Tutorial on the backtest overfitting tool

Document version: 1.0, March 2015

Contents

1 Introduction 1

2 Running options 1

3 How the tool works 2

4 Credits 3

1 Introduction

Backtest overfitting refers to a situation in which a model targets a specific behavior rather than a general one; when certain parameters are optimized to maximize the performance of a backtest, they may not work in the future.

In finance, "backtest overfitting" means using historical market data (i.e., "backtest") to develop an investment strategy, where too many variations of the strategy are tried, relative to the amount of data available. Overfit strategies typically work well when tested against the historical data, but then give disappointing performance when fielded in practice. Backtest overfitting appears to be quite widespread in the financial field, and it is hard to detect, since the number of variations attempted when developing a strategy is rarely disclosed to the users of the strategy.

For example, as we show in the paper (http://www.ams.org/notices/201405/rnoti-p458.pdf), if only five years of daily stock market data are available as a backtest, then no more than 45 variations of a strategy should be tried on this data, or the resulting strategy is likely to be overfit (in the specific sense that the strategy's Sharpe Ratio, a standard measure of investment performance, is likely to be 1.0 or greater).

If you create or invest in a systematic investment strategy (or in an exchange traded fund based on such a strategy), understanding the degree to which the strategy is overfit can help avoid disappointment and financial losses. The tool below allows one to see how easy it is to overfit an investment system, and how this can impact the financial bottom-line performance.

The tool employs a simplified version of the process many financial analysts use to create trading strategies, namely to use a computer program to find the "best" strategy based on historical stock market data, by adjusting variables such as the holding period and the stop loss level. If care is not taken to avoid backtest overfitting, such strategies may look great on paper (based on tests using historical stock market data), but then give rather disappointing results when actually deployed.

2 Running options

The following four options are available over 7 parameters:

1. Maximum holding period: the number of days that a stock can be held before it is sold.
2. Maximum stop loss: the percentage of the profit that can be lost before the stock is sold.
3. Sample length: the number of days simulated (the daily closing price can be used as the sample).
4. Standard deviation: the standard deviation of random numbers used to generate daily prices.

5. Seed: seed for the pseudo-random numbers used for IS data.
6. Entry day: the day that one enters into the market in each trading month.
7. Side: the trading strategy, either *long*, which is to make profits when stocks are rising, or *short*, which is to make profits when stocks are falling.

Options:

1. Re-building the example which is available online. Parameters are:
   (a) Maximum holding period: 20
   (b) Maximum stop loss: 23
   (c) Sample length: 1152
   (d) Standard deviation: 2
   (e) Seed: 308
2. Generating parameters randomly, where the default ranges for parameters' values are taken; every parameter value is generated randomly out of the associated range.
3. Entering parameter values by the user; only the first five:
   (a) Maximum holding period;
   (b) Maximum stop loss;
   (c) Sample length;
   (d) Standard deviation;
   (e) Seed;
4. Implementing real-world stock market data. The real-word data is taken from the S&P500 index from 1962 to 2014.

   The data was divided into two sets where the first set, IS data is for optimizing the parameters' values, and the second set, OOS data are to evaluate the optimal strategy.

3 How the tool works

This online tool performs the following four steps.

1. **Pseudorandom/real-world stock market prices.** Constructs a time series simulating "In Sample" (backtest) daily prices of a stock, where the daily price variation is given by the default pseudorandom number generator gauss from the Python package (or real-world stock market data is used). This Gaussian distribution has mean zero and a user-specified standard deviation. The random numbers are controlled by the user through three parameters: the length of the time series (number of days), the standard deviation (positive integers only), and the seed for the random generator (positive integers only).
2. **Optimal strategy.** Develops an "optimal" trading strategy based on the given data, by successively adjusting the following four variables (tries all combinations of the four parameters):
   (a) Stop loss: the percentage of profit that can be lost before the stock is sold. This tool only tries integer percentages up till the maximum which given by the user.
   (b) Holding period: the time (duration) that stock can be held before it is sold. It is given in whole number of trading days. This tool tries all integer values less or equal to the maximum given by the user.
   (c) Entry day: the day that one enters the market in each trading month, which is assumed to be 22 days long. All 22 choices are tried by this tool.
   (d) Side: the trading strategy that is employed, either long (profits are to be made when stocks are rising) or short (profits are to be made when stocks are falling). Both options are tried by this tool.

If a change of these variables yields a higher Sharpe Ratio than the previous permutations, then a new strategy is output. The program then continues to try different permutations until it has tried every possibility.

Note that among the above four parameters, the user specifies the maximum values for the first two, the maximum stop loss percentage and the maximum number of days to hold on to the stock.

3. **Evaluation of the optimal strategy on OOS data.** A second pseudorandom simulated stock market time series ("Out of Sample" data) is generated (or real-world stock market data is used), and the "optimal" strategy generated above is then applied to this second time series. This time series simply continues from "In Sample" time series, with the same pseudorandom number generator for the same number of simulated days. The two time series are shifted up by the same amount so that the minimum values in both cases are above 0.01.

4. **Visualization.** The program outputs, on the result page (allow up to two minutes for the result page to be generated), three graphs. The first two graphs show results on "In Sample" and "Out of Sample" date: the results on "In Sample" data is on the left and "Out of Sample" on the right. A "movie" showing the progression of the generation of the optimal strategy on "In Sample" (backtest) data can be played by clicking on the graph on the left (for the "In Sample" data).

   In this two graphs on the result page, the green line is the underlying time series, and the blue line shows the performance of the strategy. The "SR" notation in these graphs denotes the Sharpe Ratio. In most runs, the Sharpe Ratio of the right graph (i.e., the final strategy on Out of Sample data) is either negative or much lower than the Sharpe Ratio of the final left graph (i.e., the final strategy on In Sample data), indicating that the strategy has been overfit on the In Sample (backtest) data.

   The third graph shows the value of the Deflated Sharpe Ratio (DSR) over changes in the value of the "Number of trials", as blue line. The same for a benchmark setting (Skewness: -3 and Kurtosis: 10) has been shown as red, only to give an idea of different behavior.

4 Credits

This web page and program were constructed by Stephanie Ger, Amir Salehipour, Alex Sim, John Wu and David H. Bailey, based on an earlier Python program developed by Marcos Lopez De Prado. This program in turn is based on the following research paper:

- David H. Bailey, Jonathan M. Borwein, Marcos Lopez de Prado, and Qiji Jim Zhu, Pseudo-Mathematics and Financial Charlatanism: The Effects of Backtest Overfitting on Out-of-Sample Performance, Notices of American Mathematical Society , May 2014, pg. 458-471.

We gratefully acknowledge the helpful comments and suggestions from colleagues and friends in shaping this web site. In particular, the suggestions from Mr. David Witkin of StatisTrade, Bin Dong of LBNL, and Beytullah Yildiz of Turkey were extensive and very helpful in improving readability of the web pages. Thanks!