Simulation as a Stock Market Timing Tool

Tony Cooper[1]

Double-Digit Numerics

This version 28 February 2014

JEL classification: C02; C15; C18; C52; C53; G17

Keywords: Backtesting; Simulation; Smoothing; Data Mining Bias; Moving Averages; Momentum

1. Introduction

The rapid increase in the processing power of computers in the past two decades and the emerging availability of powerful computers in the cloud has given rise to a new form of experimentation called *in silico* (literally *in silicon*, that is, doing the experiment entirely on a computer). Badyal et al (2009), for example, show how animal experiments (*in vivo*) are being replaced by computer models – new drugs that used to be tested on mice can now be tested in computer models of mice.

This is more than just computer modeling and simulation – there is data too. Wishart et al (2006) describe Drugbank which is a massive data repository containing drug entries, protein sequences, bioinformatics and cheminformatics entries, and links to other databases. Furthermore, Drugbank is "open" – it is free and available to anyone at drugbank.ca. Now the public can develop new drugs.

We envision similar resources – models, simulations, and data – being available for conducting financial experiments *in silico*. This paper is a nod in that direction. We show by example the power and potential of conducting financial simulations to test models, hypotheses, and investment trading ideas. For example, one question we have pondered is this: which is more important for predicting financial markets – the *size* of a phenomenon (such as the correlation of past returns with future returns) or the *statistical significance* of the phenomenon? We show how we can answer that question. We might

[1] email: tonyc@ddnum.com

wonder: is it possible to time switching between the SPDR sector funds (XLB, XLE, etc)? How much dispersion in returns do we need for timing to work? We can answer those questions too.

One of the important tools of financial research is backtesting. Backtesting refers to the process of testing investment strategies on past market data in an effort to predict the strategy performance on future data.

The problem with this approach is that there usually isn't enough past data to make statistically significant conclusions on the performance of the investment strategy in the past and that past market data does not resemble closely enough what the future market will look like.

This paper solves that problem by using simulation to generate market data for testing purposes. It is not obvious at first how simulated data can be helpful for testing future markets but we show in two examples how this can be done. Beyond these examples the possibilities are endless.

This is primarily an expository paper which explains concepts that are quite simple. So we omit formal technicalities such as bootstrap, robustness, statistical tests, and stress tests and leave out mathematics. The investing strategies used as examples are very easy to apply.

The structure of this paper is as follows: we introduce a list of problems that arise from using backtesting on market data, we then show some of the problems with market data itself (in particular problems with the S&P 500 index), then we show how market noise manifests itself as disingenuous backtesting results.

Then we introduce our simulator and describe some of its capacities. Its main features are the ability to model markets in various different ways. This includes algorithms for modeling and generating noise (including heteroskedastic noise), algorithms for generating random walks of various kinds including trending and mean-reverting walks, algorithms for adding time variation to all our parameters (such as simulation a market that switches between trending and mean-reverting phases), algorithms for incorporating the Capital Asset Pricing Model for modeling whole portfolios of stocks and Exchange Trade Funds, and algorithms for detecting and generating regime changes (such as switching between volatile and calm markets). The simulator is a work in progress and will be available in open source.

We show it in action in an example using moving average strategies. We demonstrate the effect of noise on the task of optimizing the moving average strategy. We show how, even with centuries worth of market data we still would not have enough data to optimize the trading rule due to the amount of noise in the stock market. We introduce a method for dealing with the noise – that of smoothing. Then we show using simulations the effect of the smoothing on the future performance of the investment strategy.

The simulations allow us to optimize the smoothing parameter without "data mining bias" or "overfitting" or any of the other problems with backtesting. In the following section we show using charts how the data mining bias introduced by backtesting manifests itself. We show in an extraordinary chart how there is an inverse relationship between backtested returns and future actual returns – in other words, the better the returns from the backtest, the worse the returns are in real life. This is counter to the intuition of many traders and people who do backtesting.

Then we introduce a second simulation example – that of optimizing a portfolio of investment assets. We show how simulation can help in choosing the optimal number of assets and how to time the rebalancing of those assets. This time we use momentum trading rules as they are simpler to apply when timing multiple assets. We give reasonably convincing evidence that the optimal number of assets to hold is 11 even though we do not have enough data to run accurate backtests. It is interesting to note that for doing these calculations we conducted billions of market simulations. This kind of research would not have been possible a decade ago.

Finally we end with a discussion of the drawbacks of the simulation method. The most significant problem is that of ensuring that the *in silico* environment matches the *in vivo* (real life) environment. As the modeling and simulation technology evolves along with computing power the matching will improve just as it is improving in the biological realm.

This paper talks about algorithms such as *n*-dimensional smoothing algorithms. More discussion on the algorithms and open source code to implement them is provided on the web page for this paper at `www.ddnum.com/software/simulator`

2. Problems with Backtesting on Market Data

Backtesting is the process of applying an investment strategy to past market data in order to predict future returns from that strategy. There are a number of problems with the concept of backtesting. We refer readers to the book Aronson (2007) for a general discussion or to de Prado (2013) for the mathematics of backtest overfitting.

In particular the fundamental problems of backtesting that are most relevant to this paper are:

1) market data has noise in it
2) markets are non-ergodic

Markets consist of long term trends swamped by cycles and one-off trends and "jump" events all of which are swamped by noise. That gives rise to most of the problems.

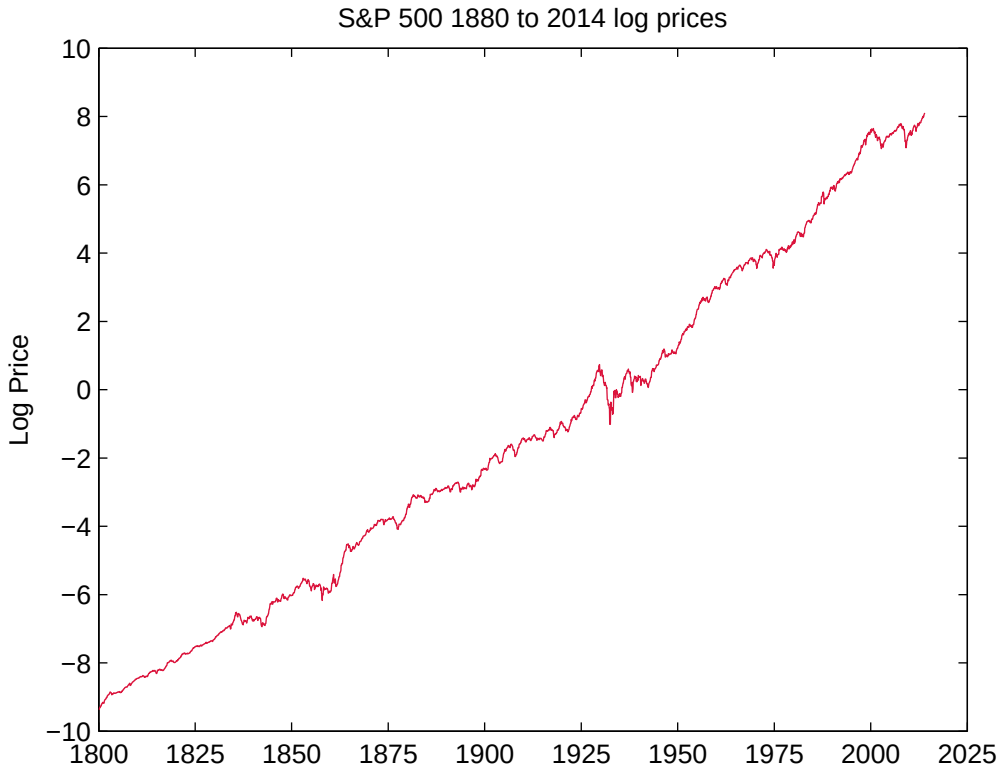
(1) is a problem because noise is, by definition, unpredictable and non-repeatable. To the extent that the investment strategy being tested latches onto noise for prediction (or incorporates it into an optimization) the strategy will suffer in the future because noise is non-predictive. This problem has various names such as *data mining bias*, *fool's gold*, *overfitting*, *snooping*, *fishing*, *data dredging*, *data torturing*, or *using the data twice*. The latter refers to the process of using past data to optimize a strategy and then using the same data to predict the future returns. This is mathematically unsound in the presence of noise in the data. For an entertaining paper title and some mathematical theory see Bailey et al (2014) "Pseudo mathematics and financial charlatanism: the effects of backtest overfitting on out-of-sample performance." We will show some examples below.

(2) is a problem because backtesting assumes that the market in the future will in some sense resemble the market in the past. Many mathematical proofs in finance start off "let {Xi} be a stationary[2] ergodic stock market." Ergodicity of a sequence of data is the property in which every sequence or sample of sufficient size is equally representative of the whole. The stock market is composed of secular regimes that will never be repeated (e.g. the emerging market boom of the 90's or the Global Financial Crisis of 2008) as well as regimes that probably will be repeated but in varying durations and intensities (e.g. bull and bear regimes). It is definitely not ergodic (Horst and Wenzelburger 2008) and we now show a disconcerting recent example.

[2] Stationary here means that the probability distribution of the market returns does not shift with time

3. The S&P 500 – Trending or Mean Reverting?

We show below a chart of the S&P 500 as extrapolated back to 1800 by Global Financial Data and plotted on a log scale. The market appears to have been remarkably consistent over the last 200 years. Indeed, this consistency is evident for the stock markets of many other countries when charted back 100 or 200 years (Credit Suisse 2014). But this consistency is an illusion and it has serious ramifications for backtesting investment strategies.



