The Perils of Unfettered In-Sample Backtesting

Tyler Yeats

June 8, 2015

Abstract

When testing a financial investment strategy, it is common to use what is known as a *backtest*, or a simulation of how well the strategy would have performed under some past data. However, it is very easy to overfit a given strategy to the training dataset, while not actually preparing it for unknown data. If a model has enough parameters that can be changed, a researcher can tweak them to create a vast number of potential strategies. Testing more strategies, however, increases the likelihood that the model is being fit to idiosyncrasies of the particular dataset, and not revealing an underlying pattern. Because of this, the number of different configurations tested must be accounted for before declaring a particular strategy to be a success. [1]

Contents

1 Introduction

When evaluating a potential investment strategy, analysts often perform *back-tests*, which are simulations of how well the strategy would have done in some past dataset. The performance can then be gauged against various different metrics, such as the *Sharpe ratio*, which is a measure of how well the strategy returns on its risk. Strategies with a higher Sharpe ratio are more desirable and are assumed to perform better.

There are two different types of testing that can happen. First is *in-sample* (IS), where evaluation happens against the dataset that was referenced in the creation of the algorithm. The other type of testing is *out-of-sample* (OOS), where the algorithm is evaluated against new data. In determining how sane a particularly strategy is, however, one must consider the possibility of overfitting. Overfitting occurs when a certain strategy is tailored to a particular dataset. This can happen when the dimension of the algorithm in question has a much higher dimension than the data. In the same way that it is absurd to fit a 10^{th} degree polynomial to linear data, adjusting parameters of an algorithm so they fit a certain dataset creates bad models.

Unfortunately, not everyone uses the most rigorous techniques for analysis. Computers make it possible to search through millions of different possible combinations of parameter values, looking for one that happens to perform well for the given data. Compounding this is the fact that often the entire current dataset is used to formulate the algorithm, leaving no data for OOS testing. In fact, the state is so bad that some, such as Harvey et al. [2], even go so far as to say that the majority of financial models are based on false priors.

1.1 Intuition

Here we will work to briefly develop the intuition behind the concept of overfitting. We will not define the Sharpe ratio yet, but it is enough to know that a higher ratio is desirable for an investment strategy.

Assume that an analyst wishes to find a strategy with a high Sharpe ratio. Once he has identified one, he can use the AIC (Akaike information criterion) statistic to determine the likelihood of overfitting [1]. However, even if the threshold is 99%, then for testing 100 strategies he will find on average one that passes the threshold, even if none of them actually identify some underlying pattern. For example, if a given model has 10 different binary parameters, then it has 1024 different possible configurations, of which about 10 will pass the 99% threshold. Of course, it is possible to construct much more complex models and test millions of different configurations to find one with a high Sharpe ratio.

When presenting findings, analysts usually do not mention the other trials conducted, and so it may appear that they have found suitable candidates.

2 Overfitting

2.1 Definitions

We will start with a few definitions that will appear throughout the paper.

Backtest 2.1.1. A backtest is a test where an investment strategy is simulated over a past dataset to see how it would have performed. There are two types of backtest: in-sample (IS) and out-of-sample (OOS). In-sample is where the algorithm is evaluated over the data that was used to devise it. Out-of-sample is where the algorithm is evaluated against data that was not used in its creation. [1]

Overfitting 2.1.2. Overfitting occurs when a model tries to model data too accurately, to the point where it is also modeling any noise in the data. It can also be thought of when IS returns outperform OOS returns. [1]

Prior 2.1.3. A belief that a certain pattern is present in the data. If the prior is true, then an algorithm that takes advantage of it should produce better than random results. If the prior is false, then such an algorithm should not do better than random. However, due to overfitting, this may appear not to be the case IS. [1]

Sharpe Ratio 2.1.4. Abbreviated SR in equations. Take R_S to be the return of strategy S, and R_b to be the returns from a "risk free" strategy (e.g. a Treasury bond, in the case of finance), on T different measurements, with q measurements taken per year. Assume that $R_t = R_S - R_b$ has a normal distribution, with mean μ and standard deviation σ. The Sharpe ratio is then defined as

$$SR = \frac{\mu}{\sigma}\sqrt{q} \tag{1}$$

This is the ex post ratio, which looks at historical data. There is also the ex ante ratio, which instead takes μ to be the expected value of S, and σ to be the predicted standard deviation. [4]

2.2 How to not Guarantee Overfitting

We will demonstrate just how easy it is to overfit a model by trying enough different configurations, and give a way to make sure you do not *guarantee* you are overfitting. From Lo [3], we have the result that as the number of years y of the sample size goes to infinity, the estimated Sharpe ratio converges to a normal distribution:

$$\mathcal{N}\left[SR, \frac{1 + \frac{SR^2}{2q}}{y}\right] \tag{2}$$

That is, a mean of the true ratio, and standard deviation of $\frac{1+\frac{SR^2}{2q}}{y}$. As we can see, the smaller the sample time, the more we run the risk of getting a value far off of the true ratio.

