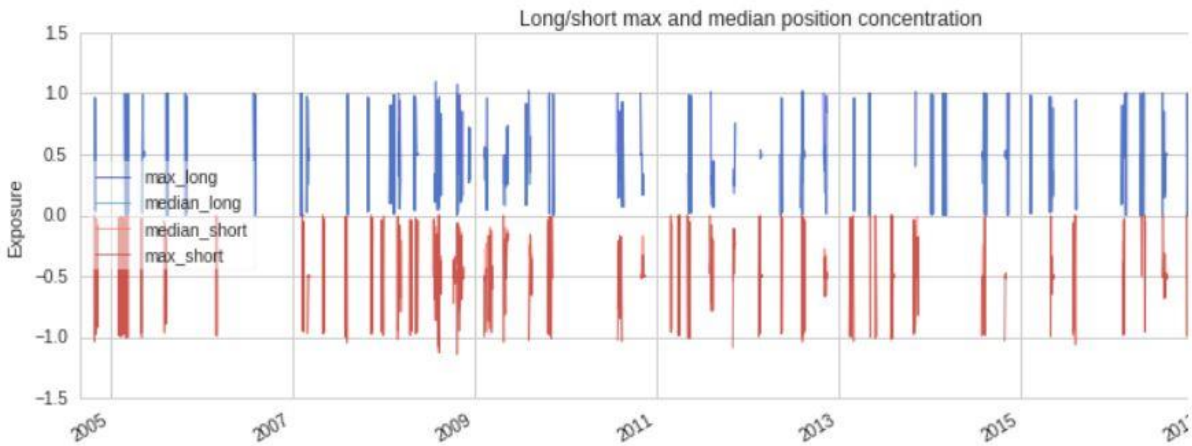
This chart shows how the volatility changes relative to the benchmark S&P 500 index with time for the algorithm. From this, it’s evident that on average the volatility of the algorithm is much better than the benchmark. Although the volatility during the great financial crisis is….. overall volatility is 11%. For the last 5 years, the volatility is just 3.5%. This shows that the algorithm is much stable and the variance of the returns is in controllable range.

4.c. Worst drawdowns

Worst drawdown	Net drawdown in %	Peak date	Valley date
0	27.37	2008-12-04	2009-02-23
1	23.41	2008-07-28	2008-08-19
2	17.98	2012-02-23	2013-08-12
3	13.49	2007-08-10	2007-11-12
4	8.46	2011-08-17	2011-11-04

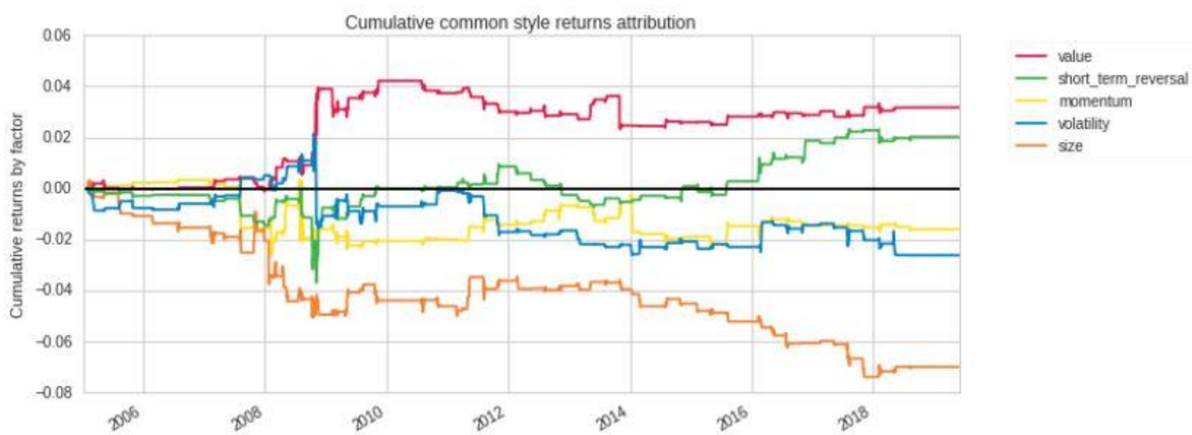
This table shows the portfolio net drawdown for that quarter, Although the 2 quarters from the time of the financial crisis were disastrous, the other two quarters had a great performance. Overall, it had a net positive year with 60% returns.