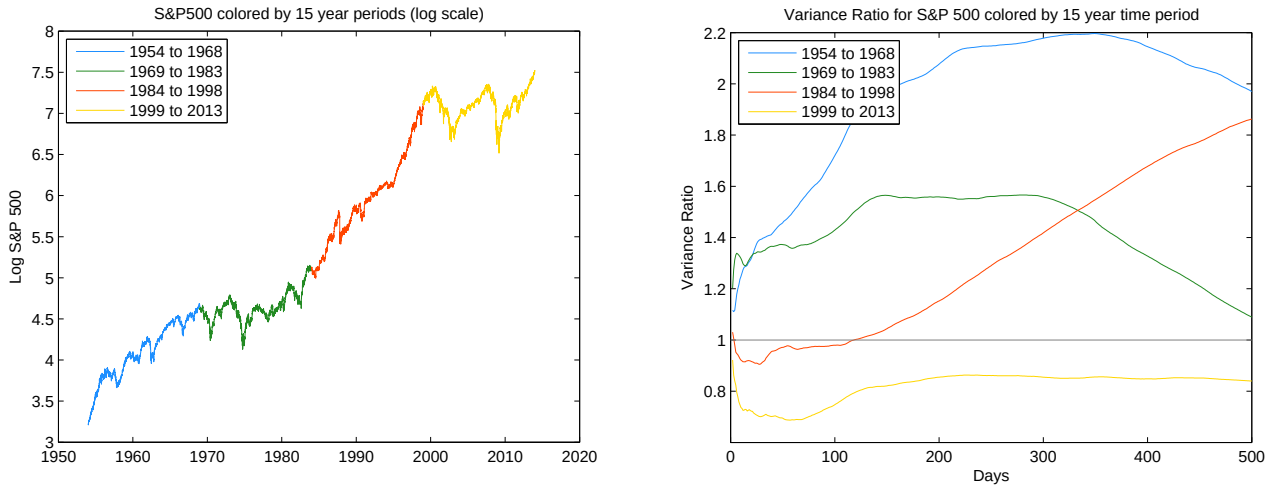
Any backtesting strategy will, presumably, use the most recent years of stockmarket data. Let us consider the most recent 15 years – a popular time period (especially for the development of Exchange Traded Funds) – and compare it with previous 15 year periods.

Let us consider, for example, the mean-reverting or trending tendencies of the S&P 500. We define the *variance ratio* (VR) of a series of prices to be the ratio of the *k*-period return to *k* times the variance of the 1-period return. The idea behind this is that when returns are uncorrelated over time, the numerator and denominator should be the same. So the VR should be 1.

But in a mean-reverting market the returns are negatively correlated and the VR will be less than one. In a trending market the returns will be positively correlated and the VR will be greater than one. So we

can calculate the VR over various periods *k* and tell if the market is trending or mean-reverting over that period.

Here is an example. The chart on the left shows the S&P 500 plotted on a log scale and colored by 15 year time intervals. The chart on the right shows the variance ratios for each 15 year time interval.



The *x* axis in the VR chart is days. We see, for example, that for the 1999 to 2013 time interval (in yellow or the lowest line in the VR chart) that over time periods of 1 to 100 days the VR is less than 0.8 so the market is mean-reverting over this time frame. For more than 100 days the VR is a little higher – about 0.86 – so the market is still mean-reverting but less so.

Compare this to the other 15 year time intervals. The VR values are much higher than one – indicating trending markets. Mean-reversion strategies won't work for these markets (except perhaps for short term less than 30 day strategies in the 1984 to 1998 market).

The last 15 years appears to have been atypical. What is the value of a backtest that covers this period? We suspect not much. Without ergodicity we cannot make any forward-looking predictions that we have confidence in.

We didn't need the VR to tell us that the last 15 years has not been typical of previous years. Visual inspection of the left hand chart tells us that. The market appears to have become cyclic. There are three peaks and two troughs – just *five* points. In statistics a sample of size 5 is minute – so small as to be

almost worthless. Any strategy that gets close to these turning points – and it's easy to devise one – will produce spectacular returns in what is essentially a sideways market.

This is not typical of the market of the last 200 years. We should be extremely wary of extrapolating the last 15 years forward to the next 15 years. That strategy is fraught with danger.

4. Timing, Noise, and Lag

A third problem with the way backtesting is typically carried out is:

3) Backtesting seeks to maximize returns instead of optimizing timing

Conventional model building and backtesting tries to optimize timing by maximizing returns. The distinction between optimizing returns and timing is subtle. It may not appear to even be useful. If we can optimize timing then optimizing returns will surely follow. This is true to a certain extent but a focus on optimizing returns has some flaws.

Market timing drives excess returns. But the observed excess returns have extra included noise. The formula is:

observed excess returns = excess returns from market timing + noise

The right hand side might be expressed in engineering terms as *signal + noise*. In stock markets the signal is invariable swamped by the noise. It is the detection of the signal that we are trying to optimize but all we can optimize through backtesting is signal + noise. We are optimizing the wrong quantity when we try to optimize returns.

In stock markets the noise is particularly capricious. It can be "fat tailed," can have large one-time jumps (or outliers), and may not even have a finite variance. It can lead backtesting algorithms astray.

For example, the crash of 1987 was a single-day event that took 20% off the market. Any backtesting strategy that includes 1987 could "accidentally" avoid that bad day and get an undeserved 20% boost in its returns.

In an attempt to remove noise from returns and to focus on the signal we can “smooth” out the noise by using the tendency for noise to cancel out if averaged over multiple instances. A popular example is the moving average.

But when those averages involve multiple time periods we introduce lag into detection of the signal. Lag means that we miss the optimal timing in practice but this may be OK if the lag is not too long. But how long is too long? For a short-duration signal the lag may be too long to catch the signal. Even long duration signals if weak enough could require too much averaging to be useful.

Backtesting cannot really give us an indication of whether or not the lag is acceptable. It is too uncontrolled an environment. We don’t know for sure if a strategy failed or succeeded simply due to chance. So we come to our fourth problem with backtesting:

4) Backtesting does not give us confidence intervals

A backtest is a sample of size one. A sample of that size does not allow any statistical statement of confidence to be made. This is regardless of whether the market is ergodic or not. If the market actually was ergodic then we could divide the time period into smaller periods (as we did with the variance ratio) and get, under certain conditions, a small increase in sample size.

But there is so much volatility in daily returns compared to the size of even the largest signals that it turns out that we need far more data than we actually have. How do we know this? We used simulation.

5. Simulation as a Testing Strategy

We have avoided defining what we mean by “signal” and “noise” in stock markets. This is because what may be noise to one trader may be signal to another. For example a buy-and-hold long term investor may be expecting an annualized return of 8% in the market. For such an investor the daily fluctuations of returns around that 8% constitutes noise.

But for a day trader who is only in the market intra-day the daily return fluctuations contain enough signal to be profitable (presumably). So there is no absolute definition of market noise. But this is no problem for simulation. For repeated simulations noise is what is different each time, signal is what

remains constant. Meucci (2009) calls the quest for prediction models the quest for market invariants – those phenomena that repeat themselves identically throughout history. Anything else is noise.

Simulation is where we simulate a signal, simulate some noise, add the two together, and then backtest the simulated data. We control what is signal and what is noise and there is no ambiguity. Its chief advantage is that we are in full control of the market – we know the signal and we know what part of returns are signal and what part are noise. We play the role of "Mr Market."

We provide two full examples of this in the rest of the paper. But a brief example for now will illustrate the principle. Suppose we want to test the usefulness of the seasonal Halloween Indicator as a signal to stay out of the market for six months of the year. Maybe staying out costs us more than we gain.

We test this by measuring the returns and variability of the in-season and out-of-season market over the last, say, 15 years, and simulating a market with those parameters. By simulating over a *million* years we can make statements such as "with these parameters the Halloween indicator isn't profitable" or "to be profitable the Halloween effect must be *this* big" or "to test the Halloween indicator if it is *this* big would require 300 years worth of data to be 95% confident that the effect is real."

Backtesting in general consists of two elements:

i) does the signal we are testing actually exist in the market?

ii) is the signal to noise ratio large enough to allow the signal, if it exists, to be exploited?

Simulation backtesting mainly addresses (ii) but is also useful for addressing (i). The iterative procedure in using simulation is: (a) form a hypothesis about a market strategy, (b) gather parameters from the market, (c) test it in the simulator then either go back to (a) or (d) test it in the market. Then repeat from (a) again.

The advantages of simulation backtesting are many:

- we know the optimal strategy (we are Mr Market)
- our focus is on hypothesis testing and understanding the market rather than seeking returns
- it avoids using the data "twice"
- we can get an upper limit to the returns achievable with a given strategy in a market with given noise

- we can get a feel for the current market (by mimicking aspects of it)
- we can see if a market inefficiency is rendered ineffective due to too much noise or lag
- we can do experiments
- it solves the sequential parameter estimation problem (see below for explanation)
- we can ask questions we wouldn't consider asking in the real market
- we can optimize for the signal directly without doing it indirectly by optimizing returns
- we can measure how much noise a strategy fit (and quantify the amount of overfitting)
- we can test and optimize different methods for estimating parameters
- it gives us an appreciation of noise and which strategies are more resistant to it
- we can repeat the simulation process and get confidence intervals for our estimates

The sequential parameter estimation problem is a tricky one to solve using backtesting on real data. It concerns the case where we have, say, 15 years worth of data and have some constant market parameter that we want to estimate. An example might be the long run mean volatility of the market.

At the beginning of the 15 years we know nothing about the parameter and have to estimate it with little data. Then by the end of the 15 years we know the parameter quite well. We want to estimate the returns going forward so are tempted to ask “suppose we knew the parameter to be this value at the beginning of the 15 years, what would our returns have been?” so we run a second backtest with the estimate from the first.

The problem here is that our first backtest *underestimated* our future returns because it didn’t have a good estimate of the parameter. And the second backtest *overestimated* our future returns because it used the same data twice (once to calculate the parameter and once to calculate the returns).

Dividing our sample into two 7 year periods reduces the bias but means our estimate of future returns comes from only half as much data so loses predictability. It doesn’t solve the problem.

Simulation solves this problem because we can simulate the whole 15 years and exactly measure the bias. Then we can test strategies for reducing it (maybe averaging the over and under estimates will work).