As an example, let us consider algorithms that have a Sharpe ratio of 0 over the time period of a single year. The normal distribution then becomes $\mathcal{N}(0,1)$. If we take a large enough number of samples, we would expect that we could, by chance, get a number of any size. In other words, the expected value of the maximum of a set of samples goes to infinity as the number of samples goes to infinity. Bailey et al. [1] give an approximation of the expected maximum after N trials to be

$$E[max_N] \approx (1-\gamma)Z^{-1}\left[1-\frac{1}{N}\right] + \gamma Z^{-1}\left[1-\frac{1}{N}e^{-1}\right] \tag{3}$$

where Z is the cumulative distribution function (CDF) of $\mathcal{N}(0,1)$[1] and γ is the Euler-Mascheroni constant. An upper bound for this is

$$\sqrt{2\ln(N)} \tag{4}$$

for large N [1]. Figure 1 shows a graph of this expected value, along with a sample of actual values from a simulation. It does not take very many trials to find a rather high Sharpe ratio, even though all the strategies in question should have a ratio of 0.

We have assumed until now that the period we are looking at is one year. If instead we look at the same number T samples over y years, then from Equation 1 we can multiply by $\frac{1}{\sqrt{y}}$ to find the new expected maximum Sharpe ratio:

$$E[max_N] \approx \frac{1}{\sqrt{y}}\left((1-\gamma)Z^{-1}\left[1-\frac{1}{N}\right] + \gamma Z^{-1}\left[1-\frac{1}{N}e^{-1}\right]\right) \tag{5}$$

The significance of this equation comes from the fact that we know the Sharpe ratio should be 0, and we want to bound the maximum ratio we find for large N. The most obvious way from the equation is to increase y. This makes intuitive sense, as if the algorithm does not take advantage of any underlying pattern, exposing it to a larger dataset will tend to decrease its apparent efficacy.

Let's say however that we want to find the length of time needed to ensure that we do not select an IS Sharpe ratio of $E[max_N]$ for strategies with expected OOS ratio 0. We can solve the previous equation for y, giving:

$$y \approx \frac{1}{E[max_N]^2}\left((1-\gamma)Z^{-1}\left[1-\frac{1}{N}\right] + \gamma Z^{-1}\left[1-\frac{1}{N}e^{-1}\right]\right)^2 \tag{6}$$

[1] the CDF of a probability distribution is a function of x, with the value that the probability a random value c of the probability distribution will be less than x

