Barbell Trading Strategy: Capitalizing on Mispriced Black Swan Probabilities

Introduction

In the book "Black Swan" written by Quant Trader Nassim Taleb, he highlights how the notion of "black swan events", characterized by their perceived rareness and unlikelihood, happens more often in capital markets than participants are led to believe. By believing in the notion of Gaussian distribution and the bell curve when it comes to price volatility in equities, investors essentially discount the notion that a major event could possibly occur to his/her portfolio.

In order to capitalize on the randomness and probabilities of large movements in the equity markets, along with being able to hedge the risk of black swans, Taleb offers what is known as the "barbell strategy". This essentially means allocating a majority of the intended portfolio into the safest investments possible, while investing the remaining small minority of capital into "very risky" investments with unlimited upside potential. This strategy looks like a barbell as both extremely safe and extremely risky holdings are included, but none that fall within the middle. By doing this, the investor is able to protect his / her holdings from large negative movements in the market while also allowing for exposure to infinite positive upside. The objective of this report will be to highlight practical implementation solutions that allow an investor to trade using the "barbell strategy" and how to build such systems within an institutional setting. The report will begin with an abstracted view of the barbell strategy and why black swan events are discounted from traditional financial statistics, an analysis of the technical and software implementation opportunities to enable trading of this strategy, and a conclusion.

Abstract

In order to begin understanding why focusing on few, large movements in markets can make for a profitable strategy, one must begin to understand the nature of stock market returns. Data collected by JP Morgan illustrates the idea of missing out on the best days in terms of market returns [1]. By missing out on just the 10 best trading days from 2000 to 2019, a portfolio that is entirely invested in the S&P 500 would have lost half of its returns. This ties into the idea that returns in the market are asymmetrical, therefore it cannot be accurately measured using standard deviations. Thus, market returns are viewed as “extremistan”, where the perceived “unlikely” events skew return averages and actually make up most of the gains made by the stock market over the decades. Therefore, by focusing on capturing value from unlikely movements, a large bulk of returns can be made.

Taking this idea, we can then utilize the barbell strategy to capitalize on high movements while protecting downside risk. For simplicity and practicality, it is assumed that the barbell strategy will utilize a portfolio allocation mix of 90% to risk-free assets, such as US Treasury bonds, while 10% is allocated to “high-risk” holdings, which will be long-dated North American options in this analysis. With this portfolio, the theoretical maximum capital that can be lost is 10% (assuming the 10% towards options all expire worthless). This is a safe assumption to make since if our risk-free assets, US treasury bonds, were to lose immense value, that effectively means Western civilization’s entire financial system might be at jeopardy.

Thus, creating a portfolio like this will limit portfolio drawdowns due to negative black swan events that go against our positions, while also having uncapped potential to be exposed to positive black swan events. Therefore, this strategy is designed to amplify returns during times of high price volatility. The 90% of risk-free assets is the easy part of the strategy and quite straightforward. Simply investing the large majority of the portfolio into risk-free will protect capital and provide liquidity for the high-risk positions. Now, the “high-risk” long dated option position will be explored in more detail and why it is a winning trading strategy.

The mathematics behind how and why this strategy would theoretically work was derived by Nassim Taleb, through who first created the idea of the barbell strategy and inspired this implementation of it. Firstly, the Black-Scholes formula, the pricing model which gives the “fair value” of what an option on an underlying security should cost, utilizes the assumption of a lognormal distribution of price along with constant volatility.
Figure 1. Black-Scholes Equation

$$C(S,t) = N(d_1)S - N(d_2)Ke^{-rT}$$

$$d_1 = \frac{ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

$C(S,t)$	(call option price)
$N(\,)$	(cumulative distribution function)
$T = (T_1 - t)$	(time left til maturity (in years))
S	(stock price)
K	(strike price)
r	(risk free rate)
σ	(volatility)

Note. Descriptive breakdown of the Black-Scholes model
To price a call option [5]

However, this is inherently wrong since the stock market operates under extremistan rules, therefore large movements are not considered into the realm of possibility within the distribution model and the possibility of a black swan event occurring is nearly null, when in actuality the stock market itself made the majority of historical gains on a few trading days of high price movements. The lognormal distribution that the Black-Scholes model operates under is inherently flawed, as it is a Gaussian distribution variation where probability exponentially drops off as you move further from the norm, heavily discounting the probability of large price movements. Thus, traditional risk measures based on Gaussian distribution don't properly gauge the effect of large price movements in the stock market (cite), and buying certain options can capitalize on the events that the lognormal curve thinks is "impossible".