6. The Simulator

We have built a market simulator that can be used to test a wide range of hypotheses about the markets. The simulator has two aspects to it – one of fitting parameters to the market and one of simulating markets with the fitted or any given parameters.

We have used the programming language MATLAB for the simulator but intend to convert it into the free programming language R (R Development Core Team, 2012) and to make the code available for free on our web site `www.ddnum.com/software/simulator`. R has far more features for fitting and simulating statistical models than MATLAB.

Features in the simulator to date include the ability to simulate:

- **noise** – from Gaussian distributions, fat-tailed distributions such as Student *t*, distributions with given skewness and kurtosis (the Pearson family), and heteroskedasticity – using GARCH processes.

- **random processes** – such as Brownian (random walk with parameters such as drift and volatility), Brownian Bridge (random walk with both ends constrained – useful to ensure a given return), Heston Model (allows correlation between returns and volatility), Ornstein–Uhlenbeck (useful for mean-reverting processes), and fractal processes.

- **time-varying parameters** – parameters can be varied according to simple sine waves, sums of sine waves, asymmetrical sine waves (useful for cyclical markets where bull phases are longer than bear phases), sawtooths, random switching (between, say, bulls and bears), autoregressive processes, random walks, and time-varying Hurst exponent.

- **alpha and beta parameters** – for simulating mutually correlated multiple assets according to the Capital Asset Pricing Model (CAPM), also we can have time varying alpha and beta values.

- **regime change** – detection and modeling through use of Hidden Markov Models

The Hurst exponent (H) is a parameter of a fractal process and has the useful property that $H < 0.5$ means the process is mean-reverting, $H = 0.5$ is a random walk, and $H > 0.5$ is a trending process. We

can vary the Hurst exponent continuously (using any of the time-varying features) to simulate a market that moves between mean-reverting and trending phases.

The idea behind having so many different simulation features is that we can try them all in a sensitivity analysis to see how an investment strategy is robust to various market properties and conditions.

7. Example 1 – Optimizing Moving Average Crossover Rules

In this section we emulate what a backtester (person or algorithm) might do to devise a trading rule to trade a given market.

The basic backtesting strategy is to devise a rule, apply it to past market returns to optimize the rule parameters, then use those optimal parameters to trade the market going forward. The logic behind this strategy seems clear: our best guess, in the absence of any extra information, for trading in the future is what worked best in the past. Since markets can be fickle we may not have confidence that this rule may actually be the best going forward but it is difficult to see how we could choose anything better.

As an example consider the much-beloved Moving Average Crossover Rule. It is a simplistic momentum strategy and serves as a standard illustration for quantitative techniques. We don't advocate the strategy but note that it can be quite profitable – as an example the MA(200, 50) strategy where the 200 and 50 refer to days is known in the industry as the "golden cross" and apparently works well for many markets. So we use it as an example in this paper.

This strategy says to calculate a short period (call it n) moving average and a long period (call it m) moving average. Then trade long if the former exceeds the latter else stay out of the market. We abbreviate this to the MA(m, n) rule. The strategy works well when a time series enters a period of strong trend and then slowly reverses the trend.

The problem that we deal with is how to estimate the best values for m and n. In particular, how do we estimate it when the only data that we have is market returns for the last, say, 15 years.

An obvious procedure is to look at what values of *m* and *n* worked best in the past and use those going forward. It doesn't seem possible to improve on that. Surely the best estimate, without any extra data or predictions about the future, of what will work in the future is what worked in the past.

The flaw in this view is that it fails to deal with the presence of noise in our estimates of *m* and *n*.

We now show how noise manifests itself in backtesting and how we can remove some of it to improve our estimation of the best parameters for trading in the future. We start with our simulator.

8. Return Surfaces for Moving Average Crossover Rules

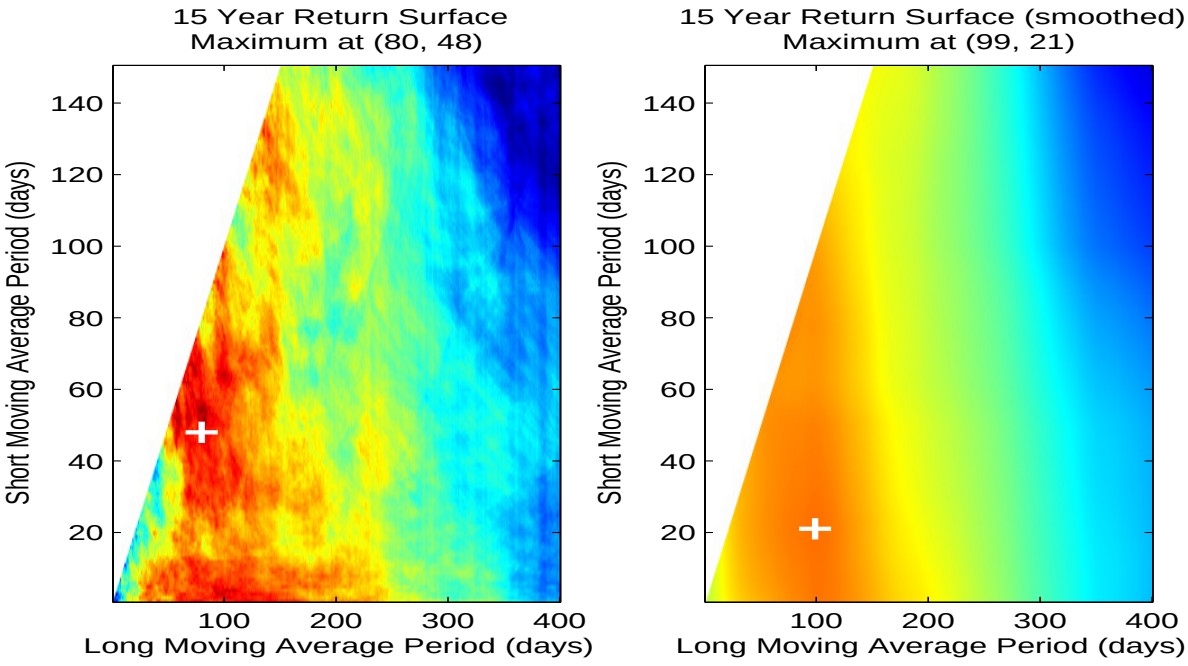
Using the simulator we generate sequences of returns for the S&P 500 for time periods of 15, 30, 60, and 480 years. We then apply backtesting to each sequence and look at the results.

If our simulation was simply a random walk with constant drift then the returns would not be time-varying and we would not be able to time the market. To simulate time-varying returns we generated a sum of two sine waves – a major wave with period of 1500 days and a minor wave of period 300 days for the bull markets and a major wave of period 500 days and a minor wave of period 250 days for the bear markets. We chose these numbers because they resemble the S&P 500 market of the last 15 years. We chose round numbers and did not attempt to accurately fit the market because an accurate fit is not required for this exercise (we know this from doing a sensitivity analysis).

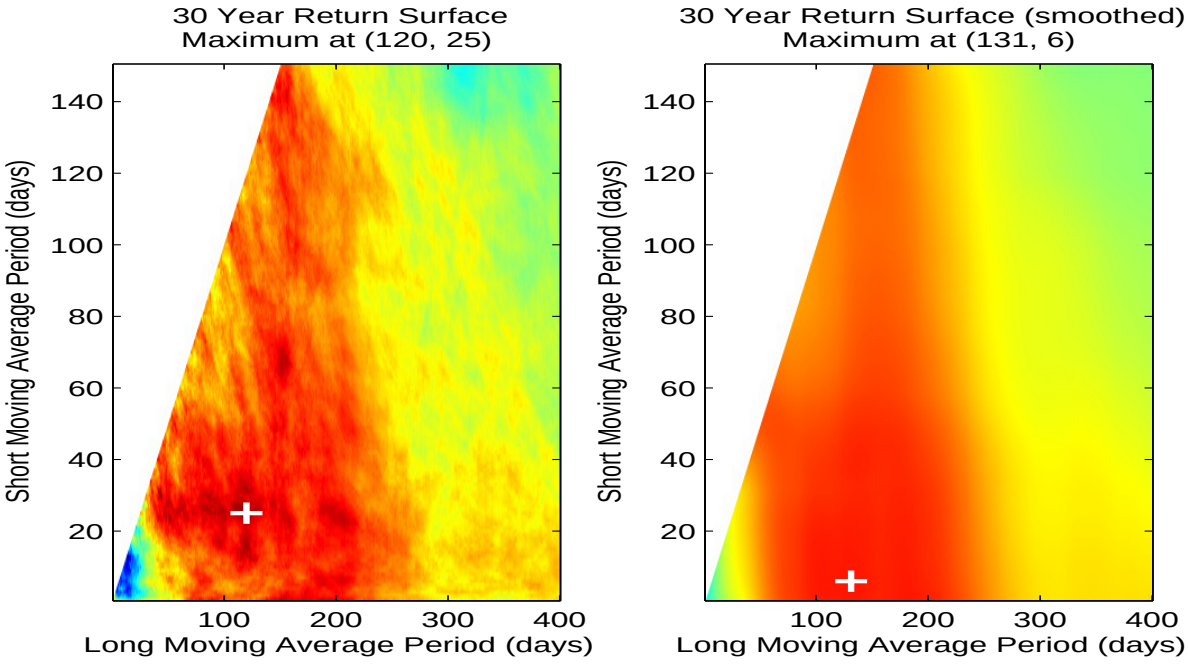
For each simulated sequence we try every possible MA(*m*, *n*) rule for *m* from 1 to 400 and *n* from 1 to 150 (note that *n* cannot exceed *m*) and plot the resulting returns in a heat diagram. In each pair of charts below the results are shown in the left hand chart. The point of maximum return is shown with a white cross – this is expected to be the trading rule that the backtester would declare optimal.

In the first chart we see the white cross at MA(80, 48). We ignore the return at that point because it is irrelevant for this discussion. What we are interested in is where the white crosses lie.