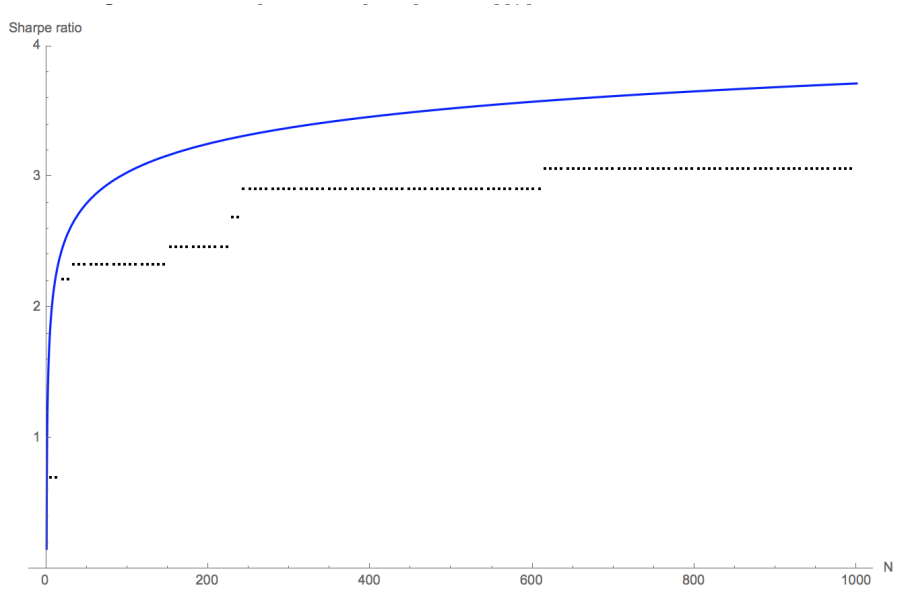

Figure 1

A graph showing the highest found Sharpe ratio as a function of the number of trials. The blue line is $\sqrt{2\ln(N)}$, and the dotted line is the actual maximum value in a simulation.

In order to prevent strategies that have an expected Sharpe ratio of 0 OOS from having IS ratios of $E[max_N]^2$, then, we must have at least this many years of data. Note that we can find an upper bound for y using Equation 4:

$$y < \frac{2\ln(N)}{E[max_N]^2} \tag{7}$$

Of course, it is always possible to be lucky (or unlucky, as the case may be) when trying configurations, and accidentally overfitting even with a small number of trials. This bound is merely the minimum that must be used so that overfitting is not virtually guaranteed.

2.3 Ease of Overfitting

To show how overfitting is easily accomplished, we will run a few simulations. To enforce the assumption that the OOS Sharpe ratio of each of the strategies we try is indeed 0, we will use random walks as our data, so there is no pattern.[2]

[2]Or at least there shouldn't be. A discussion of pseudorandomness is outside the scope of this paper.

We take a market with a single stock over a period of 1000 intervals. At each interval, we can either buy one share of stock, or sell one share.[3] Also after each interval, the price of the stock changes by some random amount. The decision whether to buy or sell is determined by creating a polynomial with random zeros on $[0, 999]$. If the value of the polynomial is positive at time t, we buy; otherwise, we sell.

After running simulations of numerous different "strategies", we run the same strategies against another random dataset. This represents the OOS test. For simplicity, we forgo the Sharpe ratio and focus on the returns of each simulation. We then plot the amount of money made in the IS test versus the OOS test. Figure 2 shows the results. As can be seen, there is no correlation between the IS performance and OOS performance. This means that how a strategy does in the IS test is no indication about how well it will do in the OOS test. Note also that the points are centered about the origin. We would expect this, as it means that the average return from the strategies is 0, and the market is a random walk. The whole point of overfitting, though, is that even with a mean of 0, there is enough variance that by looking hard enough, it is possible to find a strategy that appears to perform well IS.

2.4 Overfitting with Global Constraints

In the previous section, our IS and OOS tests were completely unrelated. When conducting the OOS test, for example, the algorithms did not have any information about the previous IS testing. They were merely selected based on how they happened to perform.

However, this is not always the case. There are many ways that the two tests may not be entirely independent, but we will focus on introducing a global constraint. In this case, we will dictate that the mean value is the same for the IS and OOS tests. To enforce a mean μ on a series of points X_t, we can take $X'_t = X_t - \overline{X}_t + \mu$, where the bar denotes the mean.

Surprisingly, when we run the simulation with this added constraint, we see that there is actually a negative correlation between IS performance and OOS performance. This is no good. Figure 3 shows the results of running the simulation.

In fact, this leads us to one of the more interesting theorems discussed in Bailey et al. [1]:

Theorem 2.1. For any two configurations J and K for a given strategy where $\sigma^J_{IS} = \sigma^K_{IS} = \sigma^J_{OOS} = \sigma^K_{OOS}$, if $\mu_J = \mu_K$, then

$$SR^J_{IS} > SR^K_{IS} \Leftrightarrow SR^J_{OOS} < SR^K_{OOS} \tag{8}$$

[3] We do not worry about having negative values of either shares or money, both of which are possible in actual markets.

