Overfitting in Quantitative Trading

October 2016

www.qplum.co

1620 Plaza 5 Jersey City, NJ 07311

(201) 377-2302 contact@qplum.co (201) 604-5422

Abstract

The hardest part of data-science is to prevent overfitting. Overfitting occurs when a mathematical model explains the training data almost too perfectly, to the extent that it does not work on untrained data.

In this report we will show a few ways in which overfitting can creep in and how to solve for it.

Data snooping

Problem: *Data-science traders might be using the same data to build the model and to test profitability.*

Suppose I am trying to use linear regression for trading. So I need to make a linear model that predicts the returns of the stock I am trading. Based on the prediction, I need to place orders. The only question I am trying to answer right now is how profitable is this strategy?

I could setup an experiment to learn the parameters of this model using all twenty two years of data that I have and then evaluate how the past returns based indicators would have predicted future 1 day return within that period.

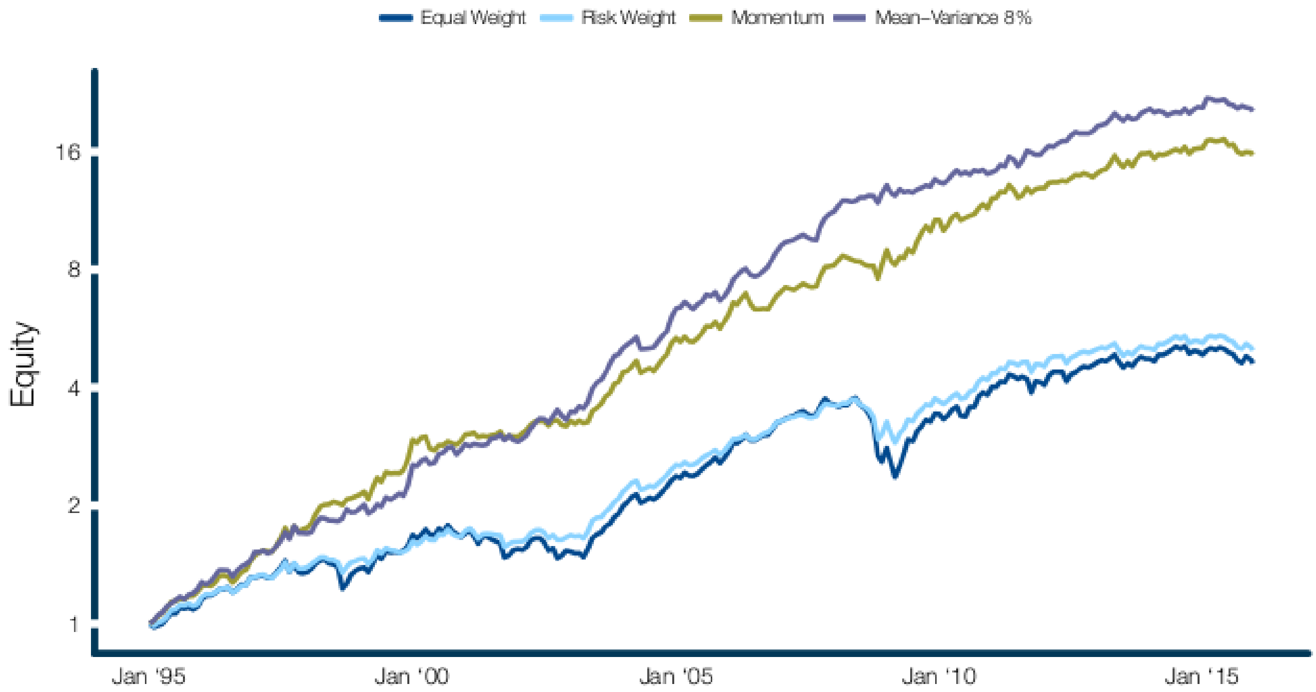
I present the back-tested PnL of a strategy that would have used the output of this regression model to calculate the allocation for each trading day.

The problem is data-snooping. In this experiment I am using the same twenty two years of data to make my model and to test its PNL. I am implicitly assuming that the PNL will be similar in future to what I see in my backtest. What is wrong is that if I had really been trading with this strategy in these twenty years, I would not have this future data to make my regression model to begin with! While creating the model, I did not go back in time with the model. The model fine tuned its parameters by looking at the data ahead in time. In short, my training data set and testing data are the same.

