Financial Data Science and Computing(15.458), Project F
Pairs trading strategy using fundamentals

Michele Marinucci, MIT Master of Finance 2021

Abstract

In this project, I attempt to implement a pairs trading strategy using fundamentals measures. Utilizing stock data in the period 2000-2019 across all stocks in the SP 500, I test whether PE ratio, leverage, and earnings quality ratio can be harnessed to pair stocks and implement a self-financing trading strategy. In particular, given stocks with similar PE ratios and leverage ratios, the algorithm I implemented computes the divergence in earnings quality and, if over a certain threshold, it appropriately longs and shorts the stocks in the pair. This strategy seem to perform well over the period it was first tested on, but performs quite poorly when backtested using other periods.

1. Introduction

The hedge fund industry is notorious for seeking ever more innovative strategies to beat market returns, utilizing a wealth of approaches that may range from value investing to systematic trading and beyond. While such hedge funds have traditionally chosen to focus more on either the quantitative or the fundamental side of investing, in this paper I attempt to evaluate the merits of a strategy that uses concepts from both worlds, combining pairs trading to value investing.

1.1. Theoretical background

Pairs trading is a relative value, market neutral investment strategy that algorithmically identifies pairs of securities whose prices should respect a certain relationship. When one of such pairs does not respect the relationship, the strategy longs and shorts the two securities in expectation that the relationship will reestablish itself. On the other hand, value investing is more long-term and looks at certain ratios of accounting or market measures to identify and buy undervalued securities.

The proposed strategy attempts to combine the two ideas. Initially, the algorithm pairs stocks by analyzing companies with similar leverage ratios and PE ratios. When these two ratios are similar, the implication is that the market may be valuing the companies similarly. Hence, one would expect that the companies' financial performance would also be comparable, with financial performance being measured in this case by the earnings quality ratio. The formulas for such ratios are the following:

$$Leverage = \frac{LongTermDebt}{TotalAssets}$$

$$PEratio = \frac{Price}{EarningsPerShare}$$

$$EarningsQuality = \frac{CashFlowfromOperations}{NetIncome}$$

Whenever the two earnings quality ratios differ significantly, the strategy buys the stock with the higher one and shorts the stock with the lower one. The rationale is that the market will eventually recognize the difference between the two companies, and the stock with higher earnings quality will start being priced at higher values than than the stock with lower earnings quality.

1.2. Practical considerations on implementation

The objective of this paper is to gauge the performance of this self-financing, fundamental-based, pairs trading strategy. Hence, performance is evaluated based on the cumulative profit or loss obtained during the period under analysis, computed by cumulating daily log-returns. For simplicity, the universe of securities that the strategy considers is the set of all 504 stocks currently comprised in the SP 500. The period under consideration is initially between 2000 and 2010, whereas strategy back-testing focuses on the 2010-2018 period. Notice that only stocks are considered and that the only possible actions that can be performed are going long or going short on the two securities comprising the pair under analysis.

2. Data

2.0.1. Data sources

I obtained my data though WRDS. Besides its accessibility and ease of use, the reason for which I chose WRDS is that not only did I need stock price data, but also fundamentals data for each company, which several databases do not have. Overall, I used both Compustat and CRSP, both available within WRDS.

As far as Compustat is concerned, I used it for three main data sets. First, I retrieved the list of companies within the S&P 500 along with their corresponding GVKEY, which is Compustat's unique company identifier.

Second, given that Compustat and CRSP use different permanent company identifiers, and given that plain tickers could not be used due to the high likelihood of a ticker change over the 18 year period, I use Compustat to get a table that links Compustat's GVKEY to CRSP's PERMNO. Unfortunately, such links were often ambiguous or missing, therefore I decided to drop those stocks in such cases. The final data set hence comprises only 301 stocks in total. This table is named *link_table*.

Third and most importantly, I used Compustat to retrieve the fundamental information needed to compute the ratios for each stock, including total assets, long term debt, dilute earnings per share excluding extraordinary items, cash flow from operations and net income. All such items have a quarterly cadence. This table is named *fundamentals*.

As far as CRSP is instead concerned, I only used it for one data set, this being the daily stock price time series for the stock considered from 01/01/2000 to 01/01/2019. In particular, I pulled both closing ask and closing bid prices. To simplify, I used SQL code to create a column with the average of the two, called price. Notice that this set of stocks did not use all stocks in the S&P 500, but only the one for which a PERMNO-GVKEY link was available in the link table from Compustat. The total number was thus 301 stocks. This table is named *prices*.

2.0.2. Data modeling

Once I obtained the link table, the fundamentals table and the prices table , I used Access and SQL to combine them. The relationships among the three data tables is shown in Figure 1. However, notice that, due to Access's limitations in database sizes and due to the ambiguous link between the three data sets, the illustrated query could not be run as it is. Instead, I had to run two separate queries.

Through the first query, I inner joined the fundamentals table with the link table to confirm that every permanent identifier effectively had a Compustat-CRSP link, as it was the case. Additionally, I used SQL code to directly compute the needed ratios and drop the now unnecessary fundamental measures to decrease data set size.

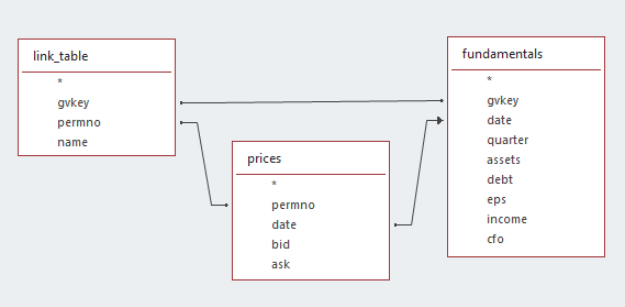

Figure 1. Figure illustrating the relationships among the data sets used.