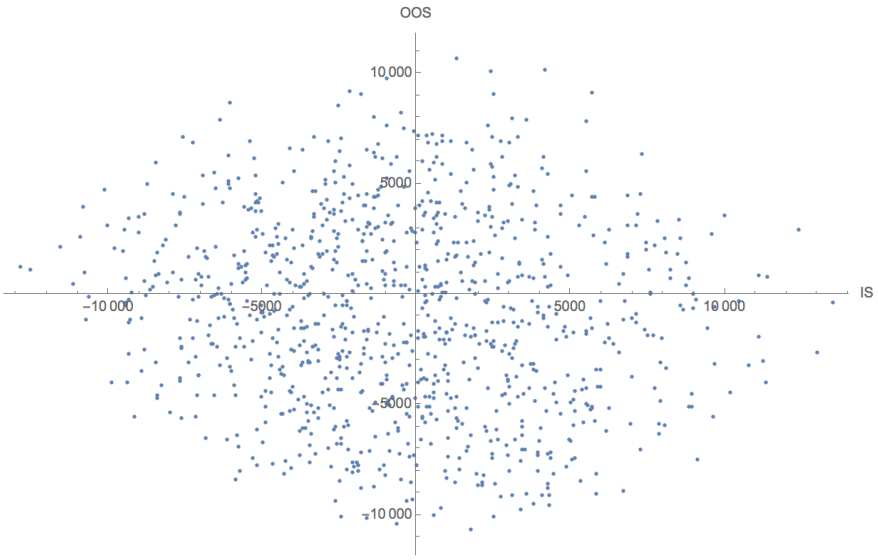

Figure 2
A plot of the OOS test versus IS test for the simulated market. The market data was random with 0 mean. As there is no correlation between IS performance and OOS performance, any strategy that performed well IS is an example of overfitting.

Proof. Let J and K be random samples of the same random walk, and call a fraction f of them be IS, the rest OOS. We assume that $\sigma^J_{IS} = \sigma^K_{IS} = \sigma^J_{OOS} = \sigma^K_{OOS}$, and also $\mu_J = \mu_K$. It follows from there that

$$\mu^J = f\mu^J_{IS} + (1-f)\mu^J_{OOS}$$

and

$$\mu^K = f\mu^K_{IS} + (1-f)\mu^K_{OOS}$$

We can then get

$$\mu^J_{IS} > \mu^J_{OOS} \Leftrightarrow \mu^J_{IS} > \mu^J \Leftrightarrow \mu^J_{OOS} < \mu^J$$

$$\mu^K_{IS} > \mu^K_{OOS} \Leftrightarrow \mu^K_{IS} > \mu^K \Leftrightarrow \mu^K_{OOS} < \mu^K$$

However, because of the fact that $\mu^J = \mu^K$,

$$\mu^J_{IS} + \frac{1-f}{f}\mu^J_{OOS} = \mu^K_{IS} + \frac{1-f}{f}\mu^K_{OOS}$$

$$\mu^J_{IS} - mu^K_{IS} = \frac{1-f}{f}(\mu^K_{OOS} - \mu^J_{OOS})$$

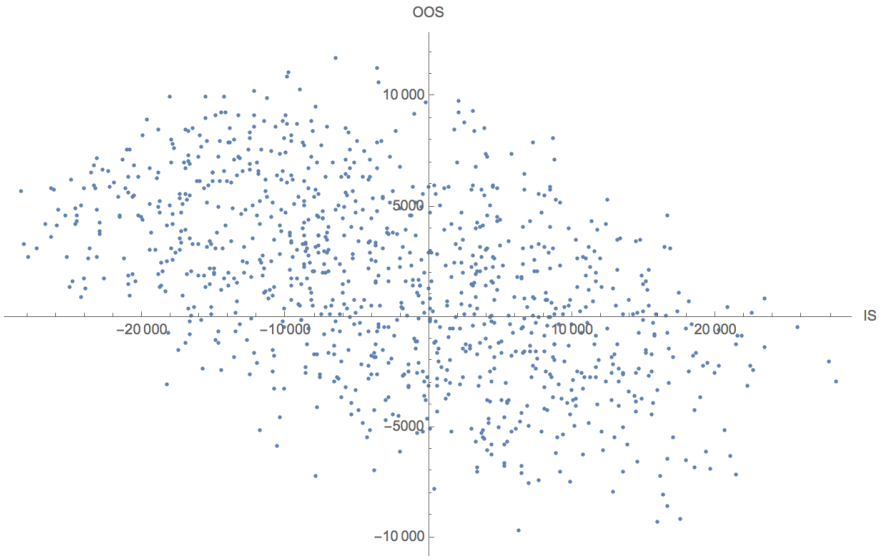

Figure 3

A plot of the OOS test versus IS test for the simulated market with the added constraint that the average value of the IS test and OOS test was the same. There is a negative correlation between performance on the IS test and OOS test, meaning that a strategy that worked well IS would fail miserably for OOS.

