Trading Strategies That Are Designed Not Fitted

Robert Carver

QuantCon 2017 / New York / 29th April 2017

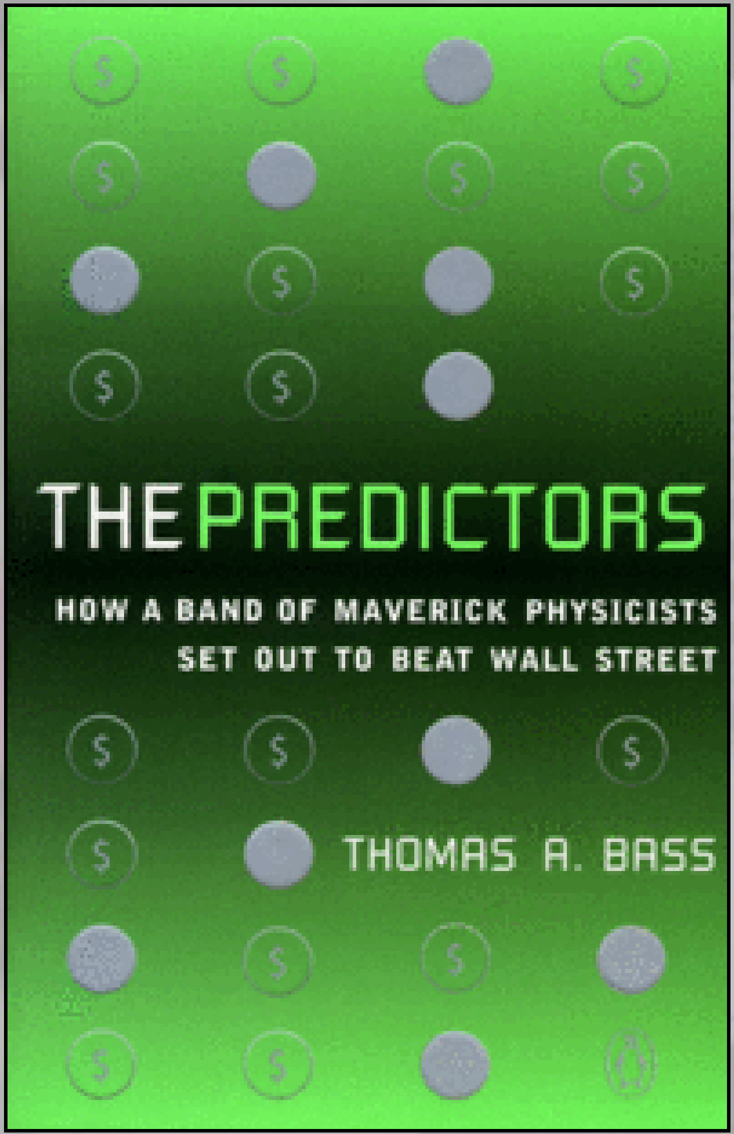
Legal boilerplate bit:

Nothing in this presentation constitutes investment advice, or an offer or solicitation to conduct investment business. The material here is solely for educational purposes.

Robert Carver is not currently regulated or authorised by the FCA, SEC, CFTC, or any other regulatory body to give investment advice, or indeed to do anything else.

Futures trading carries significant risks and is not suitable for all investors. Back tested and actual historic results are no guarantee of future performance. Use of the material in this presentation is entirely at your own risk.