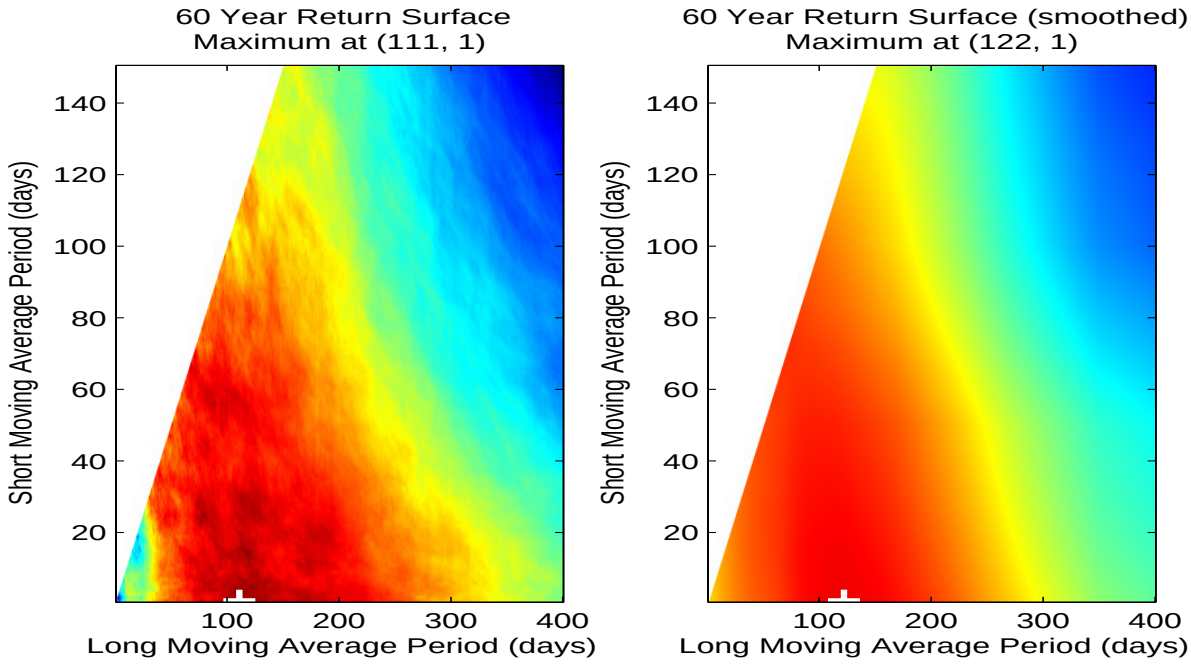
Because we set the parameters for this market in our role as Mr Market we know that the actual optimal MA trading rule is MA(134, 1). The 15 years of data produced an estimate of (80, 48) which is a long way from the optimum. So 15 years of data is not very much for this kind of backtesting.



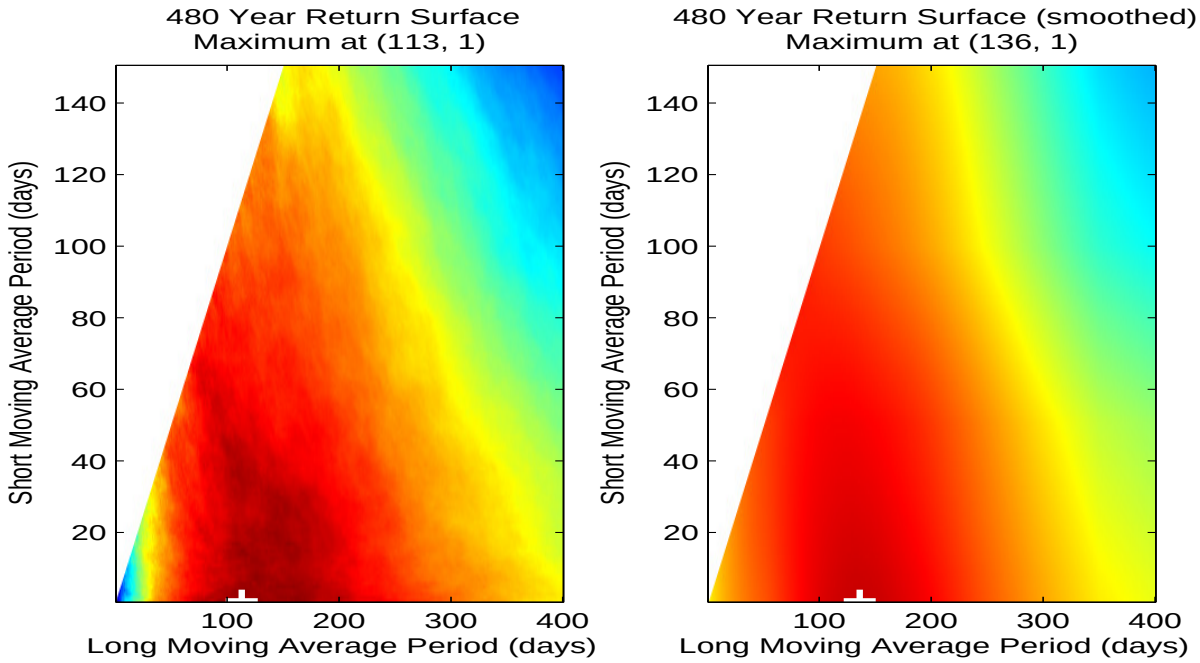
Now try 30 years. The 30 year return surface is below. The optimum is at (120, 25) so we are getting closer. How about 60 years?



The 60 year surface below shows a maximum at (111, 1) so we are getting closer to the true value.



But after 480 years we are still not close. We chose the figure of 480 years for other reasons and we will discuss that below.



9. Reducing the Noise – Smoothing

We notice that the surfaces, if pictured as 3-D objects are "rough" and it is tempting to smooth them. It's not obvious if smoothing will work though and it's not possible to test using real market data. Simulation is our only hope – we show here that it will work. Also it allows us to determine the optimal amount of smoothing which turns out to be higher than we intuitively expected.

We consider three different smoothing algorithms for smoothing the surfaces. There are more available but three is enough to illustrate the principles involved.

(1) The first we call *smooth2a* which is a simple two-dimensional local averaging. It replaces each pixel with the average pixel value of all pixels within a distance *s* of that pixel. It is quick and easy to understand but since each pixel is replaced with an average of nearby pixels it tends to flatten the peaks and troughs of the surface. Also it only works for two-dimensional surfaces whereas we would prefer something more general.

(2) The next algorithm we call *smoothn* which is described in Garcia (2010). This fits spline curves in multiple dimensions and attempts to preserve curvature in the surfaces. It is a robust smoothing that minimizes the influence of extreme data and works for more than two dimensions. The smoothing parameter *s* determines the amount of curvature allowed in the splines and thus the degree of smoothing.

(3) The third algorithm we call *ksrmv* which is a multivariate kernel smoothing regression. This places a small hill (or kernel) at each data point and adds up the hills to create the final surface. The technique is a standard one in mathematics and we used a 2008 implementation of the Nadaraya-Watson (1964) kernel regression provided by Yi Cao of Cranfield University.

In the right hand chart of each of the pair of figures above we show the effect of smoothing using *smooth2a* and s = 30. The other smoothing methods give results that look similar. It is evident that smoothing before finding the maximum finds a maximum closer to the optimal point than not smoothing. Smoothing appears to improve the estimates of the optimal MA rule. We prove this in the next section.

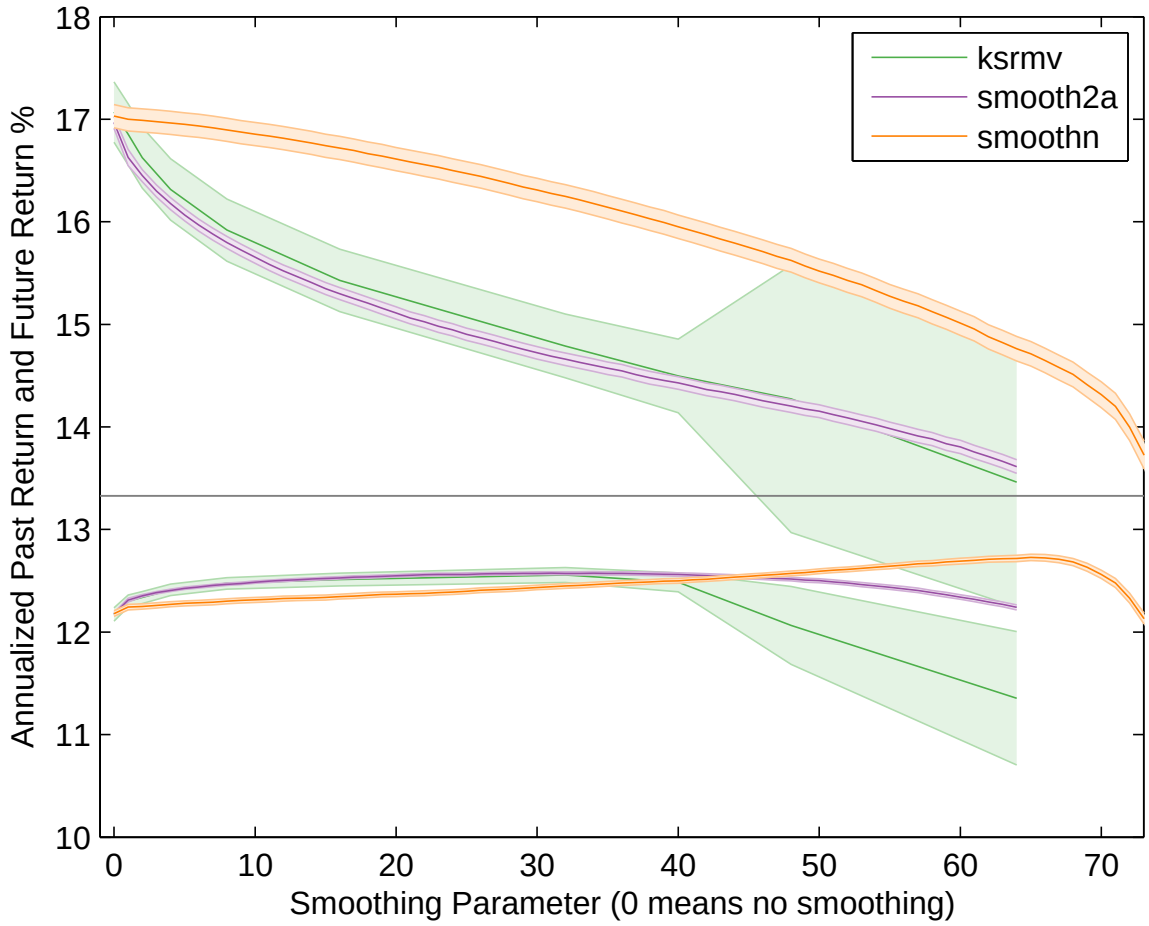
One remark that we would like to make here before moving on is that the optimal rule has $n = 1$. This is a bit unintuitive – we would have expected the optimal n to be higher to reduce variability in calculating

the short term moving average. Surely $n = 2$ would greatly reduce the variability and improve the trading rule. But no – the optimal rule has $n = 1$.