Thus,

$$\mu_{IS}^{J} > \mu_{IS}^{K} \Leftrightarrow \mu_{OOS}^{J} < \mu_{OOS}^{K}$$

Dividing everything by σ_{IS}^{J} to get

$$SR_{IS}^{J} > SR_{IS}^{K} \Leftrightarrow SR_{OOS}^{J} < SR_{OOS}^{K}$$

by the definition of the Sharpe ratio. [1]

□

3 Conclusion

The abuse of mathematics that is spurious analysis is evident in the process of unchecked backtesting. In the search for better efficiency, analysts (and others) are overemphasizing lucky configurations. In order to be able to better assess whether a particular model is valid or only does well because of overfitting, it is necessary to know how many trials were conducted as well as the details of the dataset and model. If this information were routinely made available, and if more out-of-sample testing was conducted, then the overall state of algorithmic predictions would be significantly better.

4 Appendix A

Code to generate Figure 1.

```
l = RandomVariate[NormalDistribution[], 1000];
Show[{Plot[Sqrt[2*Log[x]], {x, 0, 1000}, PlotRange -> {0, 4},
           PlotStyle -> {Blue}, AxesLabel -> {"N", "Sharpe ratio"}],
     ListPlot[Table[If[Mod[x, 7] == 0, Max[l[[1;;x]]]], {x, 1, 1000}],
              PlotStyle -> Directive[PointSize[.003], Black]]}]
```

Code to generate random strategies.

```
import numpy
import random
import os

def rand_walk(n):
    l = []
    for i in range(n):
        l.append(numpy.random.normal())
    l = numpy.cumsum(l)
    return l

def pos_fun(p1, p2, p3, p4, p5, p6, p7, p8, p9, x=0):
    val = (x - p1) * (x - p2) * (x - p3) * (x - p4) * (x - p5) *
    (x - p6) * (x - p7) * (x - p8) * (x - p9)
    return val > 0

stock_price = 0
sample_size = 1000
l = rand_walk(sample_size)
best = -1000000
best_vals = (-1, -1)
worst_vals = (-1, -1)

all_models = []
vals = [0]
ret = 0.

for m in range(1000):
    p1 = random.randint(0, sample_size)
    ...
    p9 = random.randint(0, sample_size)

    owned = 0
    money = 1000

    all_models += [(p1, p2, p3, p4, p5, p6, p7, p8, p9)]
```

```
    # IS simulations
    for i in range(sample_size):
        if pos_fun(p1, p2, p3, p4, p5, p6, p7, p8, p9, i):
            owned += 1
            money -= l[i]

            if i > 0:
                ret += money / (money + l[i])
                vals += [money / (money + l[i])]
        else:
            owned -= 1
            money += l[i]

            if i > 0:
                ret += money / (money - l[i])
                vals += [money / (money - l[i])]

    money += owned * l[-1] # convert all to money based off latest price
    all_models += [money]

    if money > best:
        best = money
        best_vals = (p1, p2, p3, p4, p5, p6, p7, p8, p9)

print "Best return", best
print "Best return params", best_vals

oos = rand_walk(1000)

points = []

# OOS simulations
for j in range(len(all_models)/2):
    money = 0
    owned = 0
    for i in range(sample_size):
        if pos_fun(*all_models[2*j], x=i):
            owned += 1
            money -= oos[i]
        else:
            owned -= 1
            money += oos[i]

    money += owned * oos[-1] # convert all to money based off latest
        price
    points += [(all_models[2*j+1], money)]

print points
```

References

[1] D. Bailey, J. Borwein, M. López de Prado, and Q. Zhu, Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance, *Notices of the American Mathematical Society*, 61(5), May 2014, pp.458-471. Available at ssrn.com/abstract=2308659

[2] C. Harvey, Y. Liu, and H. Zhu, . . . and the Cross-Section of Expected Returns (February 3, 2013). Available at ssrn.com/abstract=2249314

[3] A. Lo, The Statistics of Sharpe Ratios *Financial Analysts Journal*, Vol. 58, No. 4, July/August 2002. Available at SSRN: http://ssrn.com/abstract=377260

[4] W. Sharpe, The Sharpe Ratio, *The Journal of Portfolio Management*, 21(1), Fall 1994, pp.49-58. Available at http://www.iijournals.com/doi/abs/10.3905/jpm.1994.409501