

OPTIMIZATION DONE RIGHT!

HOW TO AVOID OVER
OPTIMIZING YOUR
STRATEGIES



ACCESSING TWO TIMEFRAMES IN EASYLANGUAGE

Don't Be Fooled By The Charlatans!

WHO IS JEFF SWANSON

I used to lose a lot of money in the markets!

I was an emotional wreck. For years as a discretionary trader I never made a dime but would continually lose money. During those times I really felt like a loser as I could not figure out this game called trading. It dawned on me that I was just another average trader losing money and losing my faith that I could ever trade and make money.

This all changed when I discovered automated trading.

Moving to a quantitative trading and automating my trading changed how I traded forever! Suddenly trading made sense as I could backtest trading ideas before putting my money at risk. I discovered what worked and what didn't before putting money on the line. The emotional baggage of pulling the trigger to open a new trade disappeared as the computer made that decision for me. Likewise the computer made all trade management decisions from how many contracts to buy and when to exit. It was awesome and I never looked back at my old ways!

There are several important aspects you need to master to become successful at system trader. On our website we highlight these aspects and go deeper into the knowledge and skills you need to know! So, I hope you take advantage of the website and this free guide. There is gold to be found here!

If you have any questions about system trading, you're invited to ask us at: questions@easylanguagemastery.com. We've dedicated to educating the retail trader just like you. We'll be happy to help in every way we can.



Our Mission

Helping You Become A Successful Trader



ACCESSING TWO TIMEFRAMES IN EASYLANGUAGE

OVERVIEW

EasyLanguage Mastery is dedicated to helping you become a successful system trader. We specialize in helping EasyLanguage users in building and trading automated trading systems.

- (1) Discover profitable market edges
- (2) Learn how to avoid over fitting your systems so they work on the live market
- (3) Helping to ensure your system work on the live market

If you're new to EasyLanguage Mastery, visit our home page and you'll find more information about the topics that interest you.

We also share exclusive tips over on our various social media channels.

Join Our Free EasyLanguage Mastery Community!
Chat, share and learn about EasyLanguage in our free FaceBook group.



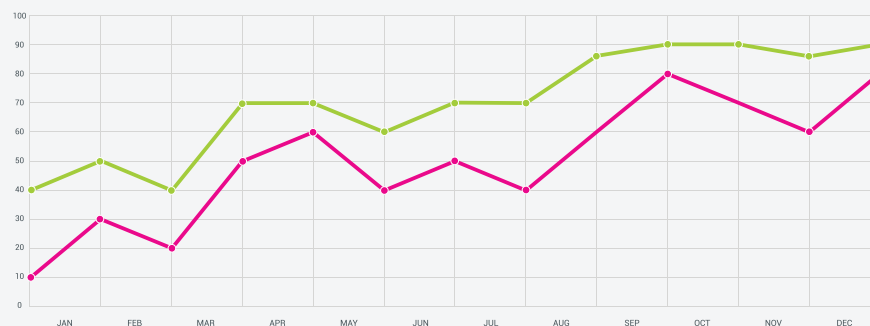
The Problem

FINDING PATTERNS WITHIN THE NOISE

As you've well known, curve fitting is a danger that all system developers must constantly be aware of. Curve fitting not only destroys your trading system, but can give you false hope that your system will work on the live market which results in you losing your hard earned money. Curve fitting often hits novice system developers without them even knowing about it. This can be particularly confusing and disheartening. But it can creep into the development process of professional developers. It's your job to be aware of curve fitting and how to avoid it.

I wrote this guide to help you better understand how to avoid curve fitting in your system development endeavors. Now, with this information you can make an up to date, intelligent decision.

Curve fitting may not be the best term to use here. Curve fitting from a mathematical standpoint is attempting to fit a line or curve through a series of data points. That's not what we are doing with a trading system. When it comes to trading, we are attempting to optimize a trading system's performance. Often this is a key performance metric, such as net profit. We do this by attempting to optimize various parameters in an attempt to maximize profits. Of course, one single metric is not all we look at when attempting to optimize a system. We take into account many different performance metrics such as profit factor, drawdown, and average profit per trade, to name a few, to help us gauge if a particular system is tradable or not.



