Developing & Backtesting Systematic Trading Strategies

Brian G. Peterson

June 16, 2015

Analysts and portfolio managers face many challenges in developing new systematic trading systems. This paper provides a detailed, repeatable process to aid in evaluating new ideas, developing those ideas into testable hypotheses, measuring results in comparable ways, and avoiding and measuring the ever-present risks of over-fitting.[1]

Constraints, Benchmarks, and Objectives[2]

It is important to understand what you are trying to achieve *before* you set out to achieve it.

Without a set of clearly defined goals there is great potential to accept a strategy or backtest that is really incompatible with the (previously unstated) business objectives and constraints. Worse, it can lead to adjusting your goal to try to follow the backtest, which can culminate in all sorts of bad decision making, and also increases the probability of erroneously accepting an overfitted backtest.

Understanding your business constraints (and advantages)

Think about the constraints, costs, and advantages that you are under first, before formulating a hypothesis, or writing a strategy specification. Many strategists skip this step. Don't. It is a waste of time and energy to design a strategy for a constraint set that is not available to you.

Some examples include:

- execution platform
- access and proximity to markets
- products you can trade
- data availability and management
- business structure
- capital available
- fees/rebates

Choosing a benchmark

The choice of what benchmarks to measure yourself against provides both opportunities for better understanding your performance and

[1] *Back-testing. I hate it — it's just optimizing over history. You never see a bad back-test. Ever. In any strategy.* - Josh Diedesch(2014)

[2] *Essentially, all models are wrong, but some are useful.* - George Box(1987)

risks of benchmark chasing and overfitting. Mutual Fund managers have it easy, or hard, because of the "Morningstar" categorization of styles and benchmarks. Mutual fund managers have benchmarks clearly defined, and can be easily measured against what are often exchange traded products or well known indices.

A good suite of benchmarks for your strategy should reflect your business objectives and constraints, as well as help you measure performance over time.

What constitutes a good benchmark for a trading strategy?

There are several categories of potential benchmarks:

1. archetypal strategies

   In many cases, you can use a well-known archetypal strategy as a benchmark. For example, a momentum strategy could use an MA-cross strategy as a benchmark.
   If you are comparing two or more candidate indicators or signal processes, you can also use one as a benchmark for the other (more on this later).

2. alternative indices

   EDHEC, Barclays, HFR, and others offer alternative investment style indices. One of these may be a good fit as a benchmark for your strategy. Be aware that they tend to have a downward bias imposed by averaging, or are even averages of averages. So 'beating the benchmark' in your backtest would be a bare minimum first step if using one of these as your benchmark. As an example, many CTA's are trend followers, so a momentum strategy in commodities may often be fairly compared to a CTA index.

3. custom tracking portfolios

   In some ways an extension of (1.) above, you can create your own benchmark by creating a custom tracking portfolio. As the most common example, a cap-weighted index is really a strategy archetype. The tracking portfolio for a capitalization-weighted index invests in a basket of securities using market capitalization as the weights for the portfolio. This portfolio is then rebalanced on the schedule defined in the index specification (typically quarterly or annually). Components may be added or removed following some rule at these rebalancing periods, or on some other abnormal event such as a bankruptcy. Other custom tracking portfolios or synthetic strategies may also be appropriate for measuring your strategy against, depending on what edge(s) the strategy hopes to capture, and from what universe of investible products.

4. market observables

   Market observable benchmarks are perhaps the simplest available benchmarks, and can tie in well to business objectives and to optimization.

   Some easily observable benchmarks in this category:

   - $/day, %/day
   - $/contract
   - % daily volume
   - % open interest

measuring performance against your benchmark

Good benchmarks help to provide insight into the drivers of strategy returns.

Choosing more than one benchmark can help to guard against targeting the benchmark (a form of overfitting). Identifying specific limitations of your benchmarks, for example:

- lack of goodness of fit,
- a different investible universe,
- lack of stationarity,
- or specific representative features of the benchmark

can also assist in refining the choice of benchmarks and avoiding taking them too seriously.

No matter what benchmarks you pick, if you are going to share these benchmarks with outside investors or other members of your team, you need to be careful that you understand and disclose the limitations of the benchmark(s) that you've chosen.

Failure to disclose known divergences from a benchmark you will be measured against may look like a limitation rather than a feature, so be sure to explain why differences should exist.

You *will* be measured by others against these benchmarks for performance, style drift, etc. so failure to disclose known limitations of your benchmark(s) will waste everyone's time.

Large bodies of research have been built up over time about how to measure performance against such benchmarks. *TrackingError*, *SharpeRatio*'s, *FactorAnalytics*, and more are well represented in **R**.

Choosing an Objective[3]

In strategy creation, it is very dangerous to start without a clear objective.

[3] *When measuring results against objectives, start by making sure the objectives are correct.* - Ben Horowitz (2014)

Market observable benchmarks (see above) may form the core of the first objectives you use to evaluate strategy ideas.

Your business objective states the types of returns you require for your capital, your tail risk objectives, the amount of leverage you intend to or are willing to use, and your drawdown constraints (which are closely related to the leverage you intend to employ).

Some of the constraints on your business objective may be dictated by the constraints your business structure has (see above).

For example: - Leverage constraints generally have a hard limit imposed by the entity you are using to access the market, whether that is a broker/dealer, a 106.J membership, a leased seat, a clearing member, or a customer relationship.

- Drawdown constraints have hard limits dictated by the leverage you intend to employ: 10:1 leverage imposes a 10% hard drawdown constraint, 4:1 imposes a 25% drawdown constraint, and so on. - Often, there will also be certain return objectives below which a strategy is not worth doing.

Ideally, the business objectives for the strategy will be specified as ranges, with minimum acceptable, desired target, and maximum acceptable or plausible targets.

Once you have these objectives described, you need to formulate them so that they may be easily used to programmatically evaluate your strategy.[4]

Formulate your business objectives in the same form commonly used in portfolio optimization:

maximize some measure of return
subject to
minimizing one or more measures of risk

The most commonly used objective of this type is the Sharpe Ratio which is most often calculated as mean return over volatility.

Many other modified Sharpe-style ratios also exist:

- Information Ratio: annualized return over annualized risk
- Profit factor: gross profits divided by gross losses
- Sortino Ratio: (annualized return - risk free) over downside volatility
- Calmar Ratio: annualized return over max drawdown
- return over expected tail loss

All of these ratios have the properties that you want to maximize them in optimization, and that they are roughly comparable across different trading systems.

Additional properties of the trading system that you may target when evaluating strategy performance but which we would not put

[4] **FIXME** *we need a version of constrained_objective() from PortA in quantstrat*

into the category of *business requirements* include:

- correlation to perfect profit
- slope of the equity curve
- linearity of the equity curve
- trade statistics etc.

A strategy conceptualized as a diversifier for an existing suite may well have different business objectives from a strategy designed to enter a new market or replicate published research or take advantage of a transient market opportunity. As such, it is further important to be aware that business objectives are likely to differ from one strategy to another even inside the same organization or on the same team.

Understanding your constraints, benchmarks, and objectives prior to embarking on hypothesis generation or strategy creation and evaluation will anchor your goals. Defining these in advance will allow you to quickly reject ideas that are incompatible with your goals and requirements. Proceeding without understanding the milieu in which the proposed strategy will operate risks overfitting and sloppy analysis, and should be avoided whenever possible.

Hypothesis Driven Development[5]

Strategy creation is expensive in time (for research) and money (for implementation into a live trading environment).

To maximize return on that investment, we follow a hypothesis driven approach that allows us to confirm ideas quickly, reject failures cheaply, and avoid many of the dangers of overfitting.

Ideas generated via a brainstorming session are a useful starting point, but they are typically not testable on their own.

creating good (testable) hypotheses

To create a testable idea (a hypothesis), we need to:

- formulate the idea as a declarative conjecture
- make sure the conjecture is predictive
- define the expected outcome of that conjecture
- describe means of verifying (testing) the expected outcome

Many ideas will fail the process at this point. The idea will be intriguing or interesting, but not able to be formulated as a conjecture.

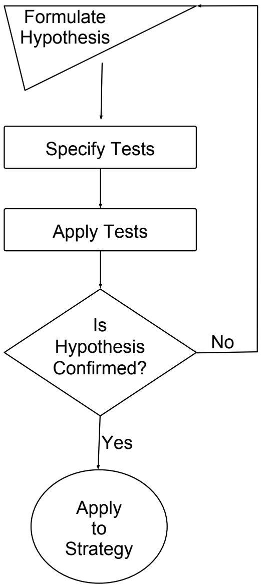
Or, once formatted as a conjecture, you won't be able to formulate an expected outcome. Often, the conjecture is a statement about the

[5] *Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise.* - John Tukey (1962) p. 13

nature of markets. This may be an interesting, and even verifiable point, but not be a *prediction* about market behavior. So a good conjecture also makes a prediction, or an observation that amounts to a prediction. The prediction must specify either a cause, or an observable state that precedes the predicted outcome.

Finally, a good hypothesis must specify its own test(s). It should describe how the hypothesis will be verified. A non-testable "working hypothesis" is a starting point for further investigation, but the goal has to be to specify tests. In some cases, it is sufficient to describe a test of the prediction. We'll cover specific tests in depth later in this document. In statistical terms, we need the hypothesis to have a testable dependent variable. In other cases it may be necessary to describe a statistical test to differentiate the hypothesized prediction from other measurable effects, or from noise, or factor interactions. Specified tests should identify measurable things that can help identify the (spurious) appearance of a valid hypothesis.