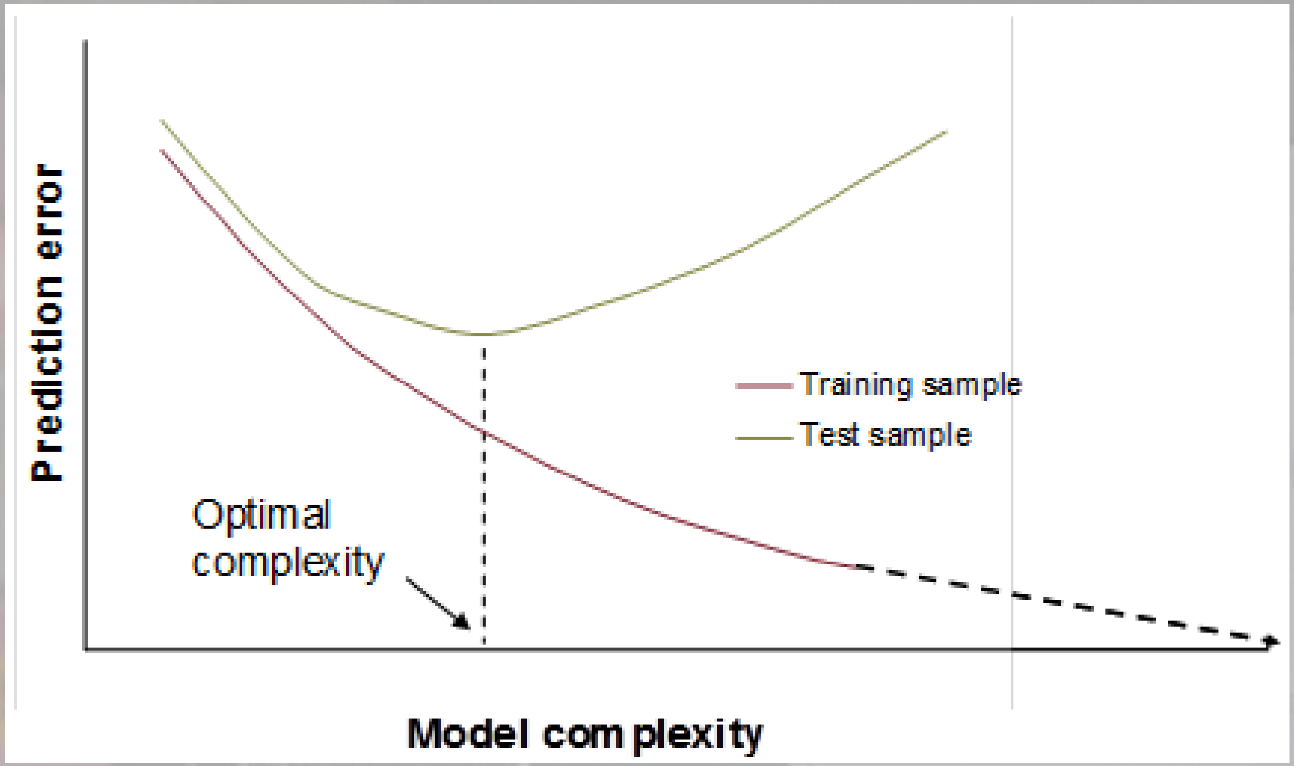
My 27 year old self



How my 27 year old self thought systematic trading design should work...

Data first

Why is fitting bad….



"The elements of statistical learning" by Hastie et al fig 2.11

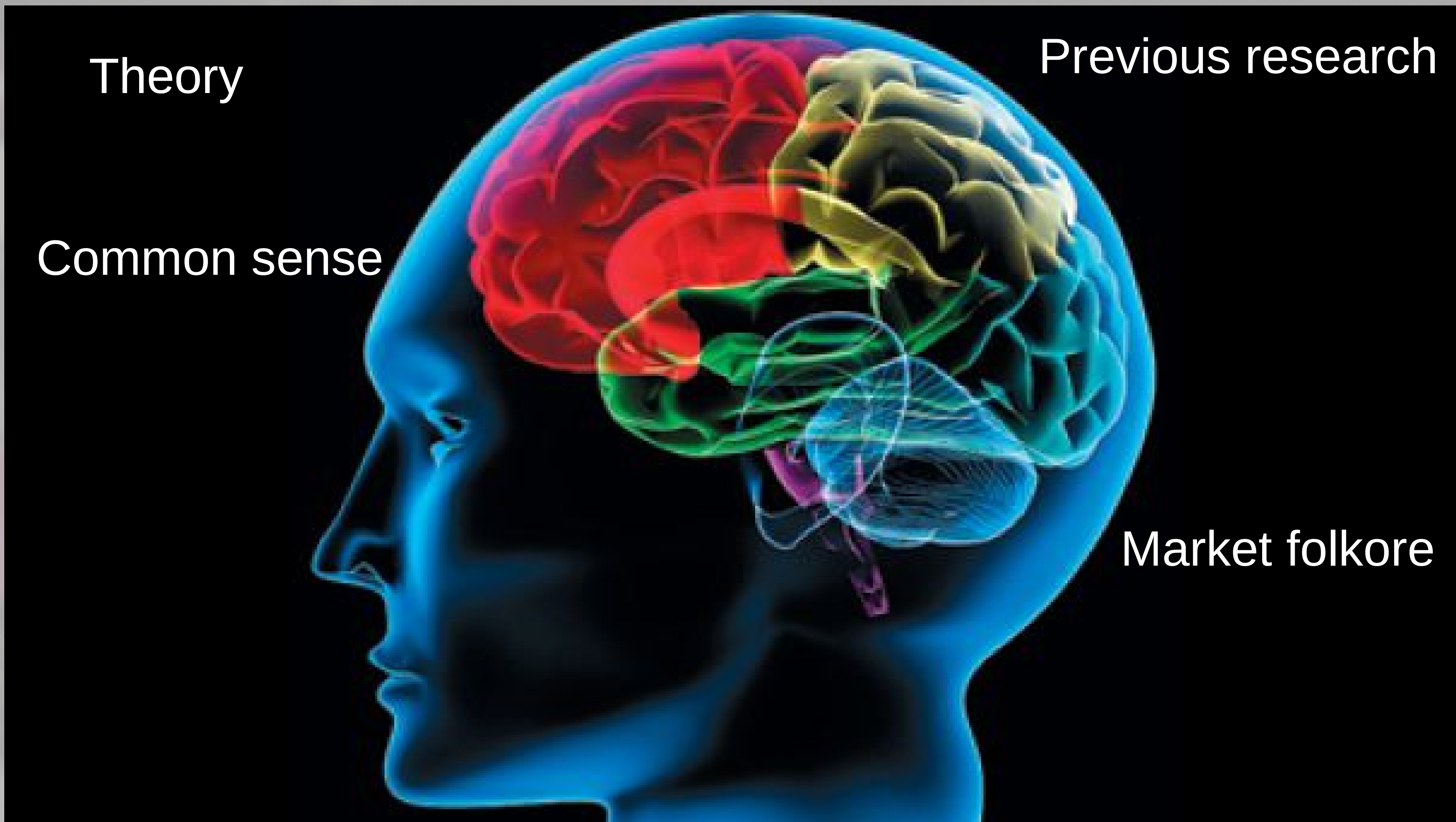
The three types of fitting...

Explicit

Implicit

Tacit

Tacit financial market knowledge



How do we **design** rather than **fit?**

- Acknowledge and embrace tacit knowledge
- Avoid implicit fitting
- Do the minimum amount of explicit fitting, and do it right.

Start with ideas not data

How my ~43 year old self *designs* trading strategies...

Tactit knowledge

Ideas first

Design process

Fake data

Real Data

Trading strategy

Algo + Parameters

How my ~43 year old self *designs* trading strategies...

Theory

Market folkore

Previous research

Common sense

Fake data

Real Data

Design process

How do people *design* real products...