When it comes to making a profitable trading system we depend upon finding market edges we can exploit. All successful trading rely on recurring patterns found in the historical market data and exploiting those patterns to make a profit.

This is true for discretionary traders as well as system traders. However, much of the historical market movement is noise or random movement. So, the problem arises when we over-optimize our system to the historical data and is unknown to us, our system begins to key-in to patterns within the noise instead of exploiting a true market edge. Finding a pattern in the noise can produce great looking equity curves on historical data, but on out-of-sample data the equity curve can break down rather quickly.

In summary, we want a trading system that exploits a true market edge and avoids keying off patterns in the market noise. The idea of separating a true market edge from the noise is what this game is all about.

So how does over-optimizing occur? It's often accomplished unknowingly as a system developer continues to tweak the trading system to obtain better results over the historical market data.

Over-optimizing can often occur when:

- Using out-of-sample data to make corrections to a system.
- Selecting highly optimized parameters that give you the best results.
- Having too many parameters.
- Performing what I call, Conjunction Optimization.



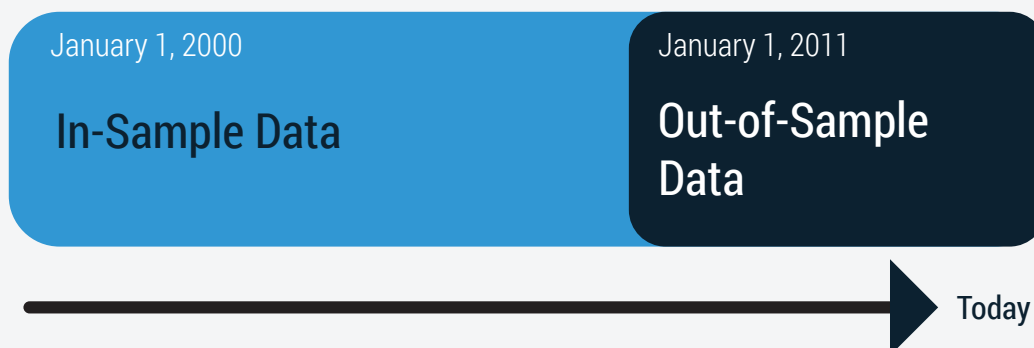
STEP 1

Proper Use Of In-Sample And Out-of-Sample Data

The first step when developing a trading system is to divide your historical data over two segments.

1. In-sample segment
2. Out-of-sample segment

This basic step is required for all system development. The in-sample segment should consist of the bulk of the historical data. Let's say about 80% of your historical data should be dedicated to this segment. The in-sample segment is the historical data used when developing your system. It's where all your development decisions are made. During your development process you never apply your work-in-progress trading system to the out-of-sample segment. This segment is reserved for testing your final system to see if it can hold up on historical data it has never "seen".



Once your development is done one of the first things you will do is now test the performance on the out-of-sample segment to see how it performs. Ideally, the system should perform well as the equity curve advances.

Here's what not to do!

Do not take the results of the out-of-sample data and tweak your trading system in an attempt to improve it. Doing this nullifies the concept of the out-of-sample segment as being out-of-sample. The idea of the out-of-sample segment is to test your strategy on "unseen" data. If you go back and start modifying your trading system you have not tainted the out-of-sample data. It's now in-sample data as you are taking knowledge based upon the system performance and tweaking the system to make it better!



When your system is applied to the out-of-sample data it either passes or fails. While it's true that mediocre results on the out-of-sample segment may be OK, don't start tweaking the parameters.

Other Important Aspects

Your in-sample data segment should cover several different market regimes. That is, both bull and bear markets should be present within your in-sample data. It's important that when developing your system it is performed over different market regimes to help ensure that your strategy will work into the future.