4.d. Net exposure to the market



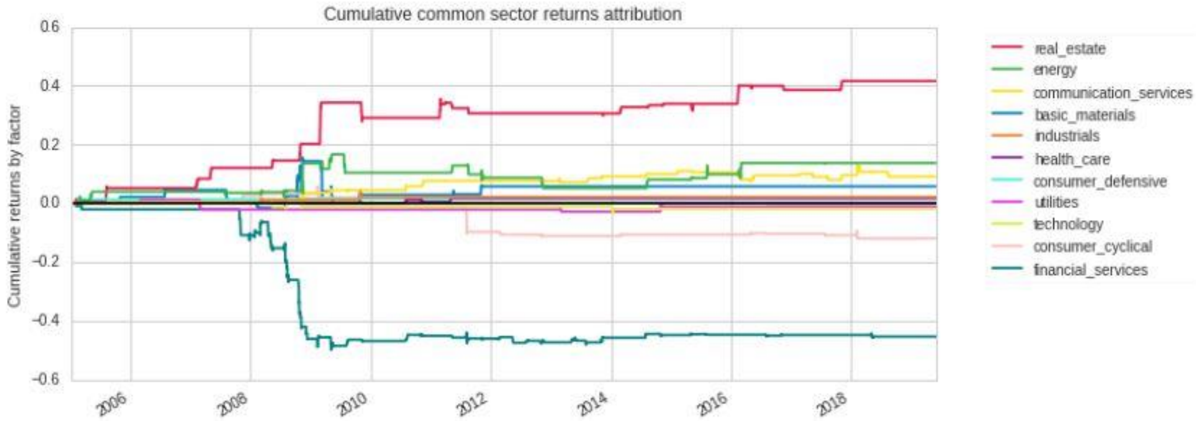
In order to limit the overall market risk to the direction of the market, the portfolio has a good balance with long and shorts. Since the strategy is based on market shocks, typically the normalized factor from the expected vs original reported quarterly reports, during the turmoil times, the algorithm appears to perform better. Since the net exposure is not biased in a single direction, the risk is relatively controlled, and as we have seen that the best returns are from both longs and shorts.

4.e. Value-based strategy



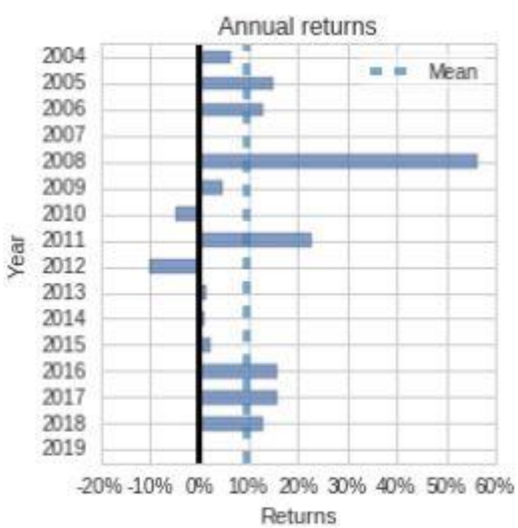
The algorithm has been constructed against typical investing styles and from the chart, we can see that most of the returns come under the value-based strategy. This is expected because our positions are based on shocks from the fundamental values of the quarterly reports and on the intuition that strong returns continue to go upon the strong earnings report whereas the week returns continue to go down, at least up till the expectations of the next quarter coming up. Also, part of the profit comes from short term reversals strategy, the reason being that on the report may be in opposition to what the wall street census which leads to short term reversals.

4.f. Sector analysis



During 2008-2009, Global Financial Crisis, the algorithm has maximum returns from the real estate sector and took the biggest hit from the financial sector, the initial idea was to remove the financial sector, but the again considered it because the next crisis may take a hit from other sectors. So, there is no bias towards a sector and apart from the GFC, more or less other sectors performed on an annual return of -10% to +15%

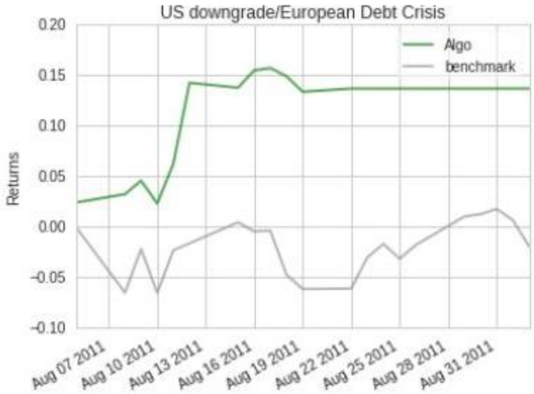
4.g. Annual Returns



The algorithm performs consistently over the years, however, the low volatility during 2013-2015 had significantly fewer positive returns. 2008 has the worst drawdowns but the other two quarters outperformed and had a good return for the year, but the returns reflected in the 2009 records.