Theory

Market research

Previous products

Personal taste

Focus group

Prototypes

Design process

Tacit knowledge: Trend following

- Market folklore:
  - “Cut your losers and let your winners run”
  - “Don’t fight the tape”
  - Turtle Traders
  - US CTA tradition (Campbell, Chesapeake, Dunn), UK CTA tradition (AHL, Winton, Aspect), Europeans (Transtrend, Systematica)

- Previous empirical research:
  - Levy 1960
  - Jegadeesh and Titman 1993
  - Carhart “fourth factor” 1997

- Theory:
  - Prospect theory (Kahneman and Tversky 1992),
  - Herding, Confirmation bias, under reaction.
  - Behaviour of other participants (eg risk parity funds)

Unanswered questions

- What period of time do trends last for?
- When should we enter trends?
- When should we exit trends?
- Should we have a stop loss rule? What is it?
- How do we identify markets that are, or aren’t “trend friendly”?
- How do we identify how strong the trend is?
- What size should our positions be?

- What is our algo?
- What are it’s parameters?

Data first

Data

Possible algos

Possible parameters

Magic box

Strategy = Best Algo +
Best parameters

What is best strategy? Data first answer:

Best <u>return versus risk</u> in dataset

Assuming leverage is possible and risk is Gaussian:

Highest Sharpe Ratio

(Other measures are available…)

Single metric of ‘best’: Performance
Single source of information: Past data

What is best strategy? Ideas first answer:

Best <u>designed</u> strategy

Multi-faceted metrics:
Performance, turnover, behaviour in given scenarios...

Multi-faceted sources of information:
Common sense, Theoretical principles, Fake data, (Limited amounts of) Real Data

Designing a trading strategy – 6 steps:

- Start with a sound framework which imposes some conditions
- Come up with the idea
- Use some random data or single scenario of real data plus theory / common sense to develop algo
- Use fake data to “fit” algo
- Real data for parameter sensitivity check / sense check
- Fit *allocation* using real data (out of sample, robust optimisation)

Designing a trading strategy – 6 steps:

- **Start with a sound framework which imposes some conditions**
- Come up with the idea
- Use some random data or single scenario of real data plus theory / common sense to develop algo
- Use fake data to “fit” algo
- Real data for parameter sensitivity check / sense check
- Fit *allocation* using real data (out of sample, robust optimisation)

Start with a sound framework

Trading rule 1

Trading rule 2

Trading rule 3

SP500

EDOLLAR

CORN

Combine forecasts from trading rules

Risk targeting

Position sizing

Portfolio: Weight instrument positions

Start with a sound framework: Conditions

- Trading rules make forecasts of risk adjusted price changes
- Forecasts are continous, not discrete entry + exit conditions
- Forecasts are scaled in an instrument / temporal independent way (no “magic numbers”)
- Forecast is proportional to E(Sharpe Ratio μ / σ)

  [Position is proportional to Forecast / σ hence position is proportional to μ / σ^2]
- E(abs(forecast)) = 10.0
- In principal all forecasts used on all markets (portfolio optimisation stage will become later)
- Use multiple variations of the same trading rule to capture different time frames (as many as possible, not too highly correlated)
- Costs are the most important thing. The second most important thing is costs. Costs are predictable – returns are not. Throw away very expensive systems.
- Throw away very slow systems (LAM)

Start with a sound framework
… remember these questions?

- What period of time do trends last for?
- When should we enter trends?
- When should we exit trends?
- Should we have a stop loss rule? What is it?
- How do we identify markets that are, or aren't "trend friendly"?
- How do we identify how strong the trend is?
- What size should our positions be?

Start with a sound framework
… remember these questions?

- ~~What period of time do trends last for?~~
- ~~When should we enter trends?~~
- ~~When should we exit trends?~~
- ~~Should we have a stop loss rule? What is it?~~
- ~~How do we identify markets that are, or aren’t “trend friendly”?~~
- **How do we identify how strong the trend is?**
- ~~What size should our positions be?~~

Fewer open questions:

Fewer parameters to “fit” or design

Designing a trading strategy – 6 steps:

- Start with a sound framework which imposes some conditions
- **Come up with the idea**
- Use some random data or single scenario of real data plus theory / common sense to develop algo
- Use fake data to “fit” algo
- Real data for parameter sensitivity check / sense check
- Fit *allocation* using real data (out of sample, robust optimisation)

Come up with the idea: What are trends?

Come up with the idea

Linear regression price against time, using Ordinary Least Squares:

$$\boldsymbol{y} = \boldsymbol{\alpha} + \boldsymbol{\beta} \boldsymbol{x} + \boldsymbol{\varepsilon} \qquad \text{minimise } \Sigma\varepsilon^2$$

with ***y*** = price, and ***x*** = some measure of time (eg years)