Furthermore, specifically buying long dated options has the effect of giving more time to allow for large price movements to occur. This, combined with the fact that theta decay has less of an effect on long-dated option price decline (accelerates as option approaches expiry), maximizes this strategy. Moreover, for practicality, it is assumed that the options will be based on underlying ETF securities. This assumption is fair to make since options on ETFs such as whole-market funds (e.g SPY) have great liquidity. Thus, the strategy for the options-side of the barbell will be to buy long-dated out-of-the money options based on underlying total market ETFs which have low implied volatility, which means the market is pricing the event as unlikely and the premium per option contract is cheaper. Moreover, the lower premiums means that the risk/reward ratio becomes higher as there is lower capital loss in case of options expiring worthless. Based on the aforementioned Gaussian property of option pricing, the payoff from the unlikely out-of-the money option benefits disproportionately compared to the price to buy the contract, given how it's basically "impossible" for the option to be in-the-money (Gaussian curves such as lognormal make events further from the "norm" have extremely small percentage chance of occurring).

Analysis

1. Technical Implementation

The technical implementation of this strategy includes the topics such as the specific assets that will be included in the portfolio and best environment for strategy implementation.

As mentioned previously, approximately 90% of the portfolio will be allocated to risk-free assets, which will be US Treasury Bonds in this case. For liquidity purposes, the specific treasury bond that will be purchased is the 10-year treasury. This bond is tied to the US interest rates set by the central bank, who are expected to hike rates towards the end of 2022 / early 2023 [3]. Currently, in a low-rate environment with 2021 average yield hovering around 1.46% [4] and with the typical rate in the last 10 years around 2-3%, this portion of the portfolio will not reap large returns (that part is for the 10%), however, will generate stable and predictable returns.

For the options portion of the portfolio, the ideal characteristic for the option contract would be an option with an implied volatility (premium over fair value) of < 5%, expiration date at least 6 months out, and out-of-the money. The general characteristics of the ideal option were already highlighted within the "abstract" section, and now those characteristics are quantified. Thus, within the software implementation of this strategy, the "bot" responsible for the execution of trades must ensure, given the correct data, that the option positions that are being opened follow this guideline.

Although this strategy can be implemented during any economic and financial environment, this strategy is best used during times of low implied volatility and high interest rate environments. Firstly, times of low implied volatility means that option pricing is closer to fair value (as indicated by black-scholes), which means that less of a premium needs to be paid.

Moreover, the max value at risk in terms of the options portion of the portfolio will be reduced as the premiums paid per contract, assuming they expire worthless, will be a smaller portion of the overall portfolio.

Within a high interest rate environment the barbell strategy will also thrive. Given that 90% of the portfolio is allocated to US Treasury bonds, a high interest rate environment means that bond yield will also be higher. Thus, the majority of the portfolio will have greater yield, making it easier to bankroll the 10% of capital allocated to option positions and overall generating higher returns.