Your in-sample data segment should have a minimum of 60 trades - ideally, 100 or more. In general, the more trades you have the better.

Likewise your out-of-sample data segment should also have a minimum of 60 trades - ideally, 100 or more. It also should cover both bull and bear markets. But I know this is not always possible.

IMPORTANT



STEP 2

How To Optimize Correctly

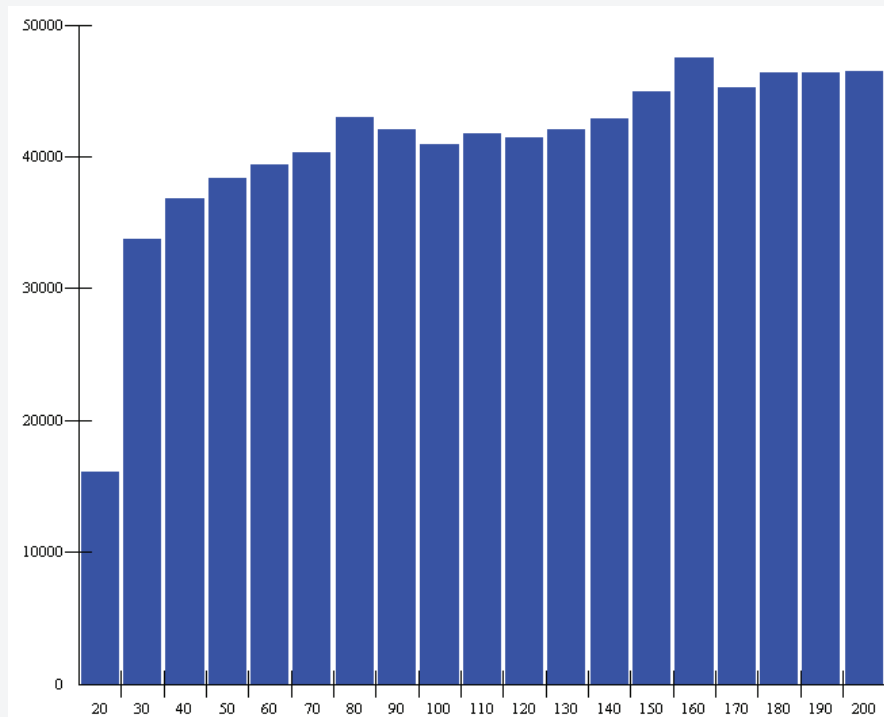
Optimizing particular strategy inputs, such as a lookback period of a moving average crossover system, can be a strong source of over-optimizing your system. However, you can significantly reduce the risk by following a few key points

1. **Don't pick the best.** When optimizing a parameter you don't want to pick the best value.
2. **Look for a stable region.**
3. Ideally, you want to **find a cluster or range of values** where your system performs well.
4. If a default value looks optimized or not robust, locate a stable region and **change the default parameter** to the middle value.
5. **Abandon** if there are no stable regions. In this case often the optimization will look choppy with drastic changes.

The key to optimizing parameters is something I call robustness in the given parameter you're optimizing.

Let's say we are attempting to optimize a look-back period for a moving average. Let's also say the default value is currently 50. When optimizing this value I want to look for two things. First, if the default value is optimized. Second, if other nearby values produce similar results. In short, I would like to test the robustness of the look-back period. For example, if we change the 50-day default value to 49 or 51, I don't want to see the system's equity curve drastically change. Furthermore, the system should not radically change if I use 47 or 52. The neighboring values around 50 should still produce positive results and in fact, produce similar results. Actually it would be ideal to see the system remain profitable over a wide range of values.

Below is a bar graph of a look-back period optimization covering the range between 20-200. The x-axis contains the look-back period while the y-axis contains the trading system's total P&L. Remember, our default value was 50 and we want to optimize it.



First, the default value of 50 is far from an optimal value. It appears that a wide variety of look-back values produce positive results. Since we don't want to pick an optimal value, how do we pick a value?