This problem becomes more acute when evaluating systematic strategies like momentum or risk-parity based on backtested performance. The asset managers have the benefit of looking at all the data to cherry pick the strategies that look best on the backtested data.

For instance, an asset manager showed the performance of a strategy on the years of backtest that they have (Exhibit 4 on the next page). It is clear that the strategy was chosen by the asset manager because it's performance on the twenty years of data is great. Fitting it so well to the past may take us to bet on aspects about the past which are very less likely to occur in future. The results of backtest then may literally have no bearing on how the strategy will perform in future.

Exhibit 4: 10 Assets, Top Half by 1, 3, 6, 9 and 12 Month Momentum, Mean-Variance Target 8% Volatility, Rebalanced Monthly (1995-2015).



Does this mean we should always discount back-tests?

Solution : *Walk-Forward optimization. Future-Testing and not back-testing*

The solution is what we are taught in the Artificial Intelligence 101 course. Treat past data as something you did not have access to before you made the strategy. For instance, suppose we are choosing between three strategies. Today we can see which of them would have done the best over the last twenty years. However fifteen years ago, we could have only seen what would have done the best in the preceding five years. In computer science terms, this is called **walk forward optimization**. Every month, we would be looking at the data that we have till then, making a model on that data and then using it for the following month.

Use a simulation model of possible future paths. This way we can never overfit to parts of past data that are irrelevant today. We have expanded on this in the last section of this paper.

Too many parameters

Problem*: Introduce lots of parameters, fine tune them and overfit!*

The general process of quantitative research is to try and see what works. Usually almost nothing works the first time. Then models are fine-tuned and more conditions are added on when to use the specific models. Too much parametrization could lead to overfitting too.

Solution*: Discount results of strategies with many parameters*

Again, we can take a leaf out of statistics here to discount results of strategies that have too many parameters. In linear regression one measure of performance is adjusted RSquared, which basically means that instead of using the raw measure of prediction accuracy, RSquared, I am going to give less points to the model if it has been using too many indicators. So while adding more indicators always increases the RSquared, adjusted RSquared falls off quickly after the first few indicators have been added.

Similarly in data-science trading, we need to assess and define a rough measure of how many parameters are in a model, and how likely is it to hold in out of sample. We need to be cognizant of the trade-off between discovering new information vs overfitting whenever another parameter in added to a model.

From back-testing to future-testing

Three truths of data-science:

1. Past performance is not what we care about. We care about future performance.
2. Past data is all we have to learn from. We don't have future data.
3. Not all aspects of past data are equally likely to occur in future.

In accordance with Truth-1, at qplum, we are trying to change our way of comparing strategies from past-performance to future performance. To do that we want to simulate different possible future scenarios and see what is likely to work best in future.

Let's say we are comparing strategies based on the projected performance over the next one year.

One approach doing so would be to do biased sampling of the past data to simulate the next day. We should only consider data points from those days which were close to the current environment. How we define closeness will make a huge difference here. Let’s assume for now that we choose the days that are closest in terms of interest rates and stock market volatility. That means our simulator could choose any of the days in the historical data set that had similar interest rates and similar stock market volatility. Based on this expectation, we can then compute the expected volatility and expected returns of each core ETF. Now we can estimate the rest of the values based on the historical correlation. In this example we are simulating one time step at a time, let's say one day at a time. Based on the path that each simulation run will take, we will have another starting set of values at the end of this day. We then find the points in history closest to this new set of values, to simulate the day after that. This way we can simulate the performance of each strategy based on past data and yet we are simulating the future and not the past.

We believe that this way of analysis will be a major departure from the way research on identifying strategies has been done till now. It will help portfolio managers a lot in choosing strategies that are the best fit for the future.

References:

[1] Quantifying Backtest Overfitting in Alternative Beta Strategies by Antti Suhonen, Matthias Lennkh, Fabrice Perez

[2] Telling the Good from the Bad and the Ugly: How to Evaluate Backtested Investment Strategies by Patrick Beaudan

[3] Statistical Overfitting and Backtest performance

[4] What to Look for in a Backtest by Marcos Lopez de Prado