A good/complete hypothesis statement includes:



- what is being analyzed (the subject),
- the dependent variable(s) (the output/result/prediction)
- the independent variables (inputs into the model)
- the anticipated possible outcomes, including direction or comparison
- addresses *how you will validate or refute each hypothesis*

Most strategy ideas will be rejected during hypothesis creation and testing.

Many ideas, thoughts, and observations about the state of the markets lack sufficient structure to become a testable hypothesis. While it is possible that these ideas are indeed true, unless you can quantify the idea into a (dis)provable conjecture, it can't be used to create a quantitative strategy with any degree of confidence. Far from being a negative thing, this is a good use of your time, and guards against prematurely moving on with a fundamentally questionable or just plain erroneous idea. The faster an untestable idea can be set aside for more verifiable ideas, the better.

Proceeding without a hypothesis risks ruin.[6] Many strategists will still try to skip robust hypotheses in favor of sloppy ones such as "I hypothesize that this strategy idea will make money".

While the value of testing a more rigorous hypothesis should be clear, it may be more difficult to see the risks imposed by having no hypothesis or a sloppy or incomplete hypothesis. An incomplete or otherwise deficient hypothesis at this stage will create a strong desire

[6] *A big computer, a complex algorithm and a long time does not equal science. - Robert Gentleman*

to "refine" the hypothesis later by adding new explanatory statements. This is called an "ad hoc hypothesis", where new hypotheses are piled on top of old to better "explain" observation. (Carroll (2011) , p. 7 or online; see also "rule burden", below) Engaging in the creation of ad hoc hypotheses risks ruin, as the in sample (IS) explanatory power of the model will seem to go up, while the out of sample (OOS) power goes down. It also invites going over the data multiple times while the model is "refined", introducing progressively more data snooping bias.

Defining the strategy

A trading strategy or investment approach is more than just a testable hypothesis.

After creating and confirming a hypothesis, you need to specify the strategy. The **R** package *quantstrat* formalizes the strategy structure into *filters*, *indicators*, *signals*, and *rules*.

Filters help to select the instruments to trade.

They may be part of the formulated hypothesis, or they may be market characteristics that allow the rest of the strategy to trade better.

In fundamental equity investing, some strategies consist only of filters.

For example, the StarMine package that was bought by Thomson Reuters defines quantitative stock screens based on technicals or fundamentals.[7] Many analysts will expand or shrink their investible universe based on screens. Lo's Variance Ratio is another measure often used as a filter to turn the strategy on or off for particular instruments (but can also be used as an indicator, since it is time-varying).

Indicators are quantitative values derived from market data.

Examples of indicators include moving averages, RSI, MACD, volatility bands, channels, factor models, and theoretical pricing models. They are calculated in advance on market data (or on other indicators). Indicators have no knowledge of current positions or trades.

One risk to watch out for when defining your indicators is confusing the indicator with the strategy. The indicator is a description of reality, a model that describes some aspect of the market, or a theoretical price. An indicator, on its own, is not a strategy. The strategy

[7] a modern, free alternative may be found at `http://finviz.com/screener.ashx`

depends on interactions with the rest of its components to be fully specified.

Signals describe the interaction between filters, market data, and indicators.

Signal processes describe the desire for an action, but the strategy may choose not to act or may not be actionable at the time. They include 'classic' things like crossovers and thresholds, as well as more complicated interactions between pricing models, other indicators, and other signal processes. It is our experience that signals may often interact with other signals. The final signal is a composite of multiple things. The combination of intermediate signals will increase the likelihood of actually placing or modifying orders based on the final signal.

Like indicators, signals are calculated in advance on market data while doing research or a backtest. In production, they are typically calculated in a streaming fashion.

As stated above, it is important to separate the desire for action from the action itself. The signal happens only in its own context. In very simple strategies, signal and action may be the same, but in most real strategies the separation serves to make clear what analysis can be done to support a decision, and what needs to be postponed until a path-dependent decision is warranted.

Rules make path-dependent actionable decisions.

Rules, in both research/backtesting and in production, are what take input from market data, indicators, and signals, and actually take action. Entry, Exits, Risk, Profit Taking, and Portfolio Rebalancing are all rule processes. Rules are path dependent, meaning that they are aware of the current market state, the current portfolio, all working orders, and any other data available at the time the rule is being evaluated. No action is instantaneous, so rules also have a cost in time.

There must be some lag between observation (indicator or signal), decision (rule), and action (order entry or modification).

strategy specification document

The strategy specification document is a complete description of the strategy. A draft covering all components of the strategy should be completed before any code is written. It describes the business context and objectives, the hypothesis and tests for that hypothesis, and all the components of the strategy itself (filters, indicators, signals and rules). It also needs to describe details such as initial parameter choices and why those choices were made, and any data requirements (tick, bar, L2, external data, etc.).

The strategy specification needs to be written at a level sufficient for the research analyst to write code from. Equally important, it should be written for review by the investment and risk committees.

One of the key reasons for writing the specification before beginning coding, beyond clarity and rigor of thinking, is that it provides another curb against overfitting. Changes to the specification should be deliberate, recorded and indicate that something was incomplete or erroneous in the initial specification. The strategy designer will learn from success and failure along the development path, and the outcomes of these incremental experiments should be clearly reflected in the strategy specification and its revision history.

Specification changes need to be conscious and recorded. Loosely adjusting the specification as you go increases the likelihood of an overfit and invalid test.

Creating the Strategy

Creating the strategy code for backtesting or production is outside the scope of this document. In **R** you'll likely use *quantstrat*, and when transitioning to production, you'll use whatever execution platforms are supported by your organization or team.

Developing and testing the strategy in stages using the tools described in the following sections makes the strategy development process less expensive, more rigorous, and less subject to over-fitting. This is in contrast to the process described by many of the references in this document which focus *only* on evaluating the entire strategy at once, and modifying the whole strategy before doing another analytical trial.

Evaluating the Strategy[8]

How do you evaluate the backtest?

Conceptually, the goal should continue to be to reject or fix failures quickly, before getting to the point of something that risks feeling too expensive to throw away. We want to evaluate the backtest against our hypothesis and business objectives. The earlier in the strategy

[8] *No matter how beautiful your theory, no matter how clever you are or what your name is, if it disagrees with experiment, it's wrong.* - Richard P. Feynman (1965)

framework we can identify and correct problems, the less likely we are to fall victim to common errors in backtesting:

Look Ahead Biases are introduced when an analysis directly uses knowledge of future events. They are most commonly found when using 'corrected' or 'final' numbers in a backtest rather than the correct 'vintage' of data. Other common examples are using the mean of the entire series, or using only surviving securities in an analysis performed after several candidate securities have been delisted. (see Baquero, Ter Horst, and Verbeek 2005) Look ahead bias is relatively easily corrected for in backtests by subsetting the data at each step so that indicators, signals, and rules only have access to the data that would have been contemporaneously available. Guarding against vintage or survivorship problems requires slightly more work, as the analyst must be aware that the data being used for the analysis comes in vintages.

Data Mining Bias arises from testing multiple configurations and parameters of a trading system without carefully controlling for the introduction of bias. It is typically caused by brute force searching of the parameter space without a driving hypothesis for why those parameters would be good candidates.(see Aronson 2006, Chapter 6, pp.287-320)

Correcting for data mining bias is covered in Aronson, and consists mostly in designing the tests such that you do not pollute your data. Walk forward analysis (covered later) is a key component of this, as are adjustments and measurements of data mining effects after they occur.

Data snooping is the process by which knowledge of the data set contaminates your statistical testing (your backtest) because you have already looked at the data. In an innocuous form, it could occur simply because you have market knowledge. (see Smith) Tukey called this "uncomfortable science" because of the bias that it can introduce, but knowledge of how markets work, and what things have worked in the past is an essential part of strategy development. What you need to be cautious of is making changes to the *strategy* to overcome specific deficiencies you see in testing. These types of changes, while sometimes unavoidable and even desirable, increase the probability of overfitting.

Sullivan, Timmermann, and White (1999) and Hansen (2005) provide tests derived from White's Reality Check (which is patented in White 2000), to measure the potential impact of data snooping on the backtested results.

One deficiency in the strategy literature is that it has been moving towards utilizing parameter optimization, machine learning, or statistical classification methods (such as random forests or neural networks) as a silver bullet for strategy development. While a full treatment is beyond the scope of this paper, it is worth mentioning that the model assessment and selection techniques of the statistical classification and machine learning literature are very valid in the context of evaluating trading strategies. (see e.g. Hastie, Tibshirani, and Friedman 2009, chap. 7 and 8) We advocate extensive use of model validation tests, tests for effective number of parameters, and careful use of training and cross validation tests in every part of strategy evaluation. Of particular utility is the application of these techniques to each *component* of the strategy, in turn, rather than or before testing the entire strategy model. (see, e.g. Kuhn and Johnson (2013))

Defining the objectives, hypotheses, and expected outcome(s) of the experiment (backtest) as declared before any strategy code is written or run and evaluating against those goals on an ongoing basis will guard against many of the error types described above by discarding results that are not in line with the stated hypotheses.

Evaluating Each Component of the Strategy[9]

It is important to evaluate each component of the strategy separately. If we wish to evaluate whether our hypotheses about the market are correct, it does not make sense to first build a strategy with many moving parts and meticulously fit it to the data until after all the components have been evaluated for their own "goodness of fit".