Trough the second query, I left joined the prices table and the table coming from the first query, which let me retain all date and all daily stock prices but produced numerous missing values in the columns corresponding to the fundamentals table, whose data was quarterly. Finally, I sorted the table by date and PERMNO.

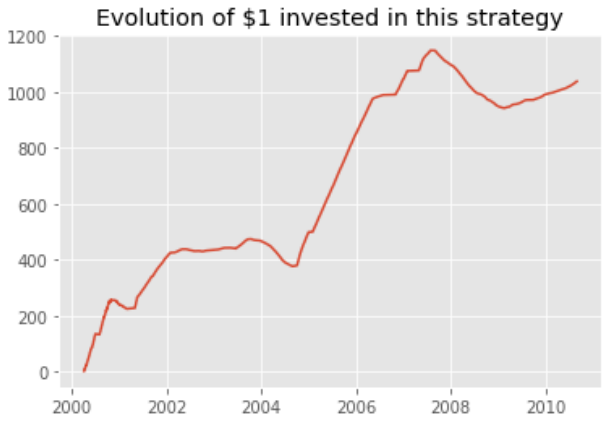

Figure 2. First attempt was quite promising, yielding very positive returns over time.

2.0.3. Missing data and feature engineering

As regards feature engineering, the only missing feature at this point is daily log returns, which I computed through the standard formula and added to the data. Furthermore, I standardized the three fundamental ratios to simplify the algorithm.

As far as missing data goes, prices in some days for some stocks were not available, thus I forward filled them. The remaining missing data in the fundamentals columns corre-

sponding to dates in which there were disclosure of quarter metrics.

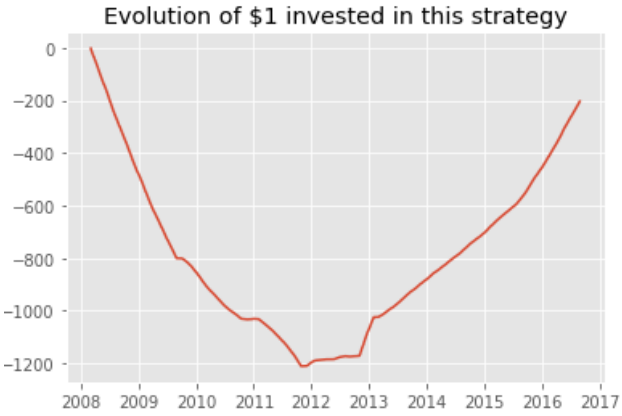

Figure 3. Back testing did not yield the best results. This tested for different period.

3. Methods

It is now time to implement the trading strategy on the data at hand. The algorithm I used is summarized above under Algorithm 1. Before moving onto the algorithm, notice that I decided to only hold one pair at the time over the period, hence I will only evaluate potential new pairs once the previous pair has been closed. Notice that the algorithm has two hyper-parameters, i.e. top closest k and divergence threshold q, both of which are explained below.

The algorithm has five steps. First, considering the first quarter of first year, it finds all possible pairs of stocks. Second, for each combination the strategy takes the absolute difference in PE ratio and the absolute difference in leverage ratio. Since they are standardized, it is fine to sum the two to get a simple and unique summary measure, which I called *distance*. Third, the algorithm ranks such distances. Forth, considering top k combinations with the lowest distance, where k is the top closest hyper-parameter, the algorithm finds the pairs with lowest distance among them and checks for divergence, or difference, in earnings quality ratios. If the pair with max divergence is over q, where q is the divergence threshold hyper-parameter, the algorithm goes long on stock with high earnings quality stock and goes short stock with low earnings quality. If instead none of the top k closest pairs have a divergence over threshold q, the algorithm waits until next quarter and tries again. Fifth and lastly, the algorithm closes the position if the divergence in earnings quality is no longer over the divergence threshold. If the position is closed, start again with first, using current quarter and current year this time.

Algorithm 1 Pairs trading

Input: top closest(k), divergence threshold(q), start date, end date, data
Initialize positions held.
while $currentDate < endDate$ **do**
  find all possible stock pairs
  find top k closest pairs in terms of fundamentals and their corresponding divergence in earnings quality
  among these, find the one with highest divergence
  if $divergence > thresholdq$ **then**
    long the stock with higher earnings quality and short the other
    while $stillOpen = True$ **do**
      move onto next date
      if $divergence < thresholdq$ **then**
        close position
        $stillOpen = False$
      end if
    end while
  end if
end while

Table 1. Cumulative returns over different windows and with different hyperparameters

	DIVERG. Q	TOP K	PERIOD	CUMRET
1ST	0.05	5	2000-2010	$1037
2ND	0.05	5	2008-2017	$(203)
3RD	0.03	3	2000-2010	$(29)

4. Results

Please refer to Table 1 for a summary of the how the strategy performed across three attempts. Each attempt used different time periods and different hyperparameters. These three different attempts were performed as a way of back-testing the strategy.

Overall, these results did not satisfy my expectations, as the performance was quite poor and close to random if the whole 2000-2018 period is considered. More specifically, looking at Table 1, it seems that this strategy worked a lot better before the 2008 crisis, whereas it reverses and turns negative afterwards. Furthermore, Figure 2 and Figure 3 further show how horizon dependent this strategy may be, with very high profits in Figure 2 and with very high loss (mostly recouped afterwards) in Figure 3.

In hindsight, such results were to be expected. This strategy may overly rely on the earnings quality ratio, disregarding several other systemic and idiosyncratic factors that may affect returns. Hence, any future work that build on this rational should beware and add such factors to their algorithms.