$\boldsymbol{\beta > 0}$ price in uptrend

$\boldsymbol{\beta < 0}$ price in downtrend

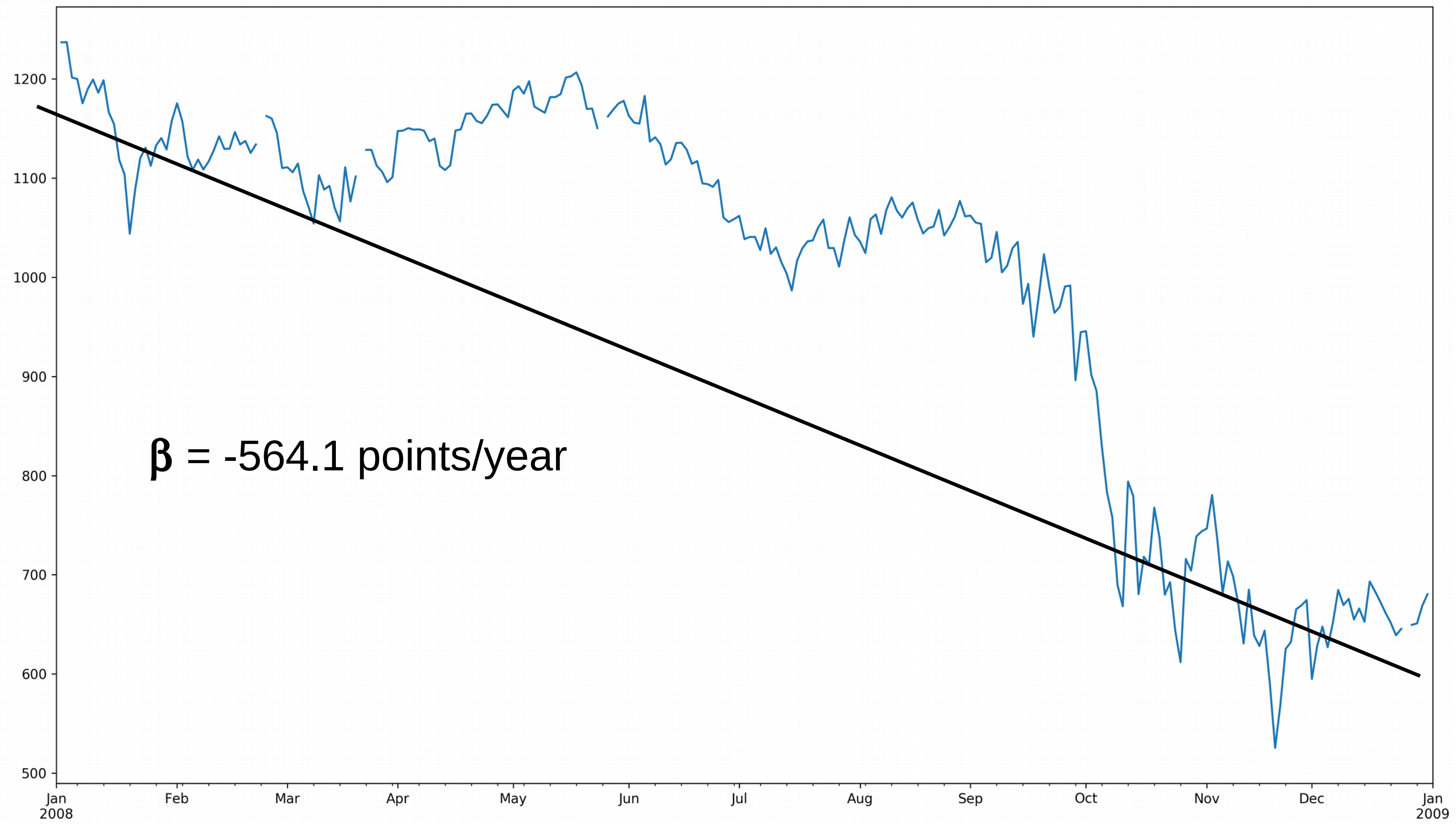
We use a rolling regression over the last N weekdays to capture different length trends.

Single parameter: **window_size**

Designing a trading strategy – 6 steps:

- Start with a sound framework which imposes some conditions
- Come up with the idea
- **Use some random data or single scenario of real data plus theory / common sense to develop algo**
- Use fake data to “fit” algo
- Real data for parameter sensitivity check / sense check
- Fit *allocation* using real data (out of sample, robust optimisation)

Develop algo: Real scenario… 2008



Conditions (reminder)

- Trading rules make forecasts of risk adjusted price changes
- Forecasts are continous, not discrete entry + exit conditions
- **Forecasts are scaled in an instrument / temporal independent way (no “magic numbers”)**
- Forecast is proportional to E(Sharpe Ratio μ / σ)

  [Position is proportional to Forecast / σ hence position is proportional to μ / σ^2]
- E(abs(forecast)) = 10.0
- In principal all forecasts used on all markets (portfolio optimisation stage will become later)
- Use multiple variations of the same trading rule to capture different time frames (as many as possible, not too highly correlated)
- Costs are the most important thing. The second most important thing is costs. Costs are predictable – returns are not. Throw away very expensive systems.
- Throw away very slow systems (LAM)

Develop algo: Evaluate design

Does this scale well? No…

(No need to look at data! **Common sense!**)

Forecast is proportional to E(Sharpe Ratio = μ / σ)
[Position is proportional to Forecast / σ hence position is proportional to μ / σ^2]