And the really interesting part of this finding is that whenever we have determined the optimal value for n in other types of simulations and in sensitivity analysis we have always found the optimal value to be 1. So we speculate that the optimal value for n is always 1. So much for the golden cross where n is 50. The investigation of this phenomenon is a topic for future papers.

10. The Backtesting Results Chart

This chart is a key chart of the paper and deserves a careful examination.



Along the horizontal x axis is the value of the smoothing parameter s ($s = 0$ corresponds to no smoothing, as s increases the amount of smoothing increases). The three colors in the chart are for the three different smoothing methods. For $s = 0$ there is no smoothing so all three lines converge to the same point. The

actual smoothing methods used are not as important as the fact that the results differ when using different methods.

To aid in judging differences 95% confidence intervals are drawn (using shading) for each curve in the chart. The intervals for the method *ksrmv* are wider than for the others because *ksrmv* requires copious amounts of CPU time so we were unable to run as many simulations as for the other methods.

The vertical *y* axis shows the mean annualized return of the various methods. The gray horizontal line is the expected return of the optimal moving average crossover strategy. This return is 13.39% and results from the optimal MA strategy which is the MA(134, 1) rule. We know this number exactly because we are Mr Market.

We chose a time period of 15 years and simulated several thousand instances of our market. For each simulation we calculated the MA empirical return surface as a backtester would do when backtesting the 15 years. Then we picked the MA empirical trading rule that maximized the return on that empirical surface. This is what a backtester would do when choosing a MA trading rule to trade going forward.

In addition, for each empirical surface we smoothed the surface using a specified smoothing method and amount of smoothing *s* and calculated new "smoothed" trading rules, one for each value of *s* from 1 to about 70. Thus each surface gave us an "unsmoothed" trading rule and 69 smoothed rules.

For these 70 trading rules we calculated the actual empirical past returns and the future expected returns (which we know because we are Mr Market). We did this over several thousand simulations and plotted them (and confidence intervals) in the chart for each value of *s*.

All values above the gray line are the actual past empirical returns, all values below the line are the expected future returns. An example may make this clearer.

Consider, for example, the case where $s = 0$ on the left. We see that the mean past return was about 17%. This means that on average a backtester who backtested 15 years worth of data in this market and used the best MA trading rule for that period would have measured the past return for that rule to be 17%.

This is clearly an overoptimistic estimate of future market returns because as we already know, the best possible trading rule has an expected future return of only 13.39%. That is a well-known peril of

backtesting, that if it includes an optimization step it will overestimate future return (Aronson, 2007, calls this overestimate "data-mining bias" or "fools gold").

Just how much does it overestimate future returns? The three lines below the gray line show expected future returns and we can see that for $s = 0$ expected future returns are about 12.2%.

The chart shows another bias – the "small sample" bias. The described backtesting methodology on average produced trading rules that are less than optimal since 12.2% is less than the optimal return of 13.39%.

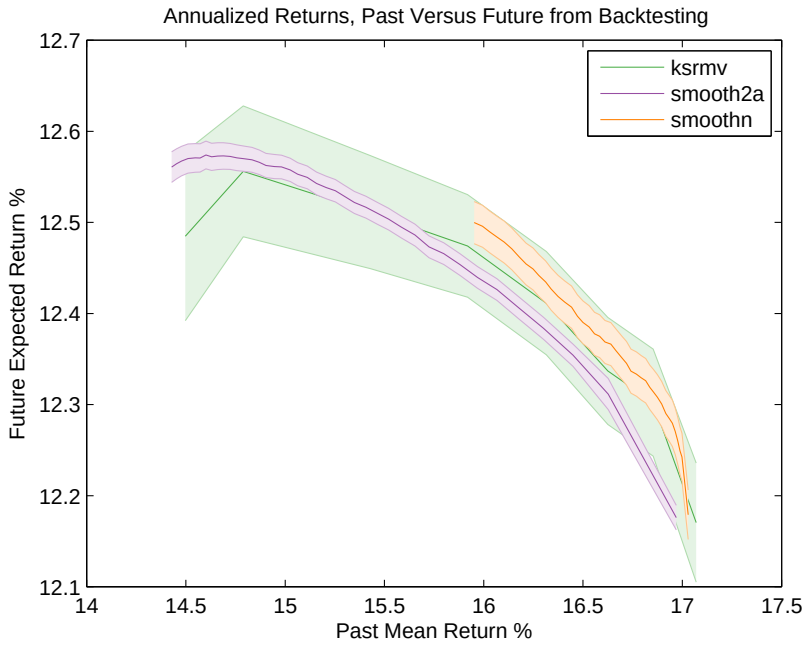
This is where smoothing comes in. By applying various amounts of smoothing to the empirical return surfaces we generate from backtesting we get better estimates of the optimal trading rule and better estimates of future return (less data-mining bias).

The former can be seen in the lower part of the chart – as *s* increases the future expected returns increase. Up to a point where we reach the optimal amount of smoothing. It looks like the optimal smoothing method is *smoothn* and the optimal value of *s* is 65 which produces an expected future return of 12.7%. This is better than the unsmoothed future return of 12.2%.

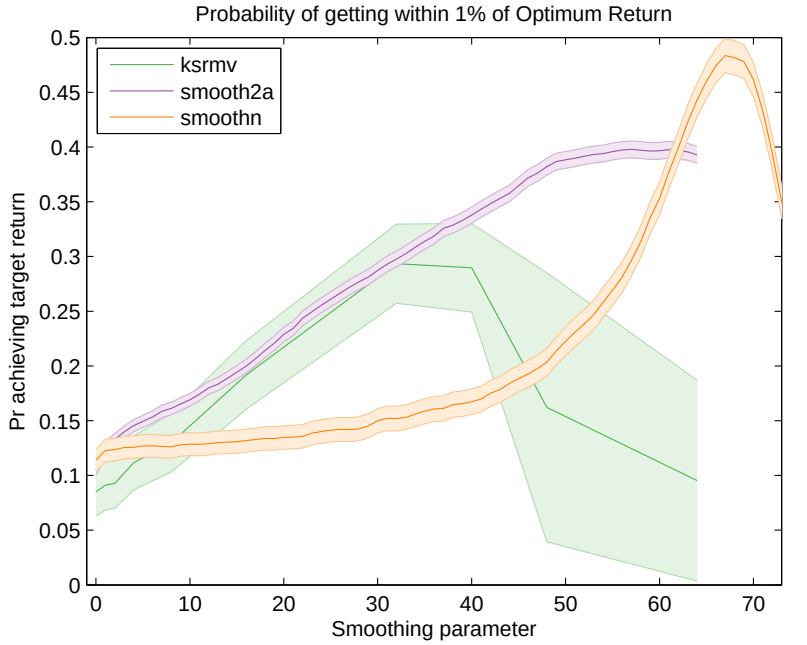
The data-mining bias is the difference between the upper and lower lines of the same color (the difference between past returns and future returns). This decreases with increasing *s* but starts increasing again with excessive smoothing.

A lesson from this chart is that optimizing past returns does not necessarily optimize future returns. This is unintuitive to a certain extent. We show it most dramatically in the next chart. Using *s* values in the range of 0 to 40 (the chart starts getting messy after 40) we plot for the three smoothing methods past mean returns versus future expected returns. The relationship is clearly negative.

Negative means that the higher the past returns, the lower the future returns. *This, more than anything else in this paper, shows the dangers of backtesting*. If you try to maximize backtested returns you run the risk of reducing your actual future returns.



Another way of looking at the success of backtesting is to calculate the probability of getting "near enough" to the optimal rule. If we define near enough to mean getting within 1% of the best possible return from a moving average crossover strategy then we get the next chart. This shows that zero smoothing gives us a probability of 0.1 of getting close. But by more smoothing we can get as high as 50% chance of getting close enough.



11. Example 2 – AAII Portfolio Optimization

The American Association of Individual Investors (AAII) maintains a selection of investment strategies (that they call screens) applied to the U.S stock markets. They provide monthly return figures for 83 strategies and indexes from 1998 to now (theoretical returns assuming no trading costs). The strategies have been garnered from books and academic papers and are listed on their web site at http://www.aaii.com/stock-screens.

The strategies can be classified into nine styles - *Value*, *Value with Price Momentum*, *Growth*, *Growth with Price Momentum*, *Growth & Value*, *Growth & Value with Price Momentum*, *Earnings Estimates*, *Specialty*, and *Indexes*. Each strategy consists of a collection of stocks of the same style so behaves similarly to an Exchange Traded Fund.