The different components of the strategy, from filters, through indicators, signals, and different types of rules, are all trying to express different parts of the strategy's hypothesis and business objectives. Our goal, at every stage, should be to confirm that each individual component of the strategy is working: adding value, improving the prediction, validating the hypothesis, etc. before moving on to the next component.

There are several reasons why it is important to test components separately:

Testing individually guards against overfitting. As described in the prior section, one of the largest risks of overfitting comes from data snooping. Rejecting an indicator, signal process, or other strategy component as quickly in the process as possible guards

[9] *Maintain alertness in each particular instance of particular ways in which our knowledge is incomplete.* - John Tukey (1962) p. 14

against doing too much work fitting a poorly conceived strategy to the data.

Tests can be specific to the technique. In many cases, specific indicators, statistical models, or signal processes will have test methods that are tuned to that technique. These tests will generally have better *power* to detect a specific effect in the data. General tests, such as *p-values* or *t-tests*, may also be valuable, but their interpretation may vary from technique to technique, or they may be inappropriate for certain techniques.

It is more efficient. The most expensive thing an analyst has is time. Building strategies is a long, intensive process. By testing individual components you can reject a badly-formed specification. Reusing components with known positive properties increases chances of success on a new strategy. In all cases, this is a more efficient use of time than going all the way through the strategy creation process only to reject it at the end.

Evaluating Indicators

In many ways, evaluating indicators in a vacuum is harder than evaluating other parts of the strategy. It is nearly impossible if you cannot express a theory of action for the indicator.

Why? This is true because the indicator is not just some analytical output that has no grounding (at least we hope not). A good indicator is describing some measurable aspect of reality: a theoretical "fair value" price, or the impact of a factor on that price, or turning points of the series, or slope.

If we have a good conceptualization of the hypothesized properties of the indicator, we can construct what the signal processing field calls the 'symmetric filter' on historical data. These are often optimized ARMA or Kalman-like processes that filter out all the 'noise' to capture some pre-determined features of the series. They're (usually) useless for prediction, but are very descriptive of past behavior.

It is also possible to do this empirically if you can describe the properties in ways that can be readily programmed. For example, if the indicator is supposed to indicate turning points, and is tuned to a mean duration of ten periods, then you can look for all turning points that precede a short term trend centered around ten periods. The downside of this type of empirical analysis is that it is inherently a one-off process, developed for a specific indicator.

How is this useful in evaluating an indicator? Once you have constructed the symmetric filter, you can evaluate the degree to which the indicator matches perfect hindsight. Using kernel methods, clustering, mean squared error, or distance measures we can evaluate the degree of difference between the indicator and the symmetric filter. A low degree of difference (probably) indicates a good indicator for the features that the symmetric filter has been tuned to pick up on.

The danger of this approach comes if you've polluted your data by going over it too many times. Specifically, if you train the indicator on the entire data set, then of course it has the best possible fit to the data. Tests for look ahead bias and data snooping bias can help to detect and guard against that, but the most important guard has to be the analyst paying attention to what they are doing, and constructing the indicator function or backtest to not use non-contemporaneously available data. Ideally, polluted training periods will be strictly limited and near the beginning of the historical series, or something like walk forward will be used (sparingly) to choose parameters so that the parameter choosing process is run over programmatically chosen subsets. Failure to exercise care here leads almost inevitably to overfitting (and poor out of sample results).

An indicator is, at it's core, a measurement used to make a prediction. This means that the broader literature on statistical predictors is valid. Many techniques have been developed by statisticians and other modelers to improve the predictive value of model inputs. See Kuhn and Johnson (2013) or Hastie `!@Hastie2009`. Input scaling, detrending, centering, de-correlating, and many other techniques may all be applicable. The correct adjustments or transformations will depend on the nature of the specific indicator.

Luckily, the statistics literature is also full of diagnostics to help determine which methods to apply, and what their impact is. You do need to remain cognizant of what you give up in each of these cases in terms of interpretability, trace-ability, or microstructure of the data.

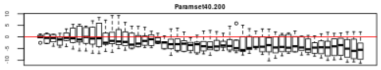
It is also important to be aware of the structural dangers of bars. Many indicators are constructed on periodic or "bar" data. Bars are not a particularly stable analytical unit, and are often sensitive to exact starting or ending time of the bar, or to the methodologies used to calculate the components (open, high, low, close, volume, etc.) of the bar. Further, you don't know the ordering of events, whether the high came before the low, To mitigate these dangers, it is important to test the robustness of the bar generating process itself, e.g. by varying the start time of the first bar. We will almost never run complete strategy tests on bar data, preferring to generate the periodic indicator, and then apply the signal and rules processes to higher frequency data.

In this way the data used to generate the indicator is just what is required by the indicator model, with more realistic market data used for generating signals and for evaluating rules and orders.

The analysis on indicators described here provides another opportunity to reject your hypothesis, and go back to the drawing board.

Evaluating Signals

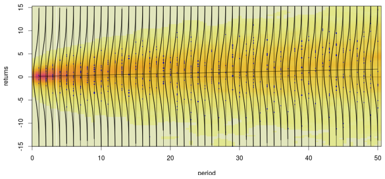
From one or more indicators, you usually proceed to examining the effectiveness of the signal generating process. One of the challenges in evaluating the full backtest is that the backtest needs to make multiple assumptions about fills. We can evaluate signals without making any assumptions about the execution that accompanies the desire to express a particular view.



A signal is a directional prediction for a point in time.

The point in time indicated for the signal will generally vary based on some parameterization of the indicator and/or signal function(s). These may be plotted using a forward looking boxplot which lines up returns from the signal forward by using the signal as t_0, and then displaying a boxplot of period returns from $t_{1 \ldots n}$. Without any assumptions about execution, we have constructed a distribution of forward return expectations from the signal generating process.

Many analyses can be done with this data. You can take a distribution of overall return expectations (quantiles, mean, median, standard deviation, and standard errors of these estimates), generate a conditional return distribution, examine a linear or non-linear fit between the period expectations, and compare the expectations of multiple different input parameters. When comparing input parameter expectations, you should see 'clusters' of similar positive and/or negative return expectations in similar or contiguous parameter combinations. Existence of these clusters indicates what Tomasini and Jaekle (2009) refer to as a 'stable region' for the parameters (see parameter optimization below). A random assortment of positive expectations is a bad sign, and should lead to reviewing whether your hypotheses and earlier steps are robust.



The signals generated by the backtest(s) are also able to be empirically classified. You can measure statistics such as the number of entry and exit signals, the distribution of the period between entries and exits, degree of overlap between entry and exit signals, and so on.

Given the empirical data about the signal process, it is possible to develop a simulation of statistically similar processes. A simulation of this sort is described by Aronson (2006) for use in evaluating entire strategy backtest output (more on this below), but it is also useful at this earlier stage. These randomly simulated signal processes should display (random) distributions of returns, which can be used to assess return bias introduced by the historical data. If you parameterize the statistical properties of the simulation, you can compare these parameterized simulations to the expectation generated from each parameterization of the strategy's signal generating process.

It is important to separate a parameterized and conditional simulation from an unconditional simulation. Many trading system Monte Carlo tools utilize unconditional sampling. They then erroneously declare that a good system must be due to luck because its expectations are far in the right-hand tail of the unconditional distribution. One of the only correct inferences that you can deduce from a unconditional simulation would be a mean return bias of the historical series.

It would make little sense to simulate a system that scalps every tick when you are evaluating a system with three signals per week. It is critical to center the simulation parameters around the statistical properties of the system you are trying to evaluate. The simulated signal processes should have nearly indistinguishable frequency parameters (number of enter/exit signals, holding period, etc.) to the process you are evaluating, to ensure that you are really looking at comparable things.

Because every signal is a prediction, when analyzing signal processes, we can begin to fully apply the literature on model specification and testing of predictions. From the simplest available methods such as mean squared model error or kernel distance from an ideal process, through extensive evaluation as suggested for Akaike's Information Criterion(AIC), Bayesian Information Criterion(BIC), effective number of parameters, cross validation of Hastie, Tibshirani, and Friedman (2009), and including time series specific models such as the data driven "revealed performance" approach of Racine and Parmeter (2009): all available tools from the forecasting literature should be considered for evaluating proposed signal processes.

It should be clear that evaluating the signal generating process offers multiple opportunities to re-evaluate assumptions about the method of action of the strategy, and to detect information bias or luck before moving on to (developing and) testing the rest of the strategy.

Evaluating Rules

By the time you get to this stage, you should have experimental confirmation that the indicator(s) and signal process(es) provide statistically significant information about the instruments you are examining. If not, stop and go back and reexamine your hypotheses. Assuming that you do have both a theoretical and empirical basis on which to proceed, it is time to define the strategy's trading rules.

Much of the work involved in evaluating "technical trading rules" described in the literature is really an evaluation of signal processes, described in depth above. Rules should refine the way the strategy 'listens' to signals, producing path-dependent actions based on the current state of the market, your portfolio, and the indicators and signals. Separate from whether a signal has predictive power or not, as described above, evaluation of rules is an evaluation of the actions taken in response to the rule.

entry rules

Most signal processes designed by analysts and described in the literature correspond to trade entry. Described another way, every *entry rule* will likely be tightly coupled to a *signal* (possibly a composite signal). If the *signal* (prediction) has a positive expectation, then the *rule* should have potential to make money. It is most likely valuable with any proposed system to test both aggressive and passive entry order rules.