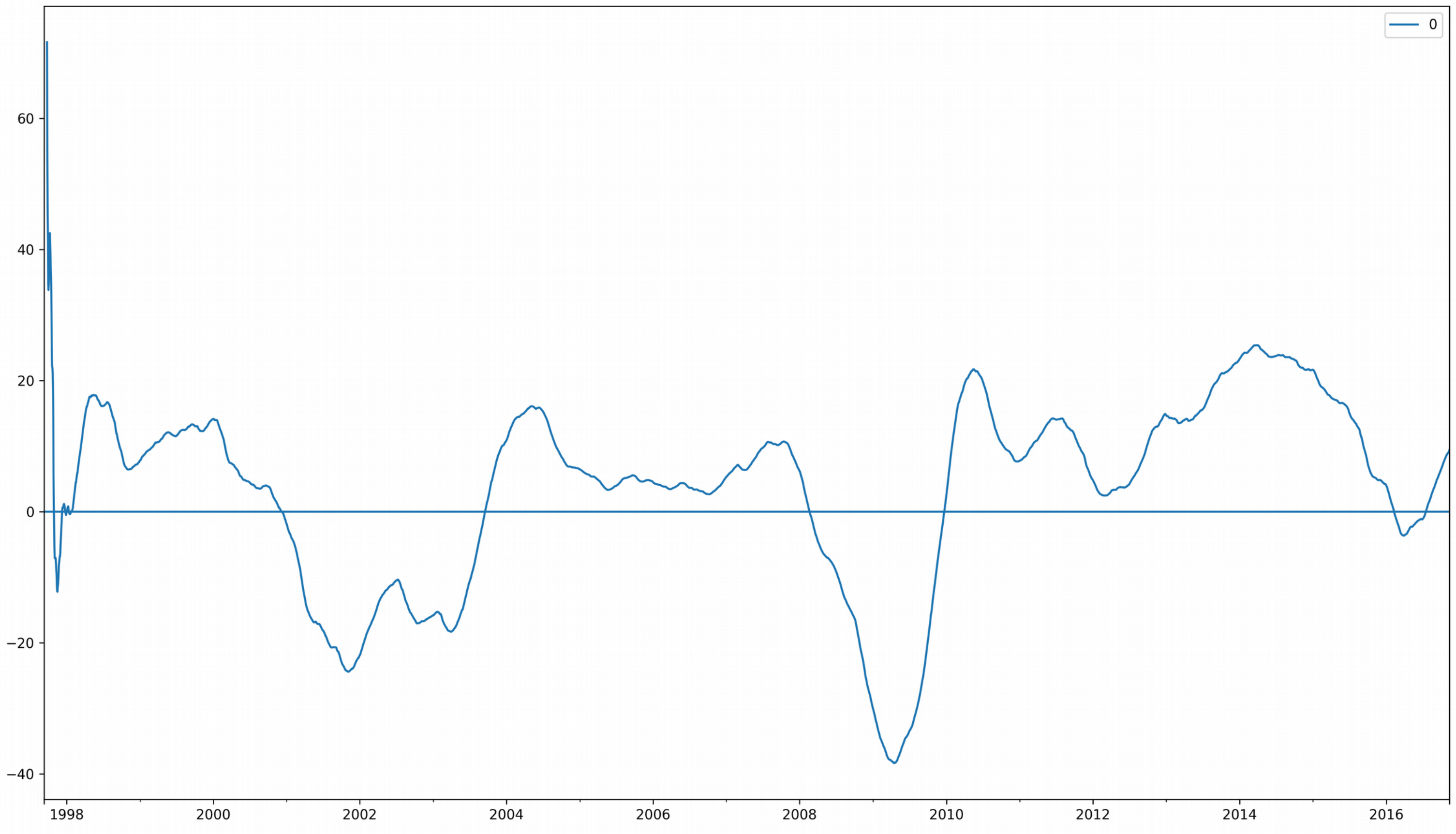
$\boldsymbol{\beta}$ in units of Δ(price) so:

Forecast = $\boldsymbol{\beta}$ / σ

Where σ is measured is annual standard deviation of Δ(price)

(No need to look at data! **Theory**)

Develop algo: 2nd iteration



Develop algo: Evaluate design

Does this scale well? Yes.

Does behaviour make sense? Yes.
Bullish in bull markets, bearish in bear markets

How about the trading speed? Seems reasonable given the length of trends involved

Anything weird? Yes
Need to set initial min_periods to a higher value (eg ***window_size*** / 4 : **Common sense!**)

Too slow? Probably N=256 is the slowest we'd go (LAM)

Designing a trading strategy – 6 steps:

- Start with a sound framework which imposes some conditions
- Come up with the idea
- Use some random data or single scenario of real data plus theory / common sense to develop algo
- **Use fake data to “fit” algo**
- Real data for parameter sensitivity check / sense check
- Fit *allocation* using real data (out of sample, robust optimisation)

Conditions (reminder)

- Trading rules make forecasts of risk adjusted price changes
- Forecasts are continous, not discrete entry + exit conditions
- Forecasts are scaled in an instrument / temporal independent way (no “magic numbers”)
- Forecast is proportional to E(Sharpe Ratio μ / σ)

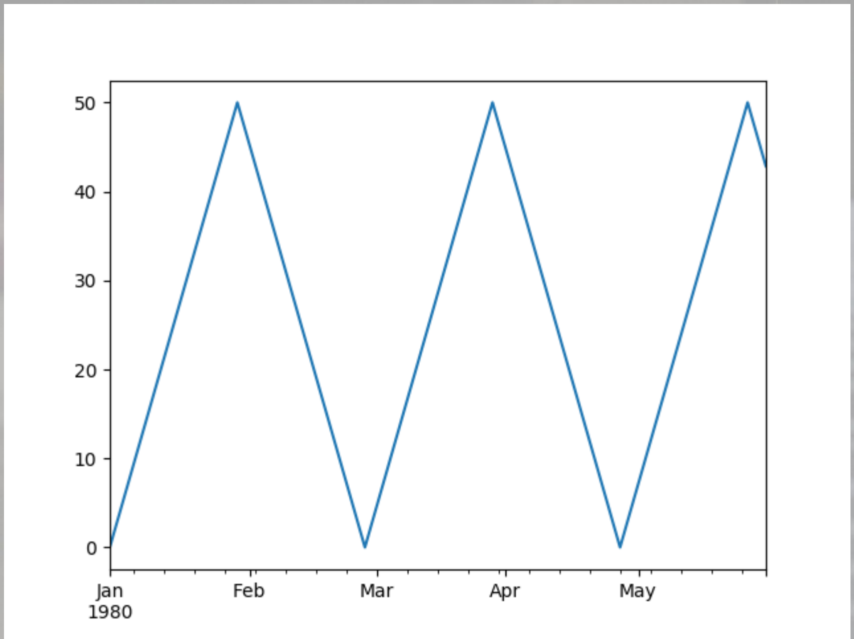
  [Position is proportional to Forecast / σ hence position is proportional to μ / σ^2]
- E(abs(forecast)) = 10.0
- In principal all forecasts used on all markets (portfolio optimisation stage will become later)
- Use multiple variations of the same trading rule to capture different time frames (as many as possible, not too highly correlated)
- Costs are the most important thing. The second most important thing is costs. Costs are predictable – returns are not. Throw away very expensive systems.
- Throw away very slow systems (LAM)

Use fake data to "fit" algo: method

What value(s) should we use for ***window_size***?