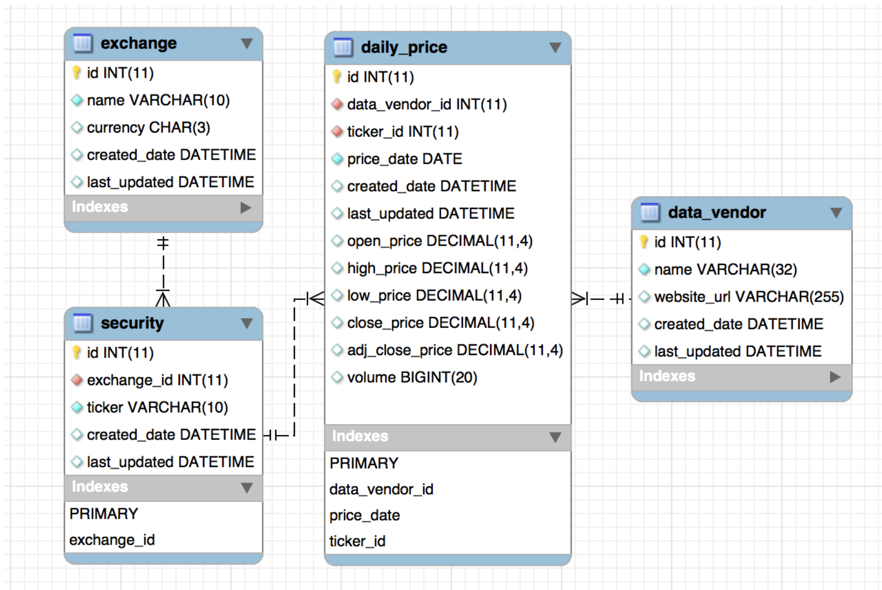
2. Software Implementation

Within the software implementation of the barbell strategy, many aspects have to be considered, including: price feed databases, front-end user dashboards, back-end templates and calculations, and actual bot execution of the strategy. To start, the building blocks of our software can begin with using Python and / or C++ which are industry standard programming languages in finance for a reason. By providing readability and efficient runtimes along with extensive libraries related to data science and finance, Python/C++ languages offer the easiest way to build our software which is what we will utilize as our coding languages. Moreover, for the front-end dashboard that will be displayed for traders who look to utilize this strategy, leveraging the use of front-end frameworks along with Javascript, and HTML/CSS allows for enhanced user experience and greater interactivity.

Price feed databases: To start, creating a relational database where each row represents a specific financial product (bond/option) and its respective data/metrics makes the most sense. Inherently with financial data, there are one to many relationships within the dataset (one product, multiple metrics), therefore there should be individual rows with multiple columns for data which correspond to one product (which can be differentiated through the use of unique IDs). Thus, this can be accomplished through the use of a SQL-based database that will be continuously updated with price history.

In terms of getting the price data, many different API implementations can be used in order to retrieve the data in the form of objects / dataframes.

Figure 2. SQL Database Schema



Note. Sample Database layout for stock data [2]

Solutions such as Yahoo Finance API and Bloomberg API allows for the extraction of data measurements, along with many other free / SaaS solutions available. After retrieving the data and setting up a feed into the relational databases, cron scripts can be utilized so that the price feed script can be run and the data updated throughout the trading day. Overall, this implementation method allows for the extraction of needed data for execution and viewing and ensures that data is updated as needed.

Front-End dashboards: For an institutional firm to utilize this strategy and its software, it must have a scalable front-end dashboard so that multiple traders can see the implementation in action and track movements of the bot automation to ensure that the strategy is being implemented correctly and to detect unusual trading activity. Thus, a user-friendly dashboard where traders can see price action, along with trading activity and overall portfolio statistics is needed.

In order to create a front-end dashboard, either a framework approach and / or build from scratch approach would work. In terms of using a framework, javascript implementations such as React.js and Angular.js would work to create the visual dashboard display for our strategy. These frameworks allow for the displaying of backend data and can be leveraged with the use of HTML/CSS together to create interactive components that would allow traders to use pricing data in real-time and monitor / execute trades as needed. Moreover, instead of using a framework, the front-end dashboard can still be created using just plain javascript and HTML/CSS, however it would just take a lot longer to develop and deploy.

Back-end schema: Depending on the institutional firm, different ways of inputting and retrieving data can be implemented into the backend of our trading systems. However, since no information is given on this topic, it is assumed that the backend can be composed from scratch. Thus, to connect the price data from our database to be displayed to our traders as a front-end dashboard, our back-end implementation will work to essentially retrieve the needed data and post it to our front-end framework, where it will be available to traders. Thus, the solution would be to utilize a backend framework for the task, which can be the python-based backend framework Django. Following the Model-Template-View schema, Django allows for the easy solution of allowing databases to be connected to views and templates which are displayed to the user. Moreover, Django works hand-in-hand with frontend frameworks like React.js, which is beneficial for the frontend framework approach.

