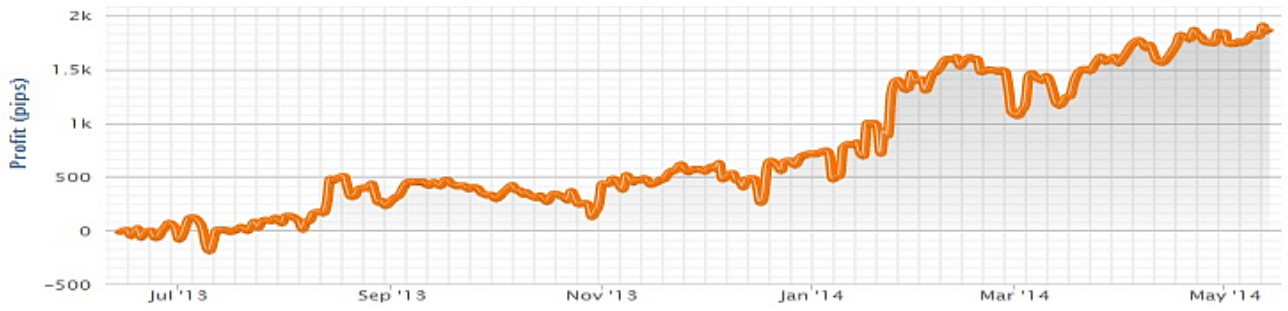
The Financial Hacker
A new view on algorithmic trading

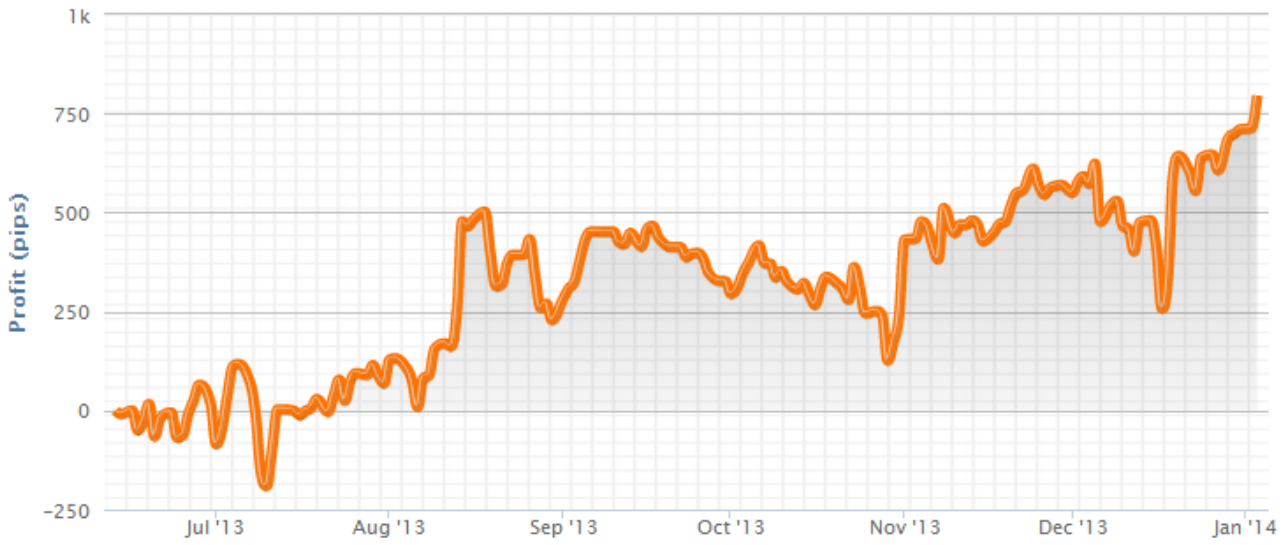
The Cold Blood Index



You've developed a new trading system. All tests produced impressive results. So you started it live. And are down by $2000 after 2 months. What now? **Carry on in cold blood, or pull the brakes in panic?** This is a situation all too familiar to any algo trader.

Several reasons can cause a strategy to lose money right from the start. It can be already **expired** since the market inefficiency disappeared. Or the system is worthless and the test wrong due to some **bias** that survived all reality checks. Or it's just a **normal drawdown**. In this article I propose an algorithm for deciding very early whether or not to abandon a system in such a situation.