Due to the groupings by similarity within style we suspect that the strategies will have time varying performances depending on which style is in vogue in the market at any one time. If true then we may be able to time the strategies and switch between them to enhance our returns.

We ask: is the dispersion in returns large enough and the noise small enough to allow us to perform market timing? Backtesting is unlikely to answer this question as we only have 192 monthly returns since 1998 and this is woefully inadequate for assessing statistical significance.

Plus there is another problem – two of the strategies (the *Piotroski* ones) are outliers in their extremely high performance (see the charts below). The effect of this is similar to the effect of the 1987 crash – it heavily rewards investment algorithms that may undeservedly stumble upon the high performing outliers.

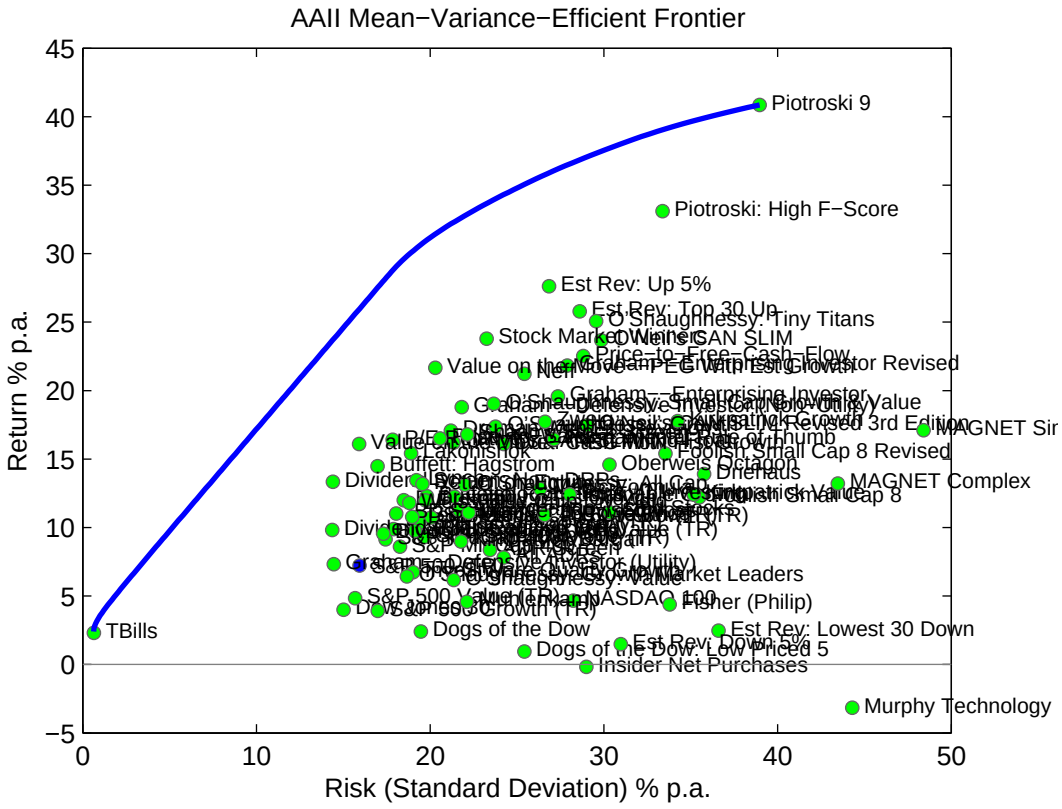
We are tempted to remove the outliers to produce a more robust assessment of any investment algorithms. But we also want to make sure that if there are any outliers then our algorithms will latch onto them. So we cannot remove them.

The only way to resolve this conflict and to solve the problem with lack of data is to use simulation.

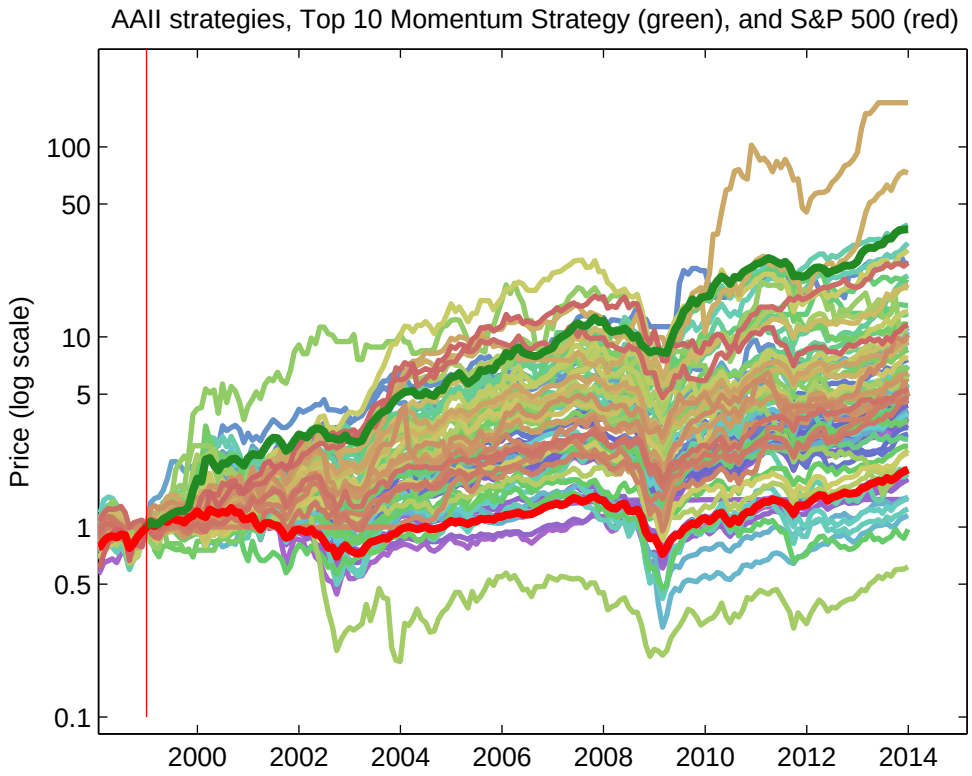
We constructed the simulation by measuring the alpha and beta of the 82 strategies against the 83rd (the S&P 500 index) according to the standard CAPM model. The alphas and betas vary with time so we

needed to use a time-varying regression model. The dlm package in R (Petris et el 2009) provides estimation for this model. Then we fitted sine curves to the varying alphas and betas. We also fitted sine curves to the S&P 500 and this process gave us in total all the parameters required for a complete ergodic market simulation.

We plot below the mean-variance diagram for all 83 AAII strategies and indexes. Although they are a bit of a mess we leave in the labels of the strategies for the benefit of readers who may be familiar with them. It should be noted that the two strategies with the highest returns are the *Piotroski* strategies. The S&P 500 index is shown as a blue circle. The efficient frontier is shown as the blue curve.



The next chart shows the equity curves for the AAII strategies and indexes. The S&P 500 is shown in thick red (lower thick line). The thick green (upper) line is the M(4, 4) algorithm which will be explained in the next section.



12. Timing the AAII Strategies

We now consider timing algorithms for the AAII strategies. The simplest might be the momentum algorithm. For multiple assets this is an improvement over the moving average crossover rule because it allows us to rank assets relative to each other.

Momentum rules have been used by practitioners for decades but it was only recently that they were recognized by academics – Jegadeesh and Titman (1993). Momentum is the tendency of investments to persist in their performance whether good or bad. This persistence implies predictability – past returns predict future returns. One question is: what is the optimal past period (or lookback period) which best predicts future returns?

When considering multiple assets there are two types of momentum: cross sectional momentum (CSM) and longitudinal momentum (LM). CSM (also called by other terms such as relative strength) is the tendency for relative performance among assets to persist – so the best performing assets tends to remain the best performing. Thus a CSM algorithm might invest the most money in the top performers of the group of assets.

LM (also called by other terms such as time series momentum or absolute momentum) is the tendency for a single asset to persist in its performance – either positive or negative. So an LM algorithm might invest the most money in assets that have positive past returns or returns greater than a threshold.

The combined momentum algorithm (also called by other terms such as dual momentum or constrained relative momentum) invests the most money in assets that have the highest CSM provided that they have positive or high LM.

This gives rise to some questions: what are the optimal cutoff thresholds (the values for which we exclude an asset from the portfolio) for the two momentums and what are the optimal lookback periods for them when applied to the AAII strategies? Also what is the optimal number of assets to hold?

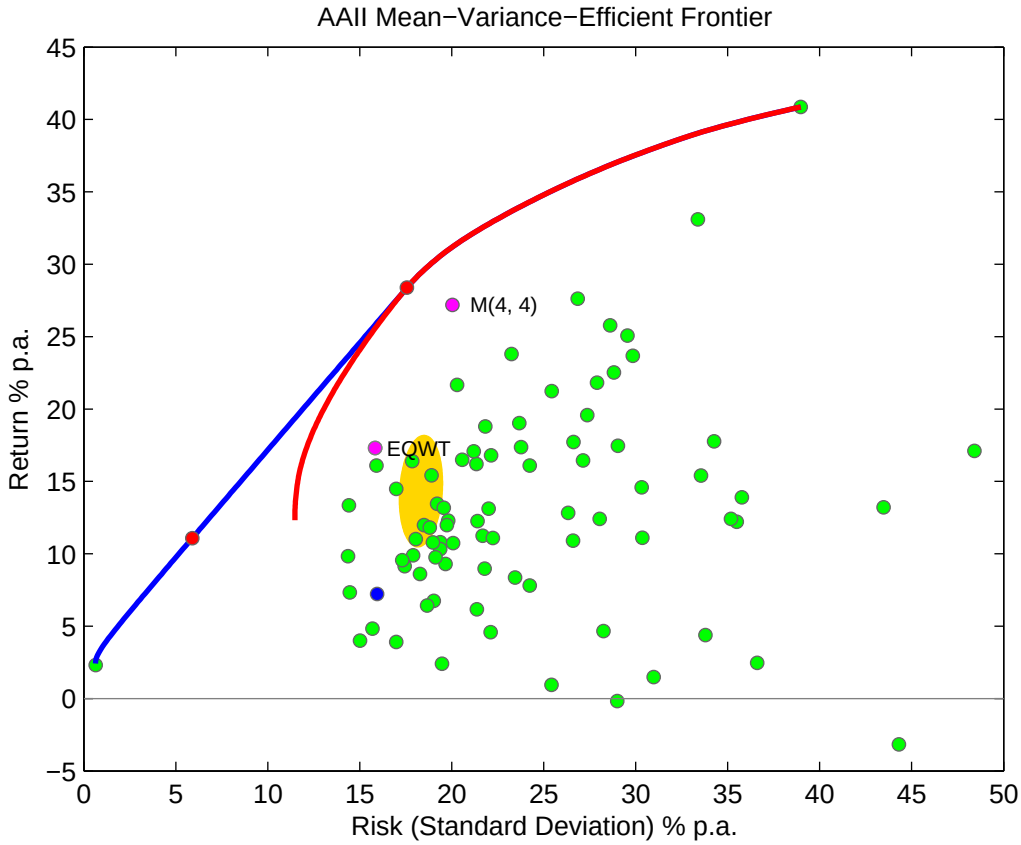
This is a five parameter optimization problem. A quote attributed to John von Neumann is "with four parameters I can fit an elephant, and with five I can make him wiggle his trunk," (Dyson 2004). Fitting five parameters with just 192 observations is fraught with danger. We have no choice but to resort to simulation.