Notice the values starting from 20 climb in net profit as we reach 80 then level off until they start to climb around 150. It's reasonable to pick a value within the range of 80-150. Often I will pick the midpoint. You can calculate this by adding the values (80+150), and divide it by 2. In this case, the value of 115 would be our new look-back period.

You could also make the case there as a stable region between 150-200. It might be worth extending the look-back study to see values beyond 200. Picking a midpoint value within this region is most likely fine as well. It would be worth testing and comparing the various system performance metrics to see if there is a reason to pick one stable region vs the other. This might simply come down to a personal preference such as preferring fewer trades. I won't explore the other regions within this guide as I think you get the picture.

When optimizing a value, you don't simply want to pick the best looking value. Find a region where the values appear stable. That is, an area with little variation. Once you find that region, pick a value within the center of that region. Doing this will help reduce overoptimizing your trading system.

If you can't find a region that is stable, and the optimization results vary wildly, then you're likely have a trading rule or strategy which will not work well on the live market. Thus, it may be best to abandon the rules or strategy.

STEP 3

Keep It Simple!



Keep in mind that simple is often better. Keep the number of parameters and trading rules to as few as possible to accomplish the task. Introducing more trading rules increases the likelihood of overoptimizing the system to the historical data. If you introduce a trading rule to a system, it must have a significant positive impact on the trading rules in order to justify keeping it. If the results are deemed mediocre, then throw out the rule.

For example, let's say a single rule introduced to your system reduced the number of trades by 1/3 or even 1/2. At the same time the profit factor increases from 1.07 to 1.50 and the average dollars per winning trade increases from \$20 to \$60. To me, this is significant because we reduced the number of trades by removing unprofitable trades. This may be an exaggerated example, but it could happen. Generating fewer trades and creating more dollars per trade is usually always a great idea even if the system is making less net profit.

Keep an eye on the drawdown as well. Maybe a rule does not significantly improve the net profit but does significantly reduce drawdown. This may be significant and worth keeping. It is true that "significant" can be subjective. That's fine because each of us will have different tolerances to risk, drawdown, and what we consider an acceptable trading system. However, keep in mind if the system does not benefit much from the rule, you are better to leave it out. A trading system with fewer rules is generally considered more robust. So when in doubt, leave it out!



STEP 4

The Independent Testing Secret

Following this important step can go a long way in preventing you from developing a system that is overly optimized to the historical data. When developing a trading system it's common to apply different filters and stops. For example, let's say you have a simple moving average crossover system. You then wish to test a trailing stop and a trend-strength indicator to filter trades. So we have the following items to test:

- Look-back period of the moving average crossover
- Threshold value used on the trend filter
- Trailing stop distance

How do you pick values for each of these indicators? How do you optimize your system? We already know not to pick the best optimization but when you have multiple items to optimize, how do you go about it? In my opinion you should attempt to perform independent testing whenever possible. What does that mean? Each new rule that we test must be done independent of the other rules. While it's not always possible to completely do, it should be done whenever possible.

Let's say we would like to test a trailing stop. After adding the trailing stop to our baseline system we determine if it improves the performance significantly. We record our results and move on to test another rule.

Next we choose to test a profit target. To test our profit target rule we must revert back to the baseline system. That is, we do not add our profit target rule to the strategy code with the trailing stop rule. Our profit target rule must be applied to the baseline system. Each new rule that we test must be done independent of the other rules. This is done to avoid overoptimization and will help maintain a robust system.

You should also test different entry methods. A limit order at the extreme of the last three bars vs market order at the open of the next bar, for example. Follow the same guidelines when testing different filters and exit methods.

CONGRATULATIONS!

Over-optimization is always a danger, but if you follow the simple guidelines provided within this eBook you will be much more effective in reducing the chance of over-optimizing your trading systems. I use these very techniques all the time. Give them a try!

For more information visit our website, **EasyLanguage Mastery** for content rich articles on all aspects of building profitable trading systems with EasyLanguage.

