Machine Learning Statistical Arbitrage Trading Strategy Analysis

Introduction

Statistical arbitrage has become one of the most commonly used strategies in modern algorithmic trading and involves the creation of a zero-sum portfolio of two correlated assets . The DTB 6 model (Derivative Trading Bot 6) was meant to study the effectiveness of advanced modeling in pairs trading targeting the commodity futures market. This method has been extremely profitable in the past yet has seen little advances in the underlying technology.

The basic algorithmic framework for the DTB 6 model reverses that of the modern era, attempting to force a stationary relationship on two assets' residuals and utilize their relationship in a trade. The residuals of the data were taken and 8 combinations of log functions and regression types for a pair of assets using machine learning were applied. From there, thousands of models of time series analysis methodologies were applied in altering combinations and orders, such as autoregressive models, generalized auto regressive heteroskedasticity models, and deterministic trend-fitted models. Each of these models were then netted out by the true residual values and tested for stationary based on past daily data and the best was selected.

The entry point of a trade occurs when the model's residual values exceed 1.7 times the standard deviation of themselves. When this mark was hit, a pairs trade was initialized. If one side of a trade had a 1% loss or more, the asset was sold for a stop loss. If this value is not hit, the trade closes when the residual value converges to within 0.4 times the standard deviation of themselves.

A study was then performed utilizing web-based software Quantopian as well as Jupyter Notebook. Results from each model's backtest were taken into account. Due to the severe limitations of Quantopian's backtesting platform, a full analysis was unable to be made. However, as will be discussed in the results section, the algorithm netted positive results.

Data Description

Data for this algorithm was pulled from continuous futures data ranging from the years 1990 to 2010. The #2 month was used, meaning the contract that is followed would be the contract that will expire second-closest to the given time. Data was pulled from *Kaggle*, a free online data-sourcing site. Below is a list of all assets analyzed and their respective sectors.

Metals
Gold
Copper
Silver
Palladium
Platinum
Energy
Natural Gas
RBOB Gasoline
Ethanol
Crude Oil
Meats
Live Cattle
Feeder Cattle
Pork Bellies
Agriculture
Wheat
Rough Rice
Corn

Soybeans
Soybean Meal/Oil

Model Creation

These data were imported into Jupyter Notebook using python library pandas. Data was first traversed into individual arrays and testing for integration in the order of one. If this returned positive results, the model was halted and no results were made. From there, 4 Ordinary Least Squares Regression models and 4 Weighted Least Squares Regression models were created. The dictionary ‘models’ was initialized to contain each model and each of the regression outputs and their residuals were stored under their respective names, ‘OLS1’ for example. The log combinations are listed as follows: 1 is (y,x); 2 is (logy,x); 3 is (y,logx); 4 is (logy,logx). Each model was then duplicated and altered by differencing the residual values by a significant lag. The key add-on ‘differenceN’ was added, where N represents a significant lag. More models were then applied for another set of differencing, following a similar creation structure. Models were fitted for deterministic trend through the use of an additional OLS model of the residuals. The residual values were differences by those returned by the model and added again to the dictionary ‘models’. Autoregression models were then formed for every value in dictionary ‘models’. Autoregression lag lengths were made for every single significant lag and results were appended to the dictionary. This process was then repeated for only the first lag. Each of these models (likely around 1000) were then given ARCH. Similar to the others, the residuals were given the treatment and appended to dictionary ‘models’, This then followed the use of GARCH modeling, once again following the same pattern. The residual values of each of these were then differenced by the true residual values, making somewhat of an “error of the errors” array. Each of these models and their residual arrays were tested for stationarity and profits were analyzed.

Backtesting Results

The best model from each pair was then put into Quantopian's backtesting tool. Models were tested over the range 2011 to 2016, or until Quantopian still tracked data, and results were recorded. Quantopian's ordering functions were not used due to inconsistencies and incorrect data values. Thus, leverage was constant at one and slippage is not fully accounted for (discussed later). The quantopian files were combined for various groupings (metals, meats/energy, and grains). However, Quantopian had no backtesting data on Palladium or nearly all of the grains. The data that was available was back tested on and yielded promising results.

Despite only having data for four metals (Gold, Silver, Copper, and Platinum), the algorithm returned positive results. Over the 5 year period, the model netted a return value of 45%. Although this value does not meet the benchmark (S&P 500), the ability of margin to be used for futures contracts will likely exponentiate these returns. The model saw the most growth in 2011, when it returned 24% year to year. The model returned negative values for just one year (2016).

Due to the lack of futures contract data available on Quantopian, meats and energy models were combined into one. There were only 5 available models for use for this grouping due to the lack of data. Despite this, the period of 2011-May, 2016 was used with the data Quantopian had available returned 42%. Similar to that of the Metals group, this group saw a huge boom in growth over the first few year and slowly decreased.

Quantopian had nearly no data for grains available. Only two models were found to be profitable from Quantopains data. These individually were extremely profitable, returning 48%

and 42% from 2011 to 2016. When supplemented by models in the same grouping, it is likely that they will have positive growth over the long term.

Discussion

When I wanted to analyze the ability to use this alternative strategy in statistical arbitrage, I did not expect the success that I saw. However, a few things must be discussed in order to fully evaluate the results.

Quantopian's platform is overall very inconsistent and thus did not provide strong evidence of a backtest. I am looking to run further backtests in the future through more sophisticated programs. Quantopian's lack of data had an enormous effect on the ability of the model to show returns. Quantopian provided data for about 40% of the models that were created. The data were either cut short or completely absent entirely. In addition, the failure of Quantopian's ordering methods forced me to create alternative strategies for the model creation. The graphed 'prof' represents the prof (in percent) that was made, not accounted for slippage.

The results showed extremely positive results in the early years, namely until 2013. However, the model saw a slowed growth in more recent years. This is likely due to the rise of more sophisticated statistical arbitrage models that even out the spread. Thus, it can likely be deduced that these pairs trading strategies are becoming less and less profitable and must be advanced.

The overall returns, despite everything, were surprisingly strong. I believe further research must be done on this strategy and the backtests it has. Each file represents a piece to the large puzzle and can thus be replicated in any backtesting platform.