If the system makes money in the backtest with a passive entry on a signal, but loses money with an aggressive entry which crosses the bid-ask spread, or requires execution within a very short time frame after the signal, the likelihood that the strategy will work in production is greatly reduced.

Conversely, if the system is relatively insensitive in the backtest to the exact entry rule from the signal process, there will likely be a positive expectation for the entry in production.

Another analysis of entry rules that may be carried out both on the backtest and in post-trade analysis is to extract the distribution of duration between entering the order and getting a fill. Differences between the backtest and production will provide you with information to calibrate the backtest expectations. Information from post trade analysis will provide you with information to calibrate your execution and microstructure parameters.

You can also analyze how conservative or aggressive your backtest fill assumptions are by analyzing how many opportunities you may have had to trade at the order price after entering the order but before the price changes, or how many shares or contracts traded at your order price before you would have moved or canceled the order.

exit rules

There are two primary classes of exit rules, signal based and empirical; evaluation and risks for these two groupings is different. Exit rules, whether being driven by the same signal process as the entry rules, or based on empirical evidence from the backtest, are often the difference between a profitable and an unprofitable strategy.

Signal driven exit rules may follow the same signal process as the entry rules (e.g. band and midpoints models, or RSI-style overbought/oversold models), or they may have their own signal process. For example, a process that seeks to identify slowing or reversal of momentum may be different from a process to describe the formation and direction of a trend. When evaluating signal based exits, the same process from above for testing aggressive vs. passive order logic is likely valuable. Additionally, testing trailing order logic after the signal may "let the winners run" in a statistically significant manner.

Empirical exit rules are usually identified after initial tests of other types of exits, or after parameter optimization (see below). They include classic risk stops (see below) and profit targets, as well as trailing take profits or pullback stops. Empirical profit rules are usually identified using the outputs of things like MEan Adverse Excursion(MAE)/Mean Favorable Excurison(MFE), for example:

- MFE shows that trades that have advanced x % or ticks are unlikely to advance further, so the trade should be taken off
- a path fit is still strongly positive, even though the signal process indicates to be on the lookout for an exit opportunity, so a trailing take profit may be in order

See more on MAE/MFE below.

risk rules

There are several types of risk rules that may be tested in the backtest, and the goals of adding them should be derived from your business objectives. By the time you get to adding risk rules in the backtest, the strategy should be built on a confirmed positive expectation for the signal process, signal-based entry and exit rules should have been developed and confirmed, and any empirical profit taking rules should have been evaluated. There are two additional "risk" rule types that should commonly be considered for addition to the strategy, empirical risk stops, and business constraints.

Empirical risk stops are generally placed such that they would cut off the worst losing trades. This is very strategy dependent. You may be able to get some hints from the signal analysis described above, some examples:

- a path dependent fit is strongly negative after putting on the trade
- the return slope is negative after x periods

Or, you may place an empirical risk stop based on what you see after doing parameter optimization or post trade analysis (see tools below) to cut off the trade after there is a very low expectation of a winning trade, for example:

- x ticks or % against the entry price
- in the bottom quantile of MAE

The business constraint example we most often see in practice is that of a drawdown constraint in a levered portfolio. The drawdown constraint is a clear business requirement, imposed by the leverage in the book. It is possible to impose drawdown constraints utilizing a risk rule that would flatten the position after a certain loss. While it is common to use catastrophic loss rules in strategies, we have had better luck controlling portfolio drawdowns via diversified asset allocation and portfolio optimization than via risk stops.

Bailey and López de Prado (forthcoming, 2014) examines the impact of drawdown based risk rules, and adjustments to order sizing based on drawdowns or other changes in account equity. They develop an autoregressive model for estimating the potential for drawdowns even in high-Sharpe or short history investments. They further model the likely time to recover as investment sizes are varied following the drawdown.

It is also possible to model risk rules such as hedges or option insurance, but these are best done in asset allocation and money management as an overlay for the strategy or a portfolio of strategies, and not in the individual strategy itself. Rules to model orthogonal portfolio hedges or option insurance significantly complicate the task of evaluating the impact of the rules designed for the regular functioning of the strategy.

As with other parts of strategy development and testing, it is again important to formulate a hypothesis about the effects of exit rules that you plan to add, and to document their intended effects, and the business reasons you've chosen for adding them. This is especially important if you are adding rules after parameter optimization. Done poorly, exit or risk rules added after parameter optimization are just cherry picking the best returns, while if done properly they may be used to take advantage of what you've learned about the statistical expectations for the system, and may improve out of sample correspondence to the in-sample periods.

order sizing

We tend to do minimal order sizing in backtests. Most of our backtests are run on "one lot" or equivalent portfolios. The reason is that the backtest is trying to confirm the hypothetical validity of the trading system, not really trying to optimize execution beyond a certain point. How much order sizing makes sense in a backtest is dependent on the strategy and your business objectives. Typically, we will do "leveling" or "pyramiding" studies while backtesting, starting from our known-profitable business practices.

We will *not* do "compounding" or "percent of equity" studies, because our goal in backtesting is to confirm the validity of the positive expectation and rules embodied in the system, not to project revenue opportunities. Once we have a strategy model we believe in, and especially once we have been able to calibrate the model on live trades, we will tend to drive order sizing from microstructure analysis, post trade analysis of our execution, and portfolio optimization.

rule burden

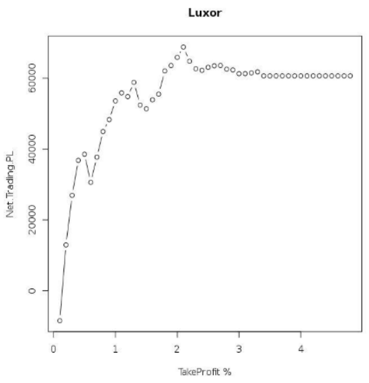
Beware of rule burden. Too many rules will make a backtest look excellent in-sample, and may even work in walk forward analysis, but are very dangerous in production. One clue that you are overfitting by adding too many rules is that you add a rule or rules to the backtest *after* running an exhaustive analysis and being disappointed in the results. This is introducing data snooping bias. Some data snooping is unavoidable, but if you've run multiple parameter optimizations on your training set or (worse) multiple walk forward analyses, and then added rules after each run, you're likely introducing dangerous biases in your output.

Parameter optimization[10]

Parameter optimization is important for indicators, signals, rules, and complete trading systems. Care needs to be taken to do this as safely as possible. That care needs to start with clearly defined objectives, as has already been stated. The goal of parameter optimization should be to locate a parameter combination that most closely matches the

[10] *Every trading system is in some form an optimization.* (Tomasini)

hypotheses and objectives. To avoid overfitting, it is also critical to avoid outliers, looking for stable regions of both in and out of sample performance. (Tomasini and Jaekle 2009 , pp. 49–56)



When a parameter or set of parameters is robust, it will have a few key properties:

- small parameter changes lead to small changes in P&L and objective expectations
- out of sample deterioration is not large, on average (see walk forward optimization)
- parameter choices have a sound theoretical or economic basis
- parameter variation should produce correlated differences in multiple objectives

For small numbers of parameters, it may be sufficient to graph parameters against your objective, use quantiles, or examine neighbors around a candidate parameter set. When there are more than a couple of parameters, or more than one objective, the inference to find a "stable region" is somewhat more difficult. Quantiles of the parameter sets lined up against the objectives are still useful, but you may need to cluster these values to find the best intersection, or apply formal global optimization solvers with a penalized objective to locate the best combination of parameters.

Limiting the degrees of freedom in your parameter optimization is another way to guard against overfitting and data mining biases.

Most real strategies have only a few influential parameters, out of many other "fine tuning" parameters. When backtesting, focus on the major drivers. These major drivers should be known in advance, from the theory of the strategy. If they are not obvious from the theory of the strategy, first work on refining the hypothesis so its premises may be tested. If that is not possible or is still insufficient, effective parameter testing (Hastie, Tibshirani, and Friedman 2009) can help to identify the major drivers in a small training and testing set from the beginning of the data. By limiting the degrees of freedom in this way, you limit the opportunities to cherry pick lucky combinations, and statistically the required adjustment for data mining bias is reduced, as are the number of trials (resulting in faster parameter searches).

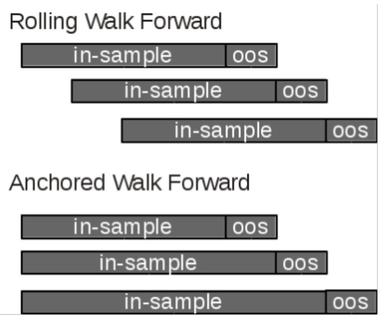
Incremental evaluation of rules is another advanced form of parameter optimization usually paired with more advanced global optimization solvers or machine learning. Incremental rule evaluation runs the strategy with different combinations of rules. In a normal evaluation of this type, rules are separated by type (entry, exit, risk, rebalance, and so on), and various combinations are tried, starting with minimal enter/exit pairs, and adding incrementally to complexity. These types of trials can be very informative as to what adds value to the strategy.

If not carefully controlled, they can also result in significant out of sample deterioration (see walk forward analysis).

Choosing the best parameter set should be a function of your business objectives. In the simplest case, your selection algorithm should maximize or minimize your business objective. In more complex cases, you will likely have multiple business objectives, and will utilize a penalized multi-objective optimization to choose the best combination. In an ideal case, you will also have a view on slope and linearity of the equity curve, described by something like Kestner's K-Ratio(2003), which is really the t-test on the linear model, or by other descriptive statistics for goodness of fit or model choice (see Ripley 2004 for an overview).