Data Manipulation and Calculations: The backbone for this strategy to work is to ensure that: 1. The portfolio is continuously rebalanced in order to maintain a steady 90/10 split between risk-free assets and long-dated options, and 2. The fair value of options can be calculated (i.e. using black-scholes) given the data stored within the databases so that the implied volatility of the real-options price can be calculated. In order to rebalance the portfolio continuously, it would be needed to utilize an API depending on the service provider that is holding the investment account and allowing for buying / selling. Since it is assumed that this strategy is to be implemented at a financial institution, then it can also be assumed that there is some internal server that allows for data requests made about the portfolio. Thus, utilizing get requests to view data, storing it and running the necessary calculations to buy/sell the correct amount of assets to maintain portfolio balance, and sending order data back to the internal server would be how the portfolio can be continuously rebalanced. In terms of how to perform the options pricing calculations, this can be done with Python and its various data libraries (such as NumPy and Pandas) on the existing options data within our database.

Bot Execution: In order for the program to automatically execute trades, it must have pre-defined parameters beforehand in order to ensure that it understands when to buy / sell.

Thus, based on previous discussions, there are two instances where the program requires executing trades: 1. When the portfolio needs to be rebalanced, and 2. When there is an opportunity to purchase low-IV option contracts. The first task of portfolio rebalancing is relatively straightforward. The trading program would take the portfolio data (the percentage makeup of all the assets) and determine the amount of either risk-free assets and / or option contracts it needs to buy / sell in order to reach the 90 / 10 portfolio split again. In terms of the second requirement, which is buying and selling options contracts, this task will be tied directly to the data calculations made. Based on what is collected within the database, the fair value option pricing can be determined (from previous calculations within the backend) and so when compared to the market price of a given option, a percentage increase from the fair value and market value of the option can be derived (the implied volatility) and if it so happens to be within the predetermined threshold of 5%, the bot will decide that the contract is an eligible candidate for the portfolio to hold a position in. Thus, in order for the program to execute these two tasks efficiently and effectively, it needs to get valid data and also have a mechanism in order to execute orders. In terms of the first issue, getting the valid data is relatively simple. It's already established that there is a database that will store the necessary asset data, and given that this is to be used within an institutional setting, getting data on the overall makeup of the portfolio should not be a hard task either, which most likely can be completed given an external outlet / API that allows the gathering of data from the portfolio. In terms of executing orders, from the same outlet / API that retrieves info, there should also be a way to send specific orders so that trades can be executed. This is the mechanism that third-party APIs, such as Alpaca API and Interactive Brokers, tend to operate.

Conclusion

In conclusion, the barbell strategy and the described practical software implementation is utilized to generate returns from the equity markets based on large price movements which have impacts that are not priced within lognormal distribution functions such as those found in the black-scholes model. Thus, this system benefits from unforeseen volatility, while also protecting the overall portfolio from large downside risk that is derived from general market risk.

This strategy can be easily implemented within an institutional setting, as highlighted by the software implementation section. Moreover, the strategy is scalable as multiple traders can work on quality control and ensuring that the options positions taken are favourable. Finally, along with this strategy, the mentality of being patient with the markets is a necessity. It is impossible to predict when the next big market move will be, much like many of the big stock market crashes in the past, but when the time comes, the barbell strategy profits handsomely.

References

[1]
Mtetwa, D. (2020, Dec 31). *Missing just a few of the best stock market days could cost you big*. The Motley Fool.
https://www.fool.com/investing/2020/12/31/missing-just-a-few-of-the-best-stock-market-days-c/
[2] Andrew Martin. (2018, Feb 1). *Stock price database* [Photograph]. findb_price_schema.png

[3] Chavez-dreyfuss, G. (2021, July 13). *U.S. fed funds, eurodollar futures raise rate hike bets after CPI data.* Reuters.
https://www.reuters.com/business/finance/us-fed-funds-futures-up-rate-hike-bets-after-cpi-data-2021-07-13/

[4] MacroTrends. (n.d). *10 year treasury rate - 54 year historical chart.* MacroTrends.
https://www.macrotrends.net/2016/10-year-treasury-bond-rate-yield-chart

[5] TradeOptionsWithMe. (n.d). *Black scholes model* [Photograph].
https://tradeoptionswithme.com/black-scholes-explained/

[6] Taleb, Nassim. (2010). *The Black Swan.* Random House Trade Paperbacks.

Useful links:
https://seekingalpha.com/article/104974-nassim-taleb-renegade-trader-with-renegade-ideas-that-work

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.695.4305&rep=rep1&type=pdf