When you start a trading strategy, you're almost always under water for some time. This is a normal consequence of **equity curve volatility**. It is the very reason why you need initial capital at all for trading (aside from covering margin costs). Here you can see the typical bumpy start of a trading system:



CHF grid trader, initial live equity curve

You can estimate from the live equity curve that this system was rather profitable (it was a grid trader exploiting the CHF price cap). It started in July 2013 and had earned about 750 pips in January 2014, 7 months later. Max drawdown was ~400 pips from September until November. So the raw annual return of that system was about 750/400 * 12/7 months = 300%. Normally an excellent value for a trade system. But you can also see from the curve that you were down 200 pips about six weeks into trading, and thus had lost almost half of the minimum initial capital. And if you had started the system in September, you had even stayed under water for more than 3 months! This is a psychologically difficult situation. Many traders get in a panic, pull out, and this way **lose money even with highly profitable systems**. Algo trading unaffected by emotions? Not true.

Not so out of sample

The basic problem: you can never fully trust your test results. No matter how out-of-sample you test it, a strategy still suffers from a certain amount of **Data-Snooping Bias**. The standard method of measuring bias – **White's Reality Check** – works well for simple mechanical systems, as in the Trend Experiment. But all human decisions about algorithms, asset selection, filters, training targets, stop/takeprofit mechanisms, WFO windows, money management and so on add new bias, since they are normally affected by testing. The out-of-sample data is then not so out-of-sample anymore. While the bias by training or optimization can often be measured, the **bias introduced by the mere development process**

is unknown. The strategy might still be profitable, or not anymore, or not at all. You can only find out by comparing live results permanently with test results.

You could do that with no risk by trading on a demo account. But if the system is really profitable, demo time is sacrificed profit and thus expensive. Often very expensive, as you must demo trade a long time for some result significancy, and many strategies have a limited lifetime anyway. So you normally demo trade a system only a few weeks for making sure that the script is bug-free, then you go live with real money.

Pull-out conditions

The simplest method of comparing live results is based on the **maximum drawdown** in the test. This is the pull-out inequality:

$$E < C + G\frac{t}{y} - D$$

E = Current account equity
C = Initial account capital
G = Test profit
t = Live trading period
y = Test period
D = Test maximum drawdown

This formula means simply that you should pull out when the live trading drawdown exceeds the maximum drawdown from the test. Traders often check their live results this way, but there are many problems involved with this method:

- The maximum backtest drawdown is more or less random.
- Drawdowns grow with the test period, thus longer test periods produce worse maximum drawdowns and later pull-out signals.
- The drawdown time is not considered.
- The method does not work when profits are reinvested by some money management algorithm.

- The method does not consider the unlikeliness that the maximum drawdown happens already at live trading start.

For those reasons, the above pullout inequality is often modified for taking the drawdown length and growth into account. The maximum drawdown is then assumed to **grow with the sqare root of time**, leading to this modified formula:

$$E < C + G\frac{t}{y} - D\sqrt{\frac{t+l}{y}}$$

E = Current account equity
C = Initial account capital
G = Test profit
t = Live trading period
y = Test period
D = Maximum drawdown depth
l = Maximum drawdown length

This was in fact the algorithm that I often suggested to clients for supervising their live results. It puts the drawdown in relation to the test period and also considers the drawdown length, as the probability of being inside the worst drawdown right at live trading start is ***y/l***. Still, the method does not work with a profit reinvesting system. And it is dependent on the rather random test drawdown. You could address the latter issue by taking the drawdown from a Montecarlo shuffled equity curve, but this produces new problems since trading results have often serial correlation.

After this lenghty introduction for motivation, here's the proposed algorithm that overcomes the mentioned issues.

Keeping cold blood

For finding out if we really must immediately stop a strategy, we calculate the deviation of the current live trading situation from the strategy behavior in the test. For this we do not use the maximum drawdown, but the backtest equity or balance curve. Here's the algorithm:

1. Determine a time window of length ***l*** (in days) that you want to check. It's normally the length of the current drawdown; if your system is not in a drawdown, you're probably in cold blood anyway. Determine the drawdown depth ***D***, i.e. the net loss during that time.
2. Place a time window of same size ***l*** at the start of the test balance curve.
3. Determine the balance difference ***G*** from end to start of the window. Increase a counter ***N*** when ***G*** <= ***D***.
4. Move the window forward by 1 day.
5. Repeat steps 3 and 4 until the window arrived at the end of the balance curve. Count the steps with a counter ***M***.