Walk Forward Analysis

A logical extension of parameter optimization, walk forward analysis utilizes a rolling or expanding window as 'in-sample' (IS) for parameterization and another period as 'out of sample' (OOS) using the chosen parameters.

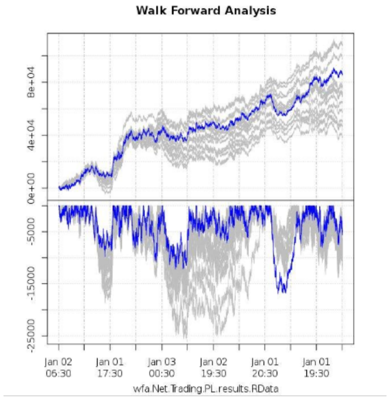


Most strategies do not have stable parameters through time. The world's top investors, in all styles and frequencies, adjust their expectations, outlook, and approach over time as market conditions change. Warrent Buffett or Ray Dalio say that they follow the same "strategy" over decades, but they still adapt to changing markets.

Walk forward analysis allows the parameterization of the strategy to change over time as market conditions change.

Walk forward analysis guards against data mining bias. At first, this may be counter-intuitive, as walk forward analysis is a data mining process.

Barring regime shift (more on this later), the strategist generally assumes local stationarity in market conditions. If we are conducting parameter optimization in alignment with our previously stated business objectives (supplemented by statistical measures of goodness of fit), then utilizing walk forward analysis describes the strategy parameterization that would have been most appropriate for the business at a point in time. If you are truly in a region of local stationarity in market characteristics, then you are choosing optimal parameters for your out of sample period *without* looking ahead and incurring bias.



Another guard against data snooping bias is to have your walk forward periods be realistic. If the strategy is hard to parameterize, or uses hard to get or periodic data, then your walk forward analysis should reflect these difficulties.

For example, you should make sure that you:

- Choose between anchored walk forward or rolling walk forward based on whether the theoretical properties of your indicators(s) would benefit from that.
- Only reparameterize the free parameters for the whole strategy on a frequency that makes sense for your strategy and business. Indicators that adapt to changes in market data automatically are probably safer than doing frequent parameter searches of the recent past.
- Don't vary the length of in and out of sample periods across multiple tests, unless you are using such variations to model out of sample deterioration.
- If you use macroeconomic data released periodically (monthly or quarterly), your in and out of sample periods should reflect when information arrives to change how the strategy operates.
- Understand whether windowing effects would strongly influence the results. If windowing effects are present, consider an anchored walk forward.

In addition to the choosing mechanism used for parameter analysis, you may additionally evaluate turnover and correspondence between in and out of sample performance. On turnover, you may choose to stay with the parameter set you are already running, or the set with the lowest turnover, if the performance of this zero or low turnover parameter set is within the error bounds of the estimation. If you are utilizing a population-based optimizer, it makes sense to evaluate the top performers from the in-sample period in the out of sample period, and utilize the ones with the least drift as part of your starting population for your next (non-overlapping) walk forward period.

Walk forward analysis should be used sparingly. One risk of using walk forward analysis is that you can introduce data snooping biases if you apply it multiple times with differing goals, looking for the best outcomes[11]. As with objective and hypothesis generation, you should be clear about the definition of success before performing the test and polluting your data set with prior knowledge.

Regime Analysis

Parameter optimization and walk forward analysis assume local stationarity.

How do you know if this is a reasonable assumption?

Regime models attempt to provide this information. Many financial time series exhibit distinct states, with specific properties differing sharply between these different 'regimes'. There are two primary classes of models that are used for detecting regimes, Markov switching models and change point models. In the context of strategy development, it can make sense to both test for regime behaviors and develop a plan for managing regime change. If your data has well-defined regimes, and you can model these, you may find that very different parameter sets are appropriate for the strategy in those regimes, or that no parameter set produces reasonable results out of sample in particularly challenging regimes, or that no change is necessary. In any case, once you have a strategy model you believe in, doing some regime analysis may provide some additional marginal benefit.

Evaluating the whole system[12]

Pardo says (pp. 202–209):

> Some of the key characteristics of a robust trading strategy are:
>
> 1. A relatively even distribution of trades
> 2. A relatively even distribution of trading profit
> 3. Relative balance between long and short profit
> 4. A large group of contiguous, profitable strategy parameters in the optimization
> 5. Acceptable trading performance in a wide range of markets
> 6. Acceptable risk
> 7. Relatively stable winning and losing runs
> 8. A large and statistically valid number of trades
> 9. A positive performance trajectory

[11] see cross validation section, below, for additional cautionary notes about correct use of walk forward analysis

[12] *Net profit as a sole evaluation method ignores many of the characteristics important to this decision. - Robert Pardo (2008)*

Evaluating Trades

Entire books have been written extolling the virtues or lamenting the problems of one performance measure over another. We have chosen to take a rather inclusive approach both to trade and P&L based measures and to return based measures (covered later). Generally, we run as many metrics as we can, and look for consistently good metrics across all common return and cash based measures. Trade and P&L based measures have an advantage of being precise and reconcilable to clearing statements, but disadvantages of not being easily comparable between products, after compounding, etc.

What is a trade, anyway?

The word "trade" has different meanings in different parts and types of strategy analysis. Certainly a single transaction is a "trade". When speaking of "trades" in "trade statistics" you usually mean a pair or more of transactions that both open and close a position.

There are several ways of calculating the round trip "trades". The most common are FIFO, tax lots, flat to flat, and flat to reduced.

1. FIFO

   FIFO is "first in, first out", and pairs entry and exit transactions by time priority. We generally do not calculate statistics on FIFO because it is impossible to match P&L to clearing statements; very few institutional investors will track to FIFO. FIFO comes from accounting for physical inventory, where old (first) inventory is accounted for in the first sales of that inventory. It can be very difficult to calculate a cost basis if the quantities of your orders vary, or you get a large number of partial fills, as any given closing fill may be an amalgam of multiple opening fills.

2. tax lots

   Tax lot "trades" pair individual entry and exit transactions together to gain some tax advantage such as avoiding short term gains, harvesting losses, lowering the realized gain, or shifting the realized gain or loss to another tax period or tax jurisdiction. This type of analysis can be very beneficial, though it will be very dependent on the goals of the tax lot construction.

3. flat to flat

   Flat to flat "trade" analysis marks the beginning of the trade from the first transaction to move the position off zero, and marks the end of the "trade" with the transaction that brings the P&L back to zero, or "flat". It will match brokerage statements of realized

P&L when the positions is flat and average cost of open positions always, so it is easy to reconcile production trades using this methodology. One advantage of the flat to flat methodology is that there are no overlapping "trades" so doing things like bootstrap, jackknife, or Monte Carlo analysis on those trades is not terribly difficult. The challenge of using flat to flat is that it will not work well for strategies that are rarely flat; if a strategy adjusts its position up or down, levels into trades, etc. then getting useful aggregate statistics from flat to flat analysis may be difficult.

4. flat to reduced

   The flat to reduced methodology marks the beginning of the "trade" from the first transaction to increase the position, and ends a "trade" when the position is reduced at all, going closer to zero. Brokerage accounting practices match this methodology exactly, they will adjust the average cost of the open position as positions get larger or further from flat, and will realize gains whenever the position gets smaller (closer to flat). Flat to reduced is our preferred methodology most of the time. This methodology works poorly for bootstrap or other trade resampling methodologies, as some of the aggregate statistics are highly intercorrelated because multiple "trades" share the same entry point. You need to be aware of the skew in some statistics that may be created by this methodology and either correct for it or not use those statistics. Specific problems will depend on the strategy and trading pattern. The flat to reduced method is a good default choice for "trade" definition because it is the easiest to reconcile to statements, other execution and analysis software, and accounting and regulatory standards.

5. increased to reduced

   An analytically superior alternative to FIFO is "increased to reduced". This approach marks the beginning of a "trade" any time the position increases (gets further from flat/zero) and marks the end of a trade when the position is decreased (gets smaller in absolute terms, closer to flat), utilizing average cost for the cost basis. Typically, flat to flat periods will be extracted, and then broken into pieces matching each reduction to an increase, in expanding order from the first reduction. This is "time and quantity" priority, and is analytically more repeatable and manageable than FIFO. If you have a reason for utilizing FIFO-like analysis, consider using "increased to reduced" instead. Be aware that the "trade" statistics will not be reconcilable to any brokerage statement, this is purely an analytical methodology.

Aggregate trade statistics are calculated on the entire backtest.

Basically all modern backtesting and execution platforms can calculate and report aggregated trade statistics for the entire test period. Typically, you will be looking at these statistics to confirm that the performance of the strategy is as expected for its class, and in-line with the expectations you established from the components of the strategy.

