Statistical Arbitrage

1 What is Statistical Arbitrage?

From Wikipedia: *In finance, statistical arbitrage (often abbreviated as Stat Arb or StatArb) is a class of short-term financial trading strategies that employ mean reversion models involving broadly diversified portfolios of securities (hundreds to thousands) held for short periods of time (generally seconds to days).*

2 Modeling Returns using Linear Regression

2.1 Motivation

We're interested in linear regression because it's a simple model for a response variable (return of a stock) and explanatory variables (systematic and idiosyncratic factors – explained below).

2.2 Decomposition of Returns

A central idea in stat arb is that you can decompose the returns of a stock into different factors:

2.2.1 Systematic Factors

External things that affect the return of a stock like war, a slump in the firm's industry, general market conditions, or anything outside of the stock itself that is causing its valuation to move one way or another.

2.2.2 Idiosyncratic Factor

The return you're left with when you remove systematic factors. You can call this the *specific return* – and probably should, because *idiosyncratic* is peak academic beard-stroking and isn't actually meaningful.

These things trigger me, sorry :(

Put another way, you're normalizing the return of the stock by removing all external influences. Put **another** way: think of an internet company during the Dot Com boom that IPO'd with zero revenue but shot up 5000% in a year. How much of that massive return was due to the stock simply riding the wave of Dot Com madness (a systematic/external factor) and how much was due to factors specific to the stock itself? The latter is the *idiosyncratic return* we're searching for – and for this hypothetical Dot Com stock, is probably close to 0.

2.3 Examples: Decomposition of Returns

AAPL
Apple roughly doubled in 2019. Let's denote this return as $T = 1$, meaning \$1 invested on Dec 31, 2018 would yield $\$1 + T = \$2$ dollars on Dec 31, 2019.

Let $T = (c_1 * R_1 + c_2 * R_2 + ... + c_n * R_n) + I$ where $\{R_i\}$ is the set of systematic factors and I is the idiosyncratic return. Some systematic factors could be performance of the tech industry or performance of the entire US market. Imagine we've selected *enough* systematic factors (idea explored more in PCA section). Now we need to know how sensitive AAPL was to each factor – this is where the c_i's come in. For example, we may think that AAPL was especially sensitive to the tech industry's general performance, so we weigh that factor by a lot – say, a coefficient of .9. But we think another factor – say, the performance of gold – was irrelevant, so we weight it by a coefficient of .02. Imagine that we did this analysis for each factor and the sum of the weighed systematic factors was .7. This means that .7 of the return of 1 was due to external/systematic factors and .3 was due to idiosyncratic factors.

SPY
SPY is a tracker of the US market, so its returns will essentially be 100% explained by systematic returns (i.e. it *is* the system).

GDX
GDX is a gold ETF. As such, it's value is relatively uncorrelated with the broader market and other common systematic factors (let's ignore the whole *gold as a hedge* idea for now). So, if we decomposed its returns we would see almost all of the return explained by the idiosyncratic component.

3 Selection of Systematic Factors

To re-cap:

1. Stat Arb is interested in finding the idiosyncratic returns of an asset (i.e. stock).
2. We define the idiosyncratic return to be the stock's return after removing the systematic return (the sum of systematic factors weighed by how sensitive the stock is to that factor).

So, our real work lies in identifying systematic factors. How can we do this? Well, there are lots of ways – most of them super secret because they're the gold mines of quant shops. But a classic method is using Principal Component Analysis. I'm going to assume that you know PCA (if you don't, a good primer is in the *papers* directory of this repository). Then, you should know that PCA can be interpreted as finding the (dimension-reduced) basis that best represents the data. When applied to a dataset of returns, we can interpret this as PCA finding a set of independent vectors (*factors*) that best represents the returns. Voila! You've identified your systematic factors – and the beautiful thing is:

1. The factors are guaranteed to be independent, since the set of Principal Components is orthogonal.

2. We introduced minimal subjectivity in the factor selection process, since we used an unsupervised process vs picking our factors by hand.

3. Sensitivity is given to us for free: simply denote the sensitivity of the stock to a systematic factor s_k as the ratio of eigenvalue λ_k to the sum of all eigenvalues (i.e. how much this components variance contributed to total variance).

3.1 How Many Factors Should We Choose?

There may be dozens and dozens of systematic factors for a stock, while only a sliver are needed for our algorithm. The idea is to choose the minimum number of factors to explain the majority of the overall systematic return. If using PCA, we can express this mathematically: We want to choose the first k components which cumulatively explain around 51% of the variance in the data.

4 Scoring

To re-cap:

1. Stat Arb is interested in finding the idiosyncratic returns of an asset (i.e. stock), so that we can tell if a stock is *cheap* or *expensive*.

2. We define the idiosyncratic return to be the stock's return after removing systematic returns (the sum of systematic factors weighed by how sensitive the stock is to that factor).

3. We can find systematic returns via PCA.

Let's jump back to the whole point of this: we want a computer to be able to trade stocks. To do this, a computer must be able to assess whether a stock is cheap or expensive. **We do this by scoring a stock** – computing a metric that numerically tells us whether its cheap or not. To do this, we make a strong assumption: the fluctuation of a stock's idiosyncratic return is a mean-reverting process. With this assumption, a computer can easily calculate a numerical measure of relative value: Is the idiosyncratic return especially high at the current moment, relative to its average? Then assign a high score. Vice-versa if its low. Then, the computer will sell stocks with a high score – since we expect its value to revert to the mean (i.e. fall) and buy stocks with a low score for the opposite reason.

5 Simple Stat Arb Algorithm

There are lots of ways to implement the strategy above. My first algorithm did it by:

1. Calculating systematic returns via PCA
2. Calculating idiosyncratic returns
3. Scoring a universe of stocks based on the relative size of their idiosyncratic return
4. Creating a portfolio of n stocks by selling the $n/2$ stocks with the highest scores and buying the $n/2$ stocks with the lowest scores
5. Balancing the portfolio using some risk optimizer
6. Sending out orders to construct the portfolio
7. Re-balancing portfolio as needed and re-constructing portfolio on some regular basis

6 Conclusion

A common phrase you'll hear is *buy low, sell high* or analogously *sell high, buy low*. With stat arb, we've can create a model that allows us to numerically assess whether a stock is *low* or *high*. In this form, trading can be conducted by a computer.

In a nutshell: Statistical arbitrage is simply the mathematical translation of the heuristics that investors have used since the dawn of time, so that a computer can understand and carry out that logic.