For expository purposes we consider just three of these parameters: the optimal lookback periods and number of assets to hold. A momentum strategy with CSM lookback period of *n* and LM lookback period of *m* is labeled M(*n*, *m*). If the momentum rules find no assets to invest in we stay out of the market and, for simplicity, assume a cash return of zero.

The next chart shows the results of a M(4, 4) strategy with 10 assets. We cherry-picked these parameters because they produce a good return which suggests that momentum may work here. But is it a significantly better return or just due to luck (or *Piotroski*)? Are these parameters optimal or can we find better ones?

The chart shows the AAII strategies on a mean-variance diagram. The S&P 500 is the blue circle. The blue frontier is for the 83 assets including T-Bills, the red for omitting T-Bills. The red circles are the tangency portfolios which give the highest Sharpe Ratios. The two labeled magenta circles are the equal weighted strategy (EQWT) which equally weights all strategies and rebalances each month and M(4, 4) is the momentum algorithm with 4 month lookbacks and rebalancing each month into the top 10 assets.

The yellow ellipse is the 95% confidence interval on the combined results of 1000 monkeys investing at random each month into 10 assets. Any algorithm that falls outside the yellow area we can say with 95% confidence is different from random.



We note that equal weighting produces less volatility than the monkeys – this is due to the monkeys holding only 10 assets whereas EQWT holds 82 (no T-Bills). It is also higher than the mean monkey but this may be due to chance or it could be due to the monthly rebalancing benefiting from some mean reversion in the prices.

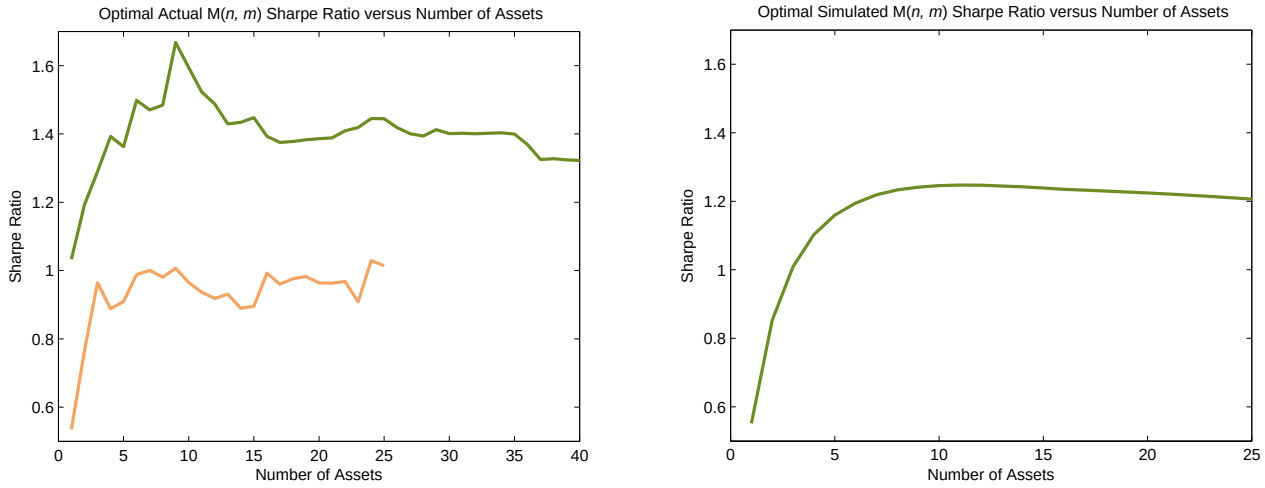
The M(4, 4) algorithm greatly boosts the returns over EQWT at the cost of a somewhat lesser boost in volatility. It is significantly different from the monkeys but this isn't useful information when it comes to optimizing the algorithm. What we really want are confidence ellipses for the M(*n*, *m*) algorithms but these are not possible with so little data. This is where simulation comes in.

For this example we focus on Sharpe Ratio – we could look at any metric such as return or volatility or maximum drawdown. We simulated 15,000 instances of the AAII strategies as described above. For each instance we calculated the Sharpe Ratio (SR) for each combination of *n* and *m* from 1 to 24 and

each number of assets from 1 to 40. This produces 23,040 parameter combinations so we ended up calculating about 66 billion returns and 346 million Sharpe Ratios. This took a whole day.

We calculated the mean of all the SRs for each parameter combination. Each mean was of 15,000 observations and this was enough to give us three decimal digits of precision in the means – enough to regard the means as exact and to be able to ignore sampling error. This meant that we could calculate the optimal strategy exactly. We now look at the results.

The maximum Sharpe Ratio obtained was 1.25 for M(23, 23) and 11 assets. By itself this figure isn't very useful. Let's look, for example, at how the optimum SR varies with the number of assets.



The chart on the left (the green upper line) shows the actual empirical backtested optimal SRs for each number of assets (up to 40). The chart on the right shows the exact results for the simulation (only up to 25). We see a steep rise followed by a period of relative stability in the exact results. We view the left hand chart in this light and we postulate that even though the actual optimum occurs at 9 assets this is a noisy estimate and perhaps a little riskily too close to the steep drop off to the left. So we surmise that the best value to use going forward is in the 11 to 15 range. We pick the value 11 since it looks stable around that value and the simulation has its optimum at 11.

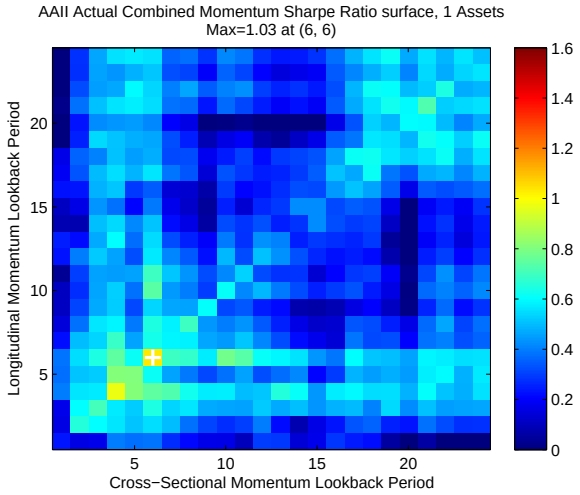
The simulation has not provided us with a definitive unique answer to the optimum size of the AAII portfolio but it did provide us with enough insight to be suspicious of the backtested optimum of 9 and let us settle for a "more reasonable" 11.

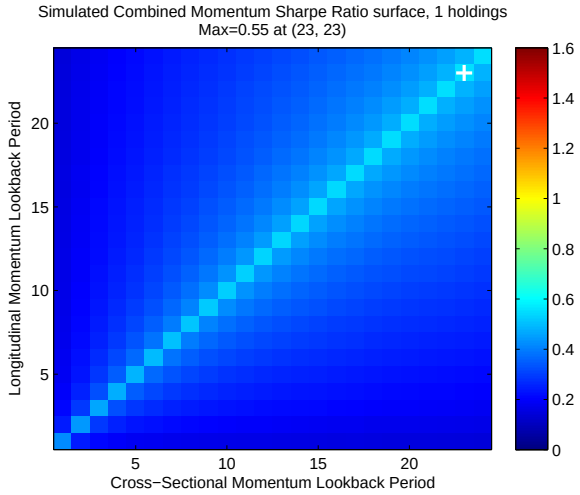
The backtested SRs are mostly around the value of 1.4 whereas the exact simulated SRs are around 1.2. We cannot say where this difference comes from – it may be due to inaccuracies in the parameters chosen for the simulation and also due to overfitting (data mining bias) in the estimation of the optimal *n* and *m* parameters. So we wonder what would happen if we used the optimal *n* and *m* values from the simulation for the actual empirical AAII data. This should remove some data mining bias.

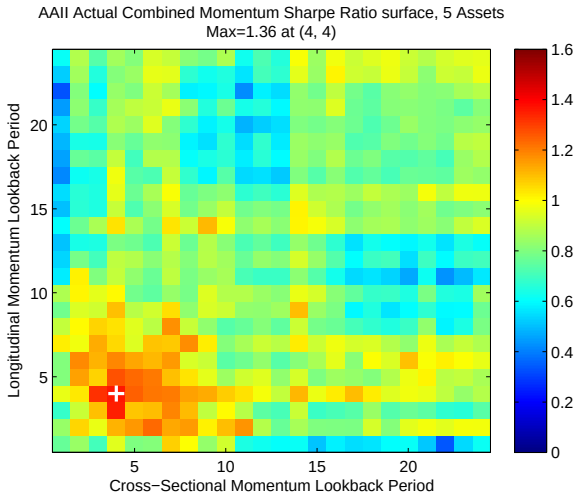
We get the orange line in the chart on the left. We only did this for curiosity – there is no guarantee that the optimal values from the simulation should be optimal for the AAII data going forward. But it still gives a rough idea of the amount of data mining bias that may be present in the AAII backtest. And it demonstrates some of the types of testing we can do in sensitivity analysis where we test how robust the findings are to changes in the parameters used for the simulation.