Any window movement takes a sample out of the curve. We have ***N*** samples that are similar or worse, and ***M***-***N*** samples that are better than the current trading situation. The probability to **not** encounter such a drawdown in ***T*** out of ***M*** samples is a simple combinatorial equation:

$$1\text{-}P = \frac{(M\text{-}N)!(M\text{-}T)!}{M!(M\text{-}N\text{-}T)!}$$

N = Number of ***G*** <= ***D*** occurrences
M = Total samples = ***y***-***l***+***1***
l = Live DD length in days
y = Test time in days
T = Samples taken = ***t***-***l***+***1***
t = Live trading time in days

P is the **cold blood index** – the similarity of the live situation with the backtest. A ***P*** of 0.5 or higher indicates that all is fine. If ***P*** is very low or zero, either our backtest was biased or the market has changed. The system can still be profitable, just not as profitable as in the test. But when the current loss ***D*** is significant, we should stop.

Often we want to calculate **P** soon after the begin of live trading. The window size ***l*** is then identical to our trading time ***t***, hence ***T*** == ***1***. This simplifies the equation to:

$$P = \frac{N}{M}$$

In such a situation I'd give up and pull out of a painful drawdown as soon as ***P*** drops below 5%.

The slight disadvantage of this method is that you must perform a backtest with the same capital allocation as in live trading, and store its balance or equity curve in a file for later evaluation when live trading. However this should only take a few lines of code in a strategy script.

Here's a small example script for Zorro that calculates ***P*** (in percent) from a stored balance curve when a trading time **t** and drawdown of length ***l*** and depth ***D*** is given:

```
int TradeDays = 40;    // t, Days since live start
int DrawDownDays = 30; // l, Days since you're in drawdown
var DrawDown = 100;    // D, Current drawdown depth in $

string BalanceFile = "Log\\BalanceDaily.dbl"; // stored double array

var logsum(int n)
{
  if(n <= 1) return 0;
  return log(n)+logsum(n-1);
}

void main()
{
  int CurveLength = file_length(BalanceFile)/sizeof(var);
  var *Balances = file_content(BalanceFile);

  int M = CurveLength - DrawDownDays + 1;
  int T = TradeDays - DrawDownDays + 1;

  if(T < 1 || M <= T) {
    printf("Not enough samples!");
    return;
  }

  var GMin=0., N=0.;
  int i=0;
  for(; i < M; i++)
  {
    var G = Balances[i+DrawDownDays-1] - Balances[i];
    if(G <= -DrawDown) N += 1.;
    if(G < GMin) GMin = G;
  }

  var P;
  if(TradeDays > DrawDownDays)
    P = 1. - exp(logsum(M-N)+logsum(M-T)-logsum(M)-logsum(M-N-T));
  else
```

```
    P = N/M;

  printf("\nTest period: %i days",CurveLength);
  printf("\nWorst test drawdown: %.f",-GMin);
  printf("\nM: %i N: %i T: %i",M,(int)N,T);
  printf("\nCold Blood Index: %.1f%%",100*P);
}
```

Since my computer is unfortunately not good enough for calculating the factorials of some thousand samples, I've summed up the logarithms instead – therefore the strange **logsum** function in the script.

Conclusion

- Finding out early whether a live trading drawdown is 'normal' or not can be essential for your wallet.
- The backtest drawdown is a late and inaccurate criteria.
- The Cold Blood Index gives you the precise probability of encountering a such a drawdown while the system was live, based on the backtest balance curve.

I've added the script above to the 2015 scripts collection. I also have suggested to the Zorro developers to implement this method for automatically analyzing drawdowns while live trading, and issue warnings when P gets dangerously low. This can also be done separately for components in a portfolio system. This feature will probably appear in Zorro version 1.44.

jcl / October 26, 2015 / Indicators, System Evaluation / Cold Blood Index, Data Mining Bias, Drawdown, Grid Trading, Zorro

1 thought on "The Cold Blood Index"

trader

October 26, 2015 at 14:08

You articles are fantastic! Thank you very much!