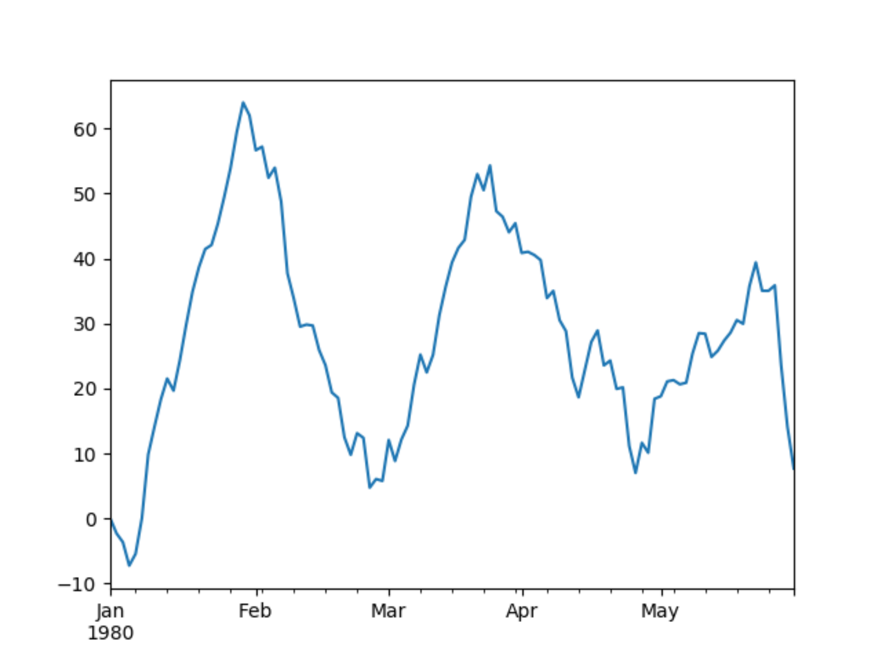
1) Get an understanding of how trend length relates to profitability of ***window_size***
2) Get an idea of how fast different ***window_size*** will trade
3) Prune any ***window_size*** that are likely to be too expensive
4) Prune any ***window_size*** that are likely to be too slow
5) Understand correlation structure to work out best ***window_size*** pattern

Use fake data to “fit” algo: Generating data



+ N(0,σ)

=



qoppac.blogspot.com/2015/11/using-random-data.html

Use fake data to “fit” algo: Trend length & window_size: pre-cost SR

		21	*64*	*128*	*192*	*256*
5	*1 week*	**6.4**	2.1	0.3	0.2	0
10	*2 weeks*	**4.5**	**2.6**	0.6	0.2	0
15	*3 weeks*	1.6	**2.9**	0.8	0.2	0
21	*1 month*	-2.0	**2.9**	**1.1**	0.2	0.1
42	*2 months*	-11	1.6	**1.4**	0.4	0.1
64	*3 months*	-0.4	-0.1	**1.1**	**0.5**	0.1
85	*4 months*	-5.0	-1.8	0.8	**0.5**	0.1
107	*5 months*	-0.1	-3.0	0.4	**0.5**	0.1
128	*6 months*		-3.0	0	**0.5**	**0.1**
150	*7 months*		-1.8	-0.5	0.3	**0.1**
171	*8 months*		-0.5	-0.7	0.3	**0.2**
192	*9 months*		-0.1	-1.0	0.1	**0.2**
213	*10 months*			-1.3	0	**0.2**
235	*11 months*			-1.6	-0.2	**0.2**
256	*12 months*			-1.6	-0.3	0.1

Window size>192 essentially pointless (LAM)

Use fake data to “fit” algo: window_size and trading speed

		Turnover/year
5	1 week	176
10	2 weeks	75
15	3 weeks	49
21	1 month	36
42	2 months	21
64	3 months	13
85	4 months	12
107	5 months	9.3
128	6 months	8.8
150	7 months	7.4
171	8 months	7.1
192	9 months	6.5
213	10 months	6.5
235	11 months	6.1
256	12 months	6.1

Turnover / year

So turnover = 52 implies holding period of one week

Barely any improvement beyond window_size>191

Use fake data to "fit" algo: window_size and costs

		Cheap eg SP500	*Expensive eg EDOLLAR*
5	1 week	17.6	176
10	2 weeks	**7.5**	75
15	3 weeks	**4.9**	49
21	1 month	**3.6**	36
42	2 months	**2.1**	21
64	3 months	**1.3**	13
85	4 months	**1.2**	**12**
107	5 months	**0.92**	**9.3**
128	6 months	**0.88**	**8.8**
150	7 months	**0.74**	**7.4**
171	8 months	**0.71**	**7.1**
192	9 months	**0.65**	**6.5**
213	10 months	**0.65**	**6.5**
235	11 months	**0.61**	**6.1**

Costs in bp /year of SR

Max allowable is 13bp

See ch.12 of my book

No point having window_size =5

Use fake data to “fit” algo: window_size and correlation structure

It turns out that if $\text{window\_size}_{n+1} = \text{window\_size}_n * \sqrt{2}$

Then $\text{correlation}(\text{forecast}_{n+1}, \text{forecast}_n) \sim 0.90$

And $\text{correlation}(\text{forecast}_{\text{another } n}, \text{forecast}_n) < 0.90$

Use fake data to “fit” algo: Final iteration

Summary of findings:

- Window size in √2 steps covers the space best
- Window size <10 too expensive for any instrument
- Window size>200 pointlessly slow

Window_size = [10,14,20,28,40,57,80, 113,160]

Should capture trends lasting for around 1 month to 18 months

Designing a trading strategy – 6 steps:

- Start with a sound framework which imposes some conditions
- Come up with the idea
- Use some random data or single scenario of real data plus theory / common sense to develop algo
- Use fake data to “fit” algo
- **Real data for parameter sensitivity check / sense check**
- Fit *allocation* using real data (out of sample, robust optimisation)

NOTE: Although I’m using real data, I’m **not** going to be looking at performance.