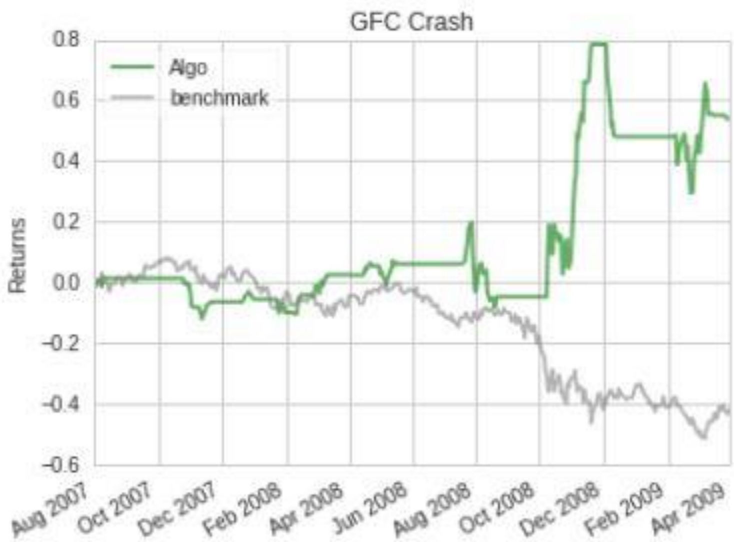
4.h. Algo performance in worst economic phases

US downgrade/European debt crisis



The grey line indicates the benchmark index performance during the US downgrade/European debt crisis and we can see that we have positive returns around 15%, because the algorithm has a healthy mix of both Longs and shorts.

Global Financial Crisis



The grey line indicates the benchmark index performance during the Global Financial Crisis, on the other hand, the green line represents the algorithm performance. We could see that the maximum drawdown is during the period 2008 Aug- Oct, however, based on the net returns, the quarter is 7% down. Since our algorithm is based on earnings shocks it performed very well because the algorithm is mixed with healthy longs and shorts, 2009 is the year that gave the majority of the returns for going long on good stocks and taking a short in the bad stocks.

Low volatility bull market

Before GFC, we have one of the best bull runs, however, there are enough shocks from the quarterly reports for us to take advantage of the fundamental analysis. The green line signifies the algorithm performance over time and the grey line represents the benchmark index performance. As we can see that the algorithm outperforms the Index.

5. Further improvements

Include short term strategy for 4 months:
The algorithm holds for 60 days and exits the position, whereas the money would be free for the next 30 days as the quarterly reports are released over 90 days, So effectively that the money would be staying as cash for nearly 4 months. Therefore, we need to find a way to invest for the short term so that the cash is in the play.

– Invest free capital in S&P500 and exit when needed
  We could invest the free capital in the S&P 500 index so that the returns would not loose against the index, and we could pull whenever needed for the trades based on quarterly reports.

Model for weights instead of passing stocks through filters
– Instead of passing the securities through step by step filters, we could weigh in as factors like Fama French model, by using weights instead of using filters.
– This makes it a Supervised regression machine learning problem, as we can get the weights of the distribution from the regression analysis

– We have had the fixed weight for each security, but the algorithm should use Portfolio Optimization Techniques, for moving around the weights effectively.

6. References

Haigang Zhou, and John Qi Zhu. "Jump on the Post-Earnings Announcement Drift." Financial Analysts Journal 68, no. 3 (2012): 63-80.

Lee, Suzanne S., and Per A. Mykland. "Jumps in Financial Markets: A New Nonparametric Test and Jump Dynamics." The Review of Financial Studies 21, no. 6 (2008): 2535-563.

Abarbanell, J. S., & Bernard, V. L. (1992). Tests of analysts' overreaction/underreaction to earnings information as an explanation for anomalous stock price behavior. The Journal of Finance, 47, 1181–1207.

https://www.quantopian.com/posts
https://www.quantopian.com/docs/index