- common trade statistics :
  - number of transactions and "trades" performed
  - gross and net trading profits/losses (P&L)
  - mean/median trading P&L per trade
  - standard deviation of trade P&L
  - largest winning/losing trade
  - percent of positive/negative trades
  - Profit Factor : absolute value ratio of gross profits over gross losses
  - mean/median P&L of profitable/losing trades
  - annualized Sharpe-like ratio
  - max drawdown
  - start-trade drawdown (Fitschen 2013, 185)
  - win/loss ratios of winning over losing trade P&L (total/mean/median)

Some authors advocate testing the strategy against "perfect profit", the potential profit of buying every upswing and selling every downswing. Such a "perfect profit" number is meaningless in almost all circumstances, as it relies on a number of assumptions of periodicity, execution technology, etc. that are certain to be false. While it would be possible to create an "ideal profit" metric for a given strategy which would have similar stochastic entry/exit properties to the strategy being evaluated, our assertion is that this type of simulation is more appropriately situated with analysis of the signal process, as described above. It is certainly possible to extend the signal analysis as described above to a given fixed set of rules, but we believe this is an invitation to overfitting, and prefer to only perform that kind of speculative analysis inside the structure of a defined experimental design such as parameter optimization, walk forward analysis, or *k*-fold cross validation on strategy implementations. Leave the simulated data much earlier in the process when confirming the power of the strategy components.

The goal of many trading strategies is to produce a smoothly upward sloping equity curve. To compare the output of your strategy to an idealized representation, most methodologies utilize linear models. The linear fit of the equity curve may be tested for slope, goodness of fit, and confidence bounds. Kestner (2003) proposes the K-Ratio, which is the p-test statistic on the linear model. Pardo (2008, 202–9) emphasizes the number of trades, and the slope of the resulting curve. You can extend the analysis beyond the simple linear model to generalized linear models or generalized additive models, which can provide more robust fits that do not vary as much with small changes to the inputs.

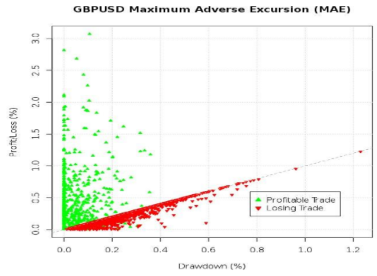
Dangers of aggregate statistics . . .

- hiding the most common outcomes
- focusing on extremes
- not enough trades or history for validity
- colinearities of flat to reduced

Per-Trade statistics can provide insight for fine-tuning. Analyzing individual trades can give you insight into how the strategy performs on a very fine-grained level.

Especially for path dependent behaviors such as:

- developing a distribution of trade performances and volatilities,
- validating the signal expectations, which were inherently forward looking,
- quick dips or run-ups immediately after the trade is put on,
- frequent drawdowns from peak before taking the trade off,
- positive or negative excursions while the trade is on.



Analyzing excursions can provide empirical support from the backtest for setting risk stops or profit taking levels.

Maximum Adverse Excursion (MAE) or Maximum Favorable Excursion (MFE) show how far down (or up) every trade went during the course of its life-cycle. You can capture information on how many trades close close to their highs or lows, as well as evaluating points at which the P&L for the trade statistically just isn't going to get any better, or isn't going to recover. While not useful for all strategies, many strategies will have clear patterns that may be incorporated into risk or profit rules.

You can also separate the individual trades into quantiles (or other slices) by any statistic that you have available. The different distributions, paths, and properties of these quantiles will frequently provide insight into methods of action in the strategy, and can lead to further strategy development.

It is important when evaluating MAE/MFE to do this type of analysis in your test set. One thing that you want to test out of smaple is whether the MAE threshold is stable over time. You want to avoid, as with other parts of the strategy, going over and "snooping" the data for the entire test period, or all your target instruments.

Post Trade Analysis

Post trade analysis offers an opportunity to calibrate the things you learned in the backtest, and generate more hypotheses for improving the strategy. Analyzing fills may proceed using all the tools described earlier in this document. Additionally, you now have enough data with which to model slippage from the model prices, as well as any slippage (positive or negative) from the other backtest statistics.

One immediate benefit for post trade analysis is that you already have all the tools to evaluate performance; they have been applied to every component of the strategy. Tests and output of all the analyses are already specified, and may be compared directly.

Some analysis is unique to post trade analysis. Direct reconciliation between the backtest and production requires both the theoretical and production output have all the same information available for modeled prices, orders, and execution reports. Once in the same format, it is possible to analyze variances in:

- modeled theoretical price (these should match in most cases)
- order timing and prices
- P&L
- evolution of the position, including partial fill analysis

Backtests tend to model complete fills, and use deliberately conservative fill assumptions. One outcome of this is that many strategies can receive more or earlier fills in production. If you've modeled your signal expectations, these may be run over the production period as well, and you can develop a model for when and how completely fills occur in production after the signals and rules of the model would have wanted to be filled. It is sometimes also reasonable to have a limited adjustment to (overly) conservative fill assumptions of the backtest more clearly match reality. You can also model slippages at this point, in price, quantity, and time.

Our practice is to impose haircuts on the research model based on production results. All negative results, whether the fault of the model or of execution, will be used to lower the backtest expectations on future trials. All positive production results better than the model will be appropriately celebrated, but will not be used to 'upward adjust' the model.

Reconciliation of the differences will often lead to improvements in the strategy, or its production execution.

Microstructure Analysis

Microstructure can not be easily tested in a backtest. The nature of backtesting is that you want to try many combinations quickly.

As such, the data you use for a backtest is almost always at least minimally reduced from full L2 market data. If you have a desire or a need to formulate an opinion about how the strategy will perform given market microstructure, this will be very hard to simulate, even from full L2 data. While simulators exist for L2 data that are quite accurate most of those are stochastic, very computationally intensive, and need to be run hundreds or thousands of times with minor variations in assumptions and timing.

Some things can be known from the L2 data without a simulator:

- distribution or composition of the book
- likelihood that the market will turn to a new price
- total shown size for a given level at a certain point in time

To refine expectations of execution, you have to gather real trade data.

By doing post trade analysis (see above) you can line up orders and fills with the L1 or L2 book data, dealing with things like jitter and skew in the timestamps. You can then draw inferences about likelihoods of getting filled, market impact, how your MAE/MFE and other trade statistics relate to the backtest.

When much of the overall net performance of the strategy depends on the execution the only way to refine that expectation is to gather real data on the target system, or to use an informative prior constructed from a similar system that was or is actually traded in production. When you know that the signal process has a positive expectation, and a reasonable backtest also shows a positive expectation, one of the best ways to adjust the theoretical expectation is to calibrate it against the output of post trade analysis.

Evaluating Returns

Returns create a *standard* mechanism for comparing multiple strategies or managers using all the tools of portfolio and risk analysis that have been developed for investment and portfolio management. Returns based analysis is most valuable at daily or lower periodicities (often requiring aggregating intraday transaction performance to daily returns) , is easily comparable across assets or strategies, and has many analytical techniques not available for trades and P&L. Returns are not easily reconcilable back to the trades both because of the transformation done to get to returns, changes in scale over time, and possible aggregation to lower or regular frequencies different from the trading frequency.

Choice of the denominator matters. In a retail account, capital is a specific number and is usually not in question. Accounting for additions and withdrawals is straightforward. In a back test or in an institutional account, the choice of denominator is less clear. Several options exist, any or all of which may be reasonable. The first is to use a fixed denominator. This has the advantage of being simple, and assumes that gains are not reinvested. Another related option is to use a fixed starting capital. This option is as simple as the fixed denominator, but assumes that gains are reinvested into the strategy. A third option is to look at return on exchange margin. Return on margin has the challenge of knowing what the required margin was, which requires access to things like a SPAN engine, or to historical portfolio reports from live trading of the strategy.

Should we use cash or percent returns? If the strategy utilizes the same order sizing in contract or cash terms throughout the tested period, then cash P&L should work fine. If instead the "money management" of the strategy reinvests gains or changes order sizes based on account equity, then you will need to resample percent returns, or your results will not be unbiased. The cash P&L would exhibit a (presumably upward) bias from the portfolio growth in the backtest portfolio over the course of the test if the strategy is reinvesting. Conversely, if the strategy does not change order sizes during the test (e.g. fixed 1-lot sizing), then using percent returns versus account equity will show a downward bias in returns, as account equity grows, but order sizes remain constant. In this case, you may choose to use a fixed/constant denominator to get to simple "returns", which would be the equivalent of saying that any cash generated by the strategy was withdrawn.

Many analytical techniques presume percent return-based analysis rather than analysis on cash P&L. Some examples include: - tail risk measures - volatility analysis - factor analysis - factor model Monte Carlo - style analysis - applicability to asset allocation (see below)

Comparing strategies in return space can also be a good reason to use percent returns rather than cash. When comparing strategies, worknig in return-space may allow for disparate strategies to be placed on a similar footing. Things like risk measures often make more sense when described against their percent impact on capital, for example.

While it may be tempting to do all of the analysis of a trading strategy in only cash P&L, or only in return space, it is valuable to analyze almost every strategy in both ways, as the two approaches provide different insight.

Rebalancing and asset allocation

Once the strategy is basically complete, and preferably in production, then you can work at marginal improvement to return and risk via rebalancing and asset allocation. We prefer to do this analysis on real trade data (see Post Trade Analysis) rather than backtest data, when possible.

Asset allocation should support your business objectives for return and risk. This may be a good time to revisit the business objective and make sure the business objective is suitable for optimization. If the business objectives are formulated as described above in the section on business objectives, this should require no additional work.

The *Kelly Criterion*, and the closely related *optimal f* and *Leverage Space Portfolio Model* (*LSPM*) of Vince (2009) seek to choose an optimal bet size given uncertainty of the outcome of any single bet. Kelly and optimalf come out of betting, where the goal is to determine the optimal bet size given some edge and a certain capital pool. LSPM extends this model to a portfolio allocation context given a joint distribution of drawdowns. All three of these models seek to maximize final wealth, the first two without regard to path, and LSPM while limiting the joint probability of some drawdown to a business-acceptable threshold. The most difficult challenge in utilizing LSPM is in estimating the joint probability and magnitude of drawdowns, making the method increasingly difficult for larger portfolios of trading strategies.