Real data check: consistent scaling

Window size:	10	14	20	28	40	57	80	113	160
Corn	0.14	0.17	0.20	0.22	0.26	0.29	0.32	0.37	0.42
Eurodollar	0.13	0.15	0.18	0.20	0.24	0.28	0.33	0.39	0.43
S&P 500	0.14	0.18	0.21	0.25	0.30	0.36	0.42	0.50	0.56
US 10 year bond	0.13	0.16	0.19	0.22	0.26	0.30	0.36	0.42	0.48

Real data check: turnover

window_size	turnover
10	80.4
14	55.2
20	36.9
28	25.8
40	18.4
57	13.6
80	10.7
113	8.6
160	7.0

Real data check: costs

Window size:	10	14	20	28	40	57	80	113	160
Corn	0.40	0.27	0.18	0.13	**0.09**	**0.07**	**0.05**	**0.04**	**0.03**
Eurodollar	0.64	0.44	0.29	0.20	0.14	**0.11**	**0.08**	**0.07**	**0.05**
S&P 500	**0.10**	**0.07**	**0.05**	**0.04**	**0.03**	**0.02**	**0.02**	**0.01**	**0.01**
US 10 year bond	0.25	0.17	**0.12**	**0.08**	**0.06**	**0.04**	**0.03**	**0.03**	**0.02**

Real data check: correlation structure

Highest correlation between any two pairs of window_size; 0.85

Designing a trading strategy – 6 steps:

- Start with a sound framework which imposes some conditions
- Come up with the idea
- Use some random data or single scenario of real data plus theory / common sense to develop algo
- Use fake data to “fit” algo
- Real data for parameter sensitivity check / sense check
- **Fit *allocation* using real data (out of sample, robust optimisation)**

First and last time I will use *performance* calculated using *real data.*

Conditions: reminder

- Trading rules make forecasts of risk adjusted price changes
- Forecasts are continous, not discrete entry + exit conditions
- Forecasts are scaled in an instrument / temporal independent way (no “magic numbers”)
- Forecast is proportional to E(Sharpe Ratio μ / σ)

  [Position is proportional to Forecast / σ hence position is proportional to μ / σ^2]
- E(abs(forecast)) = 10.0
- **In principal all forecasts used on all markets (portfolio optimisation stage will become later)**
- Use multiple variations of the same trading rule to capture different time frames (as many as possible, not too highly correlated)
- Costs are the most important thing. The second most important thing is costs. Costs are predictable – returns are not. Throw away very expensive systems.
- Throw away very slow systems (LAM)

Fit ***allocation*** using real data

Combined forecast = $\mathbf{w^1 f^1 + w^2 f^2 + w^3 f^3 +} \ldots$

f are in same vol scale so, values of **w** depend on:

- Pre-cost performance (different by market?)
- Costs (different by market)
- Correlation structure

- Well known portfolio optimisation problem….
- … with well known problems (estimation error, extreme weights)
- …. and well known solutions: clustering, shrinkage, bootstrapping…

- Only line of defence against incorporating a **(statistically sigificantly)** loss making trading rule in our system

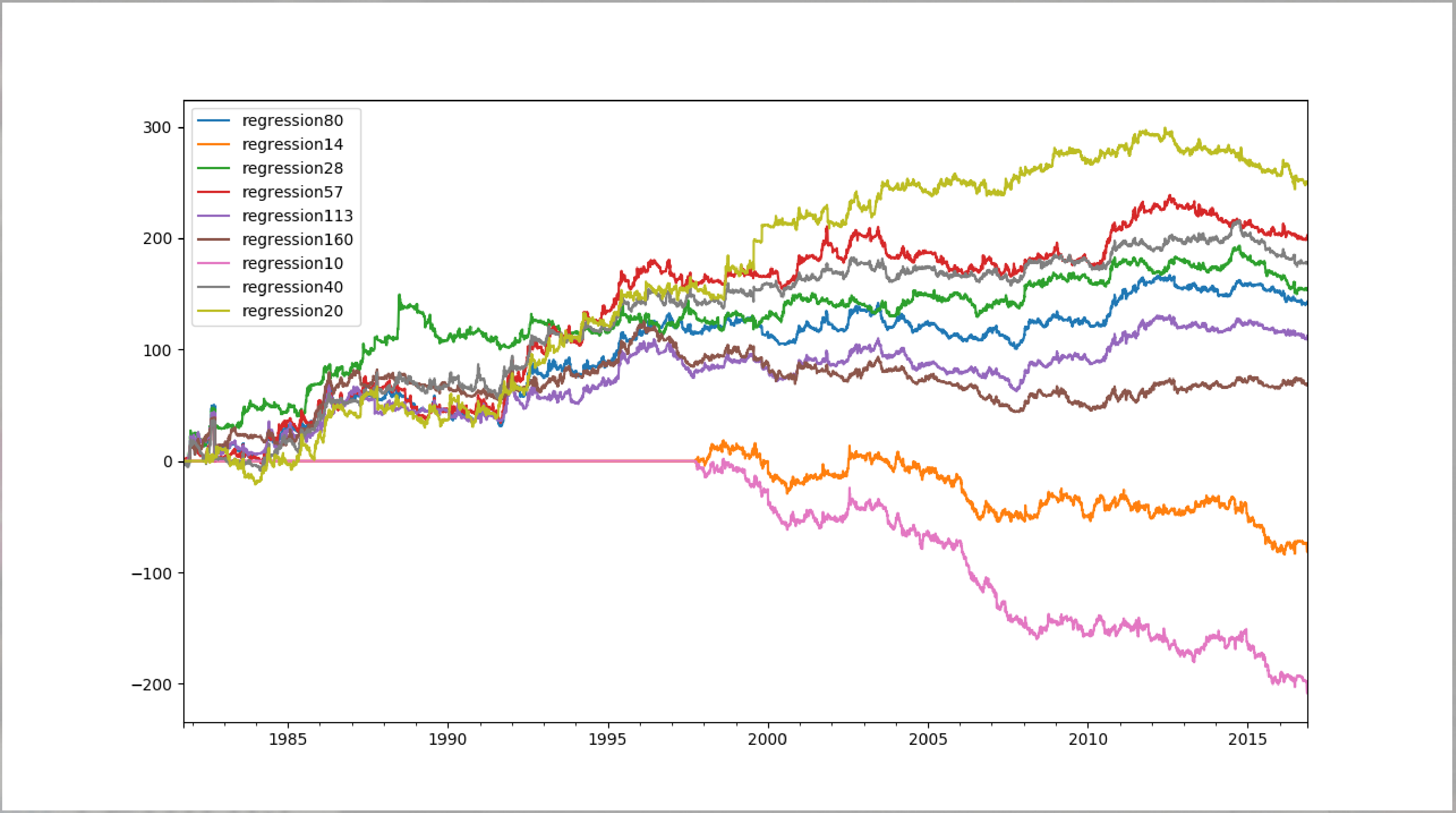
Fit ***allocation*** using real data: Hypocrisy?

