

Implement a hedge fund strategy in 10 lines

Thomas Schmelzer

thomas.schmelzer@gmail.com

<https://github.com/tschm/cs>

DISCLAIMER

This talk has been provided for information purposes only and is subject to change. It should not be interpreted as investment advice or any recommendation to buy or sell a particular security. The value of investments and the income derived from them may go down as well as up, investors may not recover what they invest. Investments in hedge funds are speculative and risks include, among other things, loss of all or a substantial portion of the investment due to leveraging, short-selling, use of derivatives or other speculative practices. FX transactions that are speculative carry, due to their complex nature, a high degree of financial risk. Investments in foreign currencies involve the additional risk that the foreign currency might lose value against

- Summer 2017: Lunch with Cyrus Fazel (CEO, Swissborg)
- Motivation to implement a CTA strategy with less than 10 lines of code
- Today: Attempt to help demystifying quantitative hedge funds and their level of sophistication.

Baz, Jamil and Granger, Nicolas M. and Harvey, Campbell R. and Le Roux, Nicolas and Rattray, Sandy, Dissecting Investment Strategies in the Cross Section and Time Series (December 4, 2015). Available at SSRN: <https://ssrn.com/abstract=2695101> or <http://dx.doi.org/10.2139/ssrn.2695101>

CTA = Commodity Trading Advisor

- Man AHL
- Winton Capital
- Aspect Capital
- BlueTrend (part of BlueCrest, today Systematica)
- Solaise Capital
- Ashenden Capital
- ...
- Barclays CTA Index

```
In [4]: # Load prices for 49 futures  
prices = pd.read_csv("data/prices.csv", index_col=0, parse_dates=True).ffill().trunca
```

In a first experiment we run a classic MACD system:

Experiment 1

In a second experiment we volatility-adjust the positions of a classic MACD system:

Experiment 2

In a third experiment we work with adjusted prices and apply oscillators

Experiment 3

```
In [ ]: from pycta.signal import *

# compute adjusted log returns
prices_adj = prices.apply(returns_adjust, com=96, clip=3).cumsum()

vola = prices.pct_change().ewm(com=96, min_periods=300).std()

# take two moving averages and apply tanh
def f(price, slow=96, fast=32, clip=3):
    return np.tanh(osc(price, fast=fast, slow=slow))

portfolio = Portfolio(prices=prices, position=prices_adj.apply(f)/vola)
```

```
In [ ]: analysis(portfolio.nav())
```

- I have published essentially the code above on LinkedIn in 2017.
- I got approximately 270 likes (that's 10σ for me), numerous comments and a few private messages.
- <https://www.linkedin.com/pulse/implement-cta-less-than-10-lines-code-thomas-schmelzer/>

- All systems we have seen are **univariate** and lack risk management (adding assets will increase the risk).
- It's a good idea to scale positions with the inverse volatility.
- The maintenance of univariate systems is almost trivial.
- All systems operate in the space of cash positions.

My personal Views

- Strategies should be **multivariate** as assets tend to be highly correlated.
- A strategy should measure and control the risk.
- The cash position is of limited use (in the context of futures).
- It's helpful to allocate risk rather than capital (although both ideas are equivalent).
- Data is here to correct **bias**. Having a strong bias is helpful and required.

We take the step from univariate to multivariate systems and interpret univariate systems as a special case of the new class:

Optimization

In a fourth experiment we use the diagonal Markowitz approach

Experiment 4

In a fifth experiment we run Markowitz (with shrinkage in the sample correlation matrix) on a Futures portfolio

Experiment 5

Conclusions I

- Applying two moving averages gives a somewhat promising start (Experiment 1)
- All positions should be scaled by the inverse price volatility (Experiment 2)
- Using adjusted prices and scaled signals helps to address data problems and opens the door for statistics (Experiment 3)

Conclusions II

- Asset allocation can be done both in cash and risk space. Risk wins (Optimization)
- The common univariate systems are closely related to diagonal Markowitz (Experiment 4)
- Ignoring cross correlations is a bad idea (Experiment 5)
- The **smart combination of univariate trading systems** is extremely helpful to boost returns.

What you haven't seen I

- In practice you hardly can rely on the analytic solution of convex problems. Need solvers such as Mosek (and cvxpy).
- We optimized the expected return (in risk space). In practice it's a good idea to apply regularization here (Ridge, Lasso, Elastic Nets) and introduce some mock trading costs to tame the trading activity. A typical Markowitz system will trade far less than the univariate system.
- Applying Lasso/Elastic Nets regularization opens the door for sparse updates.

What you haven't seen II

- The construction of more powerful signals is about the **smart combination of standard signals** by using convex programming (Index-Tracking).
- We treated all underlying markets the same, e.g. did not distinguish capacity or measure market share
- We did not address the problem of low volatilities, e.g. the system will operate with often huge positions (lack of volatility in interest rates). Constraints on the leverage help here...