There are some particular challenges in using portfolio optimization for trading strategies. Once you have real trade data, and have aggregated it to a suitable time frame (typically daily), that daily cash return for the strategy or an instrument or configuration within the strategy is essentially a synthetic instrument return. When doing portfolio allocation among multiple choices, you ideally want a large number of observations. An order of magnitude more observations for each asset, strategy, or configuration than the number of synthetic assets you'll have in the optimization is usually near the minimum stable observations set. So 60 observations for each of six synthetic assets in the portfolio, for example.

What if you don't have enough data? Let's suppose you want 500 observations for each of 50 synthetic assets based on the rule of thumb above. That is approximately two years of daily returns. This number of observations would likely produce a high degree of confidence if you had been running the strategy on 50 synthetic assets for two years in a stable way. If you want to allocate capital in alignment with your business objectives before you have enough data, you can do a number of things: - use the data you have and re-optimize frequently to check for stability and add more data - use higher frequency data, e.g. hourly instead of daily - use a technique such as Factor Model Monte Carlo (Jiang 2007, Zivot 2011,2012) to construct equal histories - optimize over fewer assets, requiring a smaller history

Fewer assets? Suppose you have three strategies, each with 20 configurations. Using the rule of thumb above, you'd want a minimum of 600 days of production results. If instead you use the aggregate return of each of the three strategies, you could likely get a directionally correct optimization result from 30 or so observations, or a month and a half of daily data with all three strategies running. More data is generally better, if available. This compares well in practice to the three years of monthly returns (36 observations) often used for fund of hedge funds investments.

What if you have lots of strategies and configurations? Suppose we have many strategies, each with many configurations or traded assets. In the likely case that you don't have enough data to optimize over all the configurations as in the example above, you can optimize over just the aggregate strategy returns as described above. At this point you most mature strategy or strategies may very well have enough data for optimization separately. This opens the way for what is called layered objectives and optimization. You may have different business objectives for a single strategy, e.g. the objectives for a market maker an a medium term trend follower are different.

In this case, it is preferable to optimize the configurations for a single strategy, generating an OOS return for the strategy on a daily scale that may be used as the input to the multi-strategy optimization described above. This layered optimization is well-supported in **R**.

How often should you rebalance? Most academic literature on optimal portfolio allocation seems to operate on an assumption of continuous rebalancing. Such an assumption is rarely warranted in practice due to:

- cost of rebalancing trades
- time to perform complex portfolio optimization
- data management required to be ready to do optimization
- uncertainty in conversion from weights to order sizes (see below)
- conversion of optimization output into strategy parameters

It typically makes sense to choose a less frequent rebalance period. Periods which may make sense for your production portfolio include, but are not limited to:

- calendar periods such as weeks or months
- periods when cash is added or withdrawn
- infrequently enough to measure out of sample deterioration with reasonable statistical confidence
- when new strategies leave small scale incubation and are ready to be scaled up to larger production sizes
- at a frequency efficient for your available computing resources

Also discuss:

- *rebalancing implications*
- *discuss implications of Factor Model Monte Carlo*
- *techniques for backing out from weights to capital to order sizes*

Probability of Overfitting[13]

This entire paper has been devoted to avoiding overfitting. At the end of the modeling process, we still need to evaluate how likely it is that the model may be overfit, and develop a haircut for the model that may identify how much out of sample deterioration could be expected.

With all of the methods described in this section, it is important to note that you are no longer measuring performance; that was

[13] *We should recognize the reality that any simulated (backtest) performance presented to us likely overstates future prospects. By how much? -Antti Ilmanen* (2011) p. 112

covered in prior sections. At this point, we are measuring statistical error, and developing or refuting the level of confidence which may be appropriate for the backtest results. We have absolutely *fitted* the strategy to the data, but is it *over*-fit? With what confidence and error estimates can we identify the fitting?

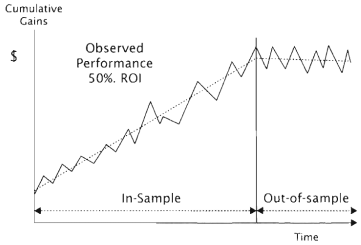


Figure 1: source:Aronson

out of sample deterioration

We need to measure the degradation that occurs OOS in a consistent fashion.

- linear measurement via lm
- measurement and reconciliation of trade stats
- application of OOS deterioration measurement to Walk Forward and *k*-folds

It can be very informative to consider all of the IS and OOS periods from walk forward analysis, and apply the OOS degradation measurements to these periods. Over time, you can hope to learn about modifications to your objective function or strategy development process which would decrease OOS degradation.

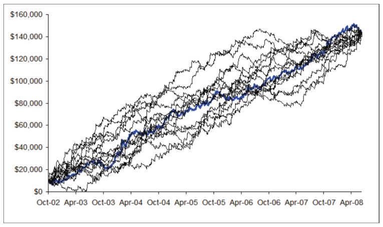


Figure 2: source:Tomasini

resampled trades

Tomasini(2009, 104–9) describes a basic resampling mechanism for trades. The period returns for all "flat to flat" trades in the backtest (and the flat periods with period returns of zero) are sampled from without replacement. After all trades or flat periods have been sampled, a new time series is constructed by applying the original index to the resampled returns. This gives a number of series which will have the same mean and net return as the original backtest, but differing drawdowns and tail risk measures. This allows confidence bounds to be placed on the drawdown and tail risk statistics based on the resampled returns.

This analysis can be extended to resample without replacement only the duration and quantity of the trades. These entries, exits, and flat periods are then applied to the original price data. This will generate a series of returns which are potentially very different from the original backtest.[14] In this model, average trade duration, percent time in market, and percent time long and short will remain the same, but all the performance statistics will vary based on the new path of the returns. This should allow a much more coherent placement of the chosen strategy configuration versus other strategy configurations with similar time series statistical properties.

Burns (2006) describes a number of tests that may be applied to the resampled returns to evaluate the key question of skill ver-

[14] We are applying these methodologies here to gain or refute confidence in backtested results. All of these analytical methods may also be applied to post trade analysis to gain insight into real trades and execution processes.

sus luck. You should be able to determine via the p-value control statistics some confidence that the strategy is the product of skill rather than luck as well as the strength of the predictive power of the strategy in a manner similar to that obtained earlier on the signal processes. You can also apply all the analysis that was utilized on evaluating the strategy and its components along the way to the resampled returns, to see where in the distribution of statistics the chosen strategy configuration fits.

What if your strategy is never flat? These resampling methods presume flat to flat construction, where trades are marked from flat points in the backtest. Things get substantially more complicated if the strategy is rarely flat, consider for example a long only trend follower which adds and removes position around a core bias position over very long periods of time, or a market making strategy which adds and reduces positions around some carried inventory. In these cases constructing resampled returns that are statistically similar to the original strategy is very difficult. One potential choice it to make use of the "increased to reduced" trade definition methodology. If you are considering only one instrument, or a series of independent instruments, the increased to reduced methodology can be appropriately applied.[15]

If the instruments are dependent or highly correlated, the difficulty level of drawing useful statistical inference goes up yet again. In the case of dependent or highly correlated instruments or positions, you may need to rely on portfolio-level Monte Carlo analysis rather than attempting to resample trades. While it may be useful to learn if the dependent structure adds to overall returns over the random independent resampled case, it is unlikely to give you much insight into improving the strategy, or much confidence about the strategy configuration that has made it all the way to these final analyses.

what can we learn from resampling methods?
what would be incorrect inferences?

Monte Carlo

If there is strong correlation or dependence between instruments in the backtest, then you will probably have to resample or perform Monte Carlo analysis from portfolio-level returns, rather than trades. You lose the ability to evaluate trade statistics, but will still be able to assess the returns of the backtest against the sampled portfolios for your primary business objectives and benchmarks. As was discussed in more detail under *Evaluating Signals*, it is important that any resampled data preserve the autocorrelation structure of the original

[15] one of the only cases in which a FIFO style methodology provides useful statistical information

data to the degree possible.

White's Reality Check

White's Data Mining Reality Check from White (2000) (usually referred to as DRMC or just "White's Reality Check" WRC) is a bootstrap based test which compares the strategy returns to a benchmark. The ideas were expanded in Hansen (2005). It creates a set of bootstrap returns and then checks via abolute or mean squared error what the chances that the model could have been the result of random selection. It applies a *p-value* test between the bootstrap distribution and the backtest results to determine whether the results of the backtest appear to be statistically significant.

cross validation

Cross validation is a widely used statistical technique for model evaluation.

In its classical statistical form, the data is cut in half, the model is trained on one half, and then the trained model is tested on the half of the model that was "held out". As such, this type of model will often be referred to as a *single hold out* cross validation model. In time series analysis, the classical formulation is often modified to use a smaller hold-out period than half the data, in order to create a larger training set. Challenges with single hold out cross validation include that one out of sample set is hard to draw firm inferences from, and that the OOS period may additionally be too short to generate enough signals and trades to gain confidence in the model.