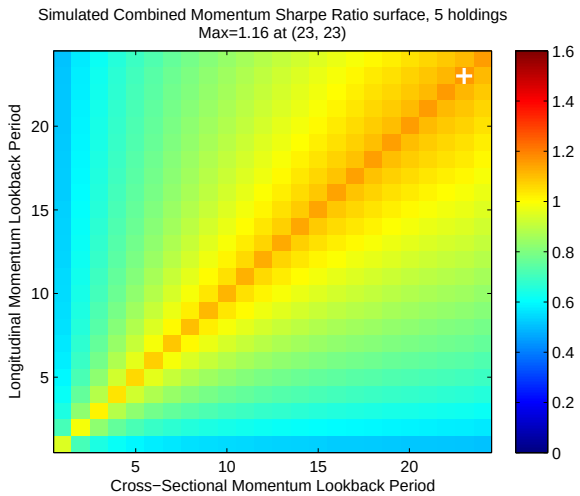
What of the optimal *n* and *m* values themselves? Can we learn anything about them from the simulation. The answer is "definitely."

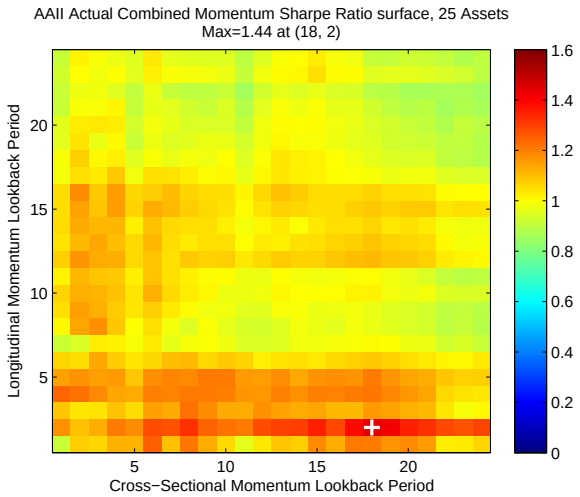
The AAII data for each number of asset values gives us a surface where we vary *n* and *m* each from 1 to 25 and plot the resulting Sharpe Ratio. We show samples of these surfaces below for number of assets 1, 5, and 25 in the left hand chart of each pair. We also show on the right the corresponding charts from the simulation.

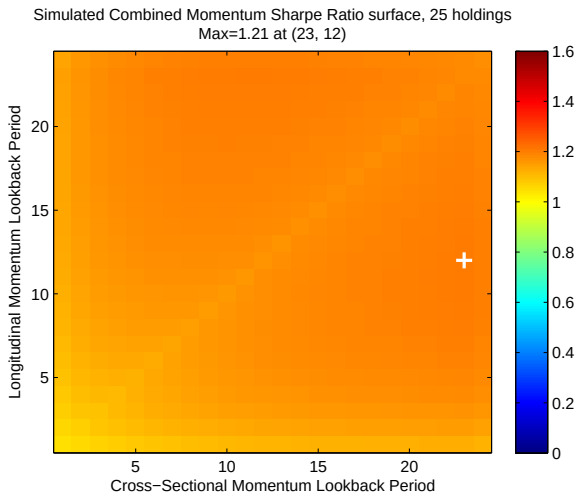












We see the same tendencies in both series of charts. For small numbers of assets the optimum seems to be $n = m$ whereas for larger numbers the optimum moves to the lower right corner. This might seem reasonable – when we have to pick out more assets we have to look at more data to be able to distinguish them relatively. This is certainly a topic to investigate further.

In the meantime, both charts tell us the same story – use $n = m$. But they differ on the optimal sizes of n and m. The actual empirical AAII data likes small values – about 5 or 6 months – and the simulations like larger values – around 24 months. We wonder why. Maybe the actual data is dominated by the *Piotroski* effect which encourages short term trading. Maybe the simulation doesn't match the AAII data well enough. More on this matching problem in the next section.

Having determined the effectiveness of momentum strategies we now ask – can we improve on them? Comparing returns over a lookback period is an inherently noisy process – returns have a huge amount of noise in them. Perhaps we can smooth the returns using regression or the Kalman Filter before doing the comparison. Would a *robust* regression be even better? Should we include quadratic terms in the regression (Li-Wen and Hsin-Yi, 2013 say so)? Maybe we can use the residuals from a regression (see Blitz et al 2011 for a discussion of residual momentum). du Plessis (2013) splits our cross sectional and longitudinal momentums into six different varieties (three for each type) which gives six single plus 36 combined algorithms to test. More strategies abound in the literature.

Indeed, some of these strategies do improve the backtested performance on the actual AAII data. But is the performance genuine? Is this an extreme example of data mining? Only simulation can remove the noise and reveal the true winner.

13. The Problem with Simulation

There is an old parable about a drunk who has lost his keys in the night and is searching for them under a streetlight. A policeman helps him search but after a few minutes to no avail he asks the drunk if he is sure he lost them there, and the drunk replies, no, that he lost them in the park. The policeman asks why he is searching there, and the drunk replies, "this is where the light is." This is called the streetlight effect (Freedman 2010).

A similar effect applies here. We seek the solution to a market problem by leaving the market, creating a new problem by simulation, and seeking the answer there.

This is the weakness of the simulation method. The answer in the simulation may not be the answer in the market. If the optimal smoothing parameter is 40 in the simulation is the optimal value in the market going to be 40?

We suggest no, not exactly. The simulated answer is not to be taken definitively even though it might be calculated to 10 decimal digits. But it *is* indicative. Prior to the simulations we had intuitively used the parameter value 6 for method *smooth2a*. That valued seemed to give appealing visual smoothes.

But since doing the simulations we have switched to using the value 40. That seems by eye to smooth out too many of the nice peaks and troughs. However, we trust the simulations more than we trust our untrained eyes. We don't expect to be around for another 500 years to find out if we were right to switch. But 40 is evidence-based and our value of 6 wasn't. So even though the evidence is not definitive we feel that instead of being totally in the dark the simulations shed useful light on the problem.

To some extent we have replaced one challenge – that of insufficient data – with another challenge – that of modeling a market. The data problem cannot be fixed with time – we can only collect new data at the rate of 12 months' worth every year but we need centuries worth, not years. The modeling problem is being solved at an exponential rate – computer power is still increasing exponentially (Moore's Law) and *Data Science* is a brand new exponentially growing evolution of *Statistics* started over just the last few years to handle the vast quantities of data becoming available from space and from the global internet. New algorithms in this field are being published every week. Most are implemented in code that is open source.

The best is yet to come.

14. References

Aronson, David R. (2007). *Evidence-based technical analysis: applying the scientific method and statistical inference to trading signals* (Vol. 274). John Wiley and Sons, New Jersey.

Badyal, D. K., Modgill, V., & Kaur, J. (2009). "Computer simulation models are implementable as replacements for animal experiments." *Altern Lab Anim*, *37*, 191-5.

Bailey, D. H., Borwein, J. M., de Prado, M. L., & Zhu, Q. (2013). Pseudo mathematics and financial charlatanism: the effects of backtest overfitting on out-of-sample performance. *Submitted Notices of the AMS, September*.

Blitz, D., Huij, J., & Martens, M. (2011). "Residual momentum." *Journal of Empirical Finance*, *18*(3), 506-521.

Chen, Li-Wen and Yu, Hsin-Yi, “Investor Attention, Visual Price Pattern, and Momentum Investing,” (2013). Available at SSRN: http://ssrn.com/abstract=2292895 or http://dx.doi.org/10.2139/ssrn.2292895

Credit Suisse (2014) Global Investment Returns Yearbook

Dyson, Freeman, “A meeting with Enrico Fermi,” *Nature* 427 (22 January 2004) p. 297

Freedman, David H. (August 1, 2010). "The Streetlight Effect". *Discover magazine*.

Garcia D, “Robust smoothing of gridded data in one and higher dimensions with missing values.” *Computational Statistics & Data Analysis*, 2010;54:1167-1178.

Horst, U., & Wenzelburger, J. (2008). On non-ergodic asset prices. *Economic Theory*, *34*(2), 207-234.

Jegadeesh, Narasimhan and Sheridan Titman (1993), “Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency,” Journal of Finance 48(1), 65-91

Lopez de Prado, Marcos, What to Look for in a Backtest (2013). Available at SSRN: http://ssrn.com/abstract=2308682 or http://dx.doi.org/10.2139/ssrn.2308682

Meucci, A. (2009). *Risk and asset allocation*. Springer.

Nadaraya, E. A. (1964). “On Estimating Regression.” *Theory of Probability and its Applications* **9** (1): 141–2. doi:10.1137/1109020

Petris, G., Petrone, S., & Campagnoli, P. (2009). *Dynamic linear models with R*. Springer.

du Plessis, Johan (2013), *Demystifying momentum*, Masters thesis, Korteweg-de Vries Institute for Mathematics

R Development Core Team (2012). R: A language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Watson, G. S. (1964). “Smooth regression analysis.” *Sankhyā: The Indian Journal of Statistics, Series A* **26** (4): 359–372.

Wishart, D. S., Knox, C., Guo, A. C., Shrivastava, S., Hassanali, M., Stothard, P., ... & Woolsey, J. (2006). “DrugBank: a comprehensive resource for in silico drug discovery and exploration.” *Nucleic acids research*, *34* (suppl 1), D668-D672.