In many ways, walk forward analysis is related to cross validation. The OOS periods in walk forward analysis are effectively validation sets as in cross validation. You can and should measure the out of sample deterioration of your walk forward model between the IS performance and the OOS performance of the model. One advantage of walk forward is that it allows parameters to change with the data. One disadvantage is that there is a temptation to make the OOS periods for walk forward analysis rather small, making it very difficult to measure deterioration from the training period. Another potential disadvantage is that the IS periods are overlapping, which can be expected to create autocorrelation among the parameters. This autocorrelation is mixed from an analysis perspective. A degree of parameter stability is usually considered an advantage. The IS periods are not all independent draws from the data, and the OOS periods will later be used as IS periods, so any analytical technique that assumes *i.i.d.* observations should be viewed at least with skepticism.

k-fold cross validation improves on the classical single hold-out

OOS model by randomly dividing the sample of size *T* into sequential sub-samples of size *T/k*.(Hastie, Tibshirani, and Friedman 2009) Alternately or additionally, something like a jackknife or block bootstrap may be used to maintain some of the serial structure while providing more samples. The strategy model is fit using the fitting procedure (e.g. parameter search and/or walk forward analysis) on the entire set of *k-1* samples, in turn, and then tested on the *k-th* sample which was held out. This procedure is repeated using each *k* sample as the holdout. There is some work which must be done on price-based vs return-based time series to make the series continuous, e.g. the series must be turned into simple differences and then back converted into a price series after the folds have been recombined, and an artificial time index must be imposed. One weakness with *k*-fold cross validation for trading strategies resides in the fact that the length of each *T/k* section must be long enough to have a reasonable error bound on the objectives in the OOS segments.(Bailey, Borwein, López de Prado, et al. 2014 , p.17) While we think that this is important to note, in the same manner that it is always important to understand the statistical error bounds of your calculations, it is not a fatal flaw.

Some question exists whether *k*-fold cross validation is appropriate for time series in the same way that it is for categorical or panel data. Rob Hyndman addresses this directly here[16] and here[17]. What he describes as "forecast evaluation with a rolling origin" is essentially Walk Forward Analysis. One important takeaway from Prof. Hyndman's treatment of the subject is that it is important to define the expected result and tests to measure forecast accuracy *before* performing the (back)test. Then, all the tools of forecast evaluation may be applied to evaluate how well your forecast is doing out of sample, and whether you are likely to have overfit your model.

linear models such as Bailey, Borwein, López de Prado, et al. (2014) and Bailey and López de Prado (2014)

- modifying existing expectations
- track the number of trials
- how do you define a 'trial'?
- CCSV sampling
- combinatorially symmetric cross validation "generate $S/2$ testing sets of size $T/2$ by recombining the S slices of the overall sample of size T. (Bailey, Borwein, López de Prado, et al. 2014, 17)

[16] `http://robjhyndman.com/hyndsight/tscvexample/`

[17] `https://www.otexts.org/fpp/2/5/`

- prior remarks about overlapping periods, parameter autocorrelation, and *i.i.d.* assumptions apply here as well
- Harvey and Liu (2013a) , Harvey and Liu (2014) , Harvey and Liu (2013b) look at Type I vs Type II error in evaluating backtests and look at appropriate haircuts based on this.

data mining bias

- data mining bias and cross validation from Aronson (2006)
- Cawley and Talbot (2010)
- Keogh and Kasetty (2003)

Acknowledgements

I would like to thank my team for thoughtful comments and questions, John Bollinger, Ilya Kipnis, and Stephen Rush at the University of Connecticut for insightful comments on early drafts of this paper.

All remaining errors or omissions should be attributed to the author. All views expressed in this paper are to be viewed as those of Brian Peterson, and do not necessarily reflect the opinions or policies of DV Trading or DV Asset Management.

The most recently published version of this document may be found at `http://goo.gl/na4u5d`

References

Aronson, David. 2006. *Evidence-Based Technical Analysis: Applying the Scientific Method and Statistical Inference to Trading Signals*. Wiley.

Bailey, David H, and Marcos López de Prado. 2014. "The Deflated Sharpe Ratio: Correcting for Selection Bias, Backtest Overfitting and Non-Normality." *Journal of Portfolio Management, Forthcoming*. `http://www.davidhbailey.com/dhbpapers/deflated-sharpe.pdf`.

———. forthcoming, 2014. "Drawdown-Based Stop-Outs and the 'Triple Penance' Rule." *Journal of Risk*. `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2201302`.

Bailey, David H, Jonathan M Borwein, Marcos López de Prado, and Qiji Jim Zhu. 2014. "The Probability of Backtest Overfitting." `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2326253`.

Bailey, David H, Jonathan M Borwein, Marcos López de Prado, and Qiji Jim Zhu. 2014. "Pseudomathematics and Financial Charlatanism: The Effects of Backtest Over Fitting on Out-of-Sample Performance." *Notices of the AMS* 61 (5): 458–71.

Baquero, Guillermo, Jenke Ter Horst, and Marno Verbeek. 2005. "Survival, Look-Ahead Bias, and Persistence in Hedge Fund Performance." *Journal of Financial and Quantitative Analysis* 40 (03). Cambridge Univ Press: 493–517. `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=371051`.

Box, George E.P., and Norman R. Draper. 1987. *Empirical Model-Building and Response Surfaces*. John Wiley & Sons.

Burns, Patrick. 2006. "Random Portfolios for Evaluating Trading Strategies." `http://www.burns-stat.com/pages/Working/evalstrat.pdf`.

Carroll, Robert. 2011. *The Skeptic's Dictionary: A Collection of Strange Beliefs, Amusing Deceptions, and Dangerous Delusions*. John Wiley & Sons. `http://skepdic.com/adhoc.html`.

Cawley, Gavin C, and Nicola LC Talbot. 2010. "On over-Fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation." *The Journal of Machine Learning Research* 11. JMLR. org: 2079–2107. `http://www.jmlr.org/papers/volume11/cawley10a/cawley10a.pdf`.

Diedesch, Josh. 2014. "2014 Forty Under Forty." *Chief Investment Officer*. California State Teachers' Retirement System. `http://www.ai-cio.com/Forty_Under_Forty_2014.aspx?page=9`.

Feynman, Richard P, Robert B Leighton, Matthew Sands, and EM Hafner. 1965. *The Feynman Lectures on Physics*. Vols. 1-3.

Fitschen, Keith. 2013. *Building Reliable Trading Systems: Tradable*

Strategies That Perform as They Backtest and Meet Your Risk-Reward Goals. John Wiley & Sons, Inc.

Hansen, Peter R. 2005. “A Test for Superior Predictive Ability.” *Journal of Business and Economic Statistics*.

Harvey, Campbell R., and Yan Liu. 2013a. “Backtesting.” *SSRN*. `http://ssrn.com/abstract=2345489`.

———. 2013b. “Multiple Testing in Economics.” *SSRN*. `http://ssrn.com/abstract=2358214`.

———. 2014. “Evaluating Trading Strategies.” *Journal of Portfolio Management* 40 (5): 108–18. `https://faculty.fuqua.duke.edu/~charvey/Research/Published_Papers/P116_Evaluating_trading_strategies.pdf`.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition*. Springer.

Horowitz, Ben. 2014. *The Hard Thing About Hard Things: Building a Business When There Are No Easy Answers*. HarperCollins.

Ilmanen, Antti. 2011. *Expected Returns: An Investor's Guide to Harvesting Market Rewards*. John Wiley & Sons.

Keogh, Eamonn, and Shruti Kasetty. 2003. “On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration.” *Data Mining and Knowledge Discovery* 7 (4). Springer: 349–71. `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.2240&rep=rep1&type=pdf`.

Kestner, Lars. 2003. *Quantitative Trading Strategies: Harnessing the Power of Quantitative Techniques to Create a Winning Trading Program*. McGraw-Hill.

Kuhn, Max, and Kjell Johnson. 2013. *Applied Predictive Modeling*. Springer. `http://appliedpredictivemodeling.com/`.

Pardo, Robert. 2008. *The Evaluation and Optimization of Trading Strategies, Second Edition*. John Wiley & Sons.

Racine, Jeffrey S, and Christopher F Parmeter. 2009. “Data-Driven Model Evaluation: A Test for Revealed Performance.” *Mac Master University*. `https://editorialexpress.com/cgi-bin/conference/download.cgi?db_name=FEMES09&paper_id=152`.

Ripley, Brian D. 2004. “Selecting Amongst Large Classes of Models.” *Methods and Models in Statistics: In Honor of Professor John Nelder, FRS*, 155–70.

Smith, Martha K. “Common Misteaks Mistakes in Using Statistics: Spotting and Avoiding Them - Data Snooping.” `https://www.ma.utexas.edu/users/mks/statmistakes/datasnooping.html`.

Sullivan, Ryan, Allan Timmermann, and Halbert White. 1999. “Data Snooping, Technical Trading Rule Performance, and the Bootstrap.” *The Journal of Finance* 54 (5): 1647–91.

Tomasini, Emilio, and Urban Jaekle. 2009. "Trading Systems: A New Approach to System Development and Portfolio Optimisation."

Tukey, John W. 1962. "The Future of Data Analysis." *The Annals of Mathematical Statistics*. JSTOR, 1–67. `http://projecteuclid.org/euclid.aoms/1177704711`.

Vince, Ralph. 2009. *The Leverage Space Trading Model: Reconciling Portfolio Management Strategies and Economic Theory*. Vol. 425. John Wiley; Sons.

White, Halbert L. 2000. "System and Method for Testing Prediction Models and/or Entities." Google Patents. `http://www.google.com/patents/US6088676`.