An aside, Why is fitting model parameters bad...
… but optimising model portfolio allocations acceptable?

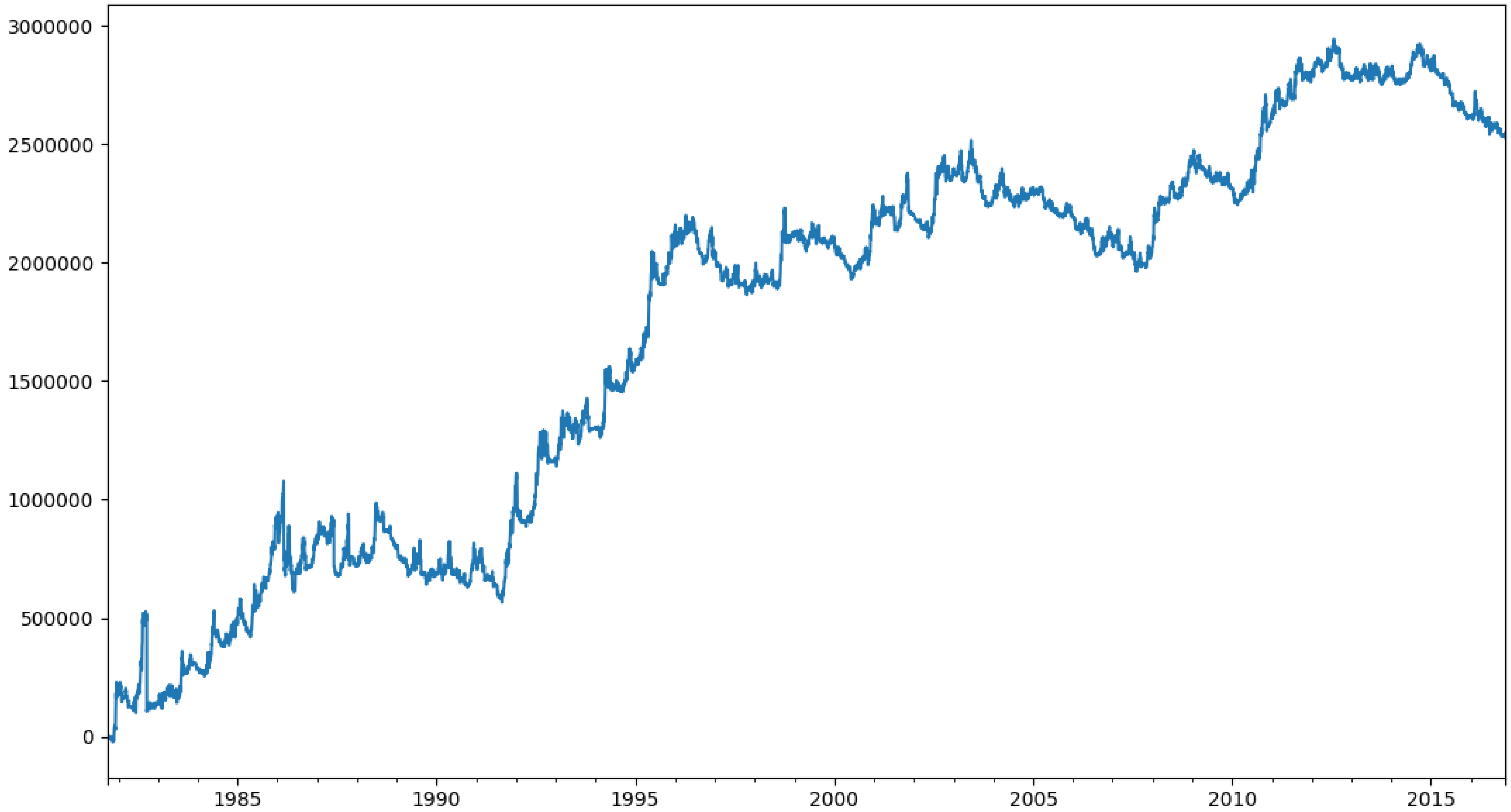
Answers:

- Parameter space much smaller
- Rolling out of sample is feasible
- Nicer surface
- Well developed techniques exist to cope with problems and use correct amount of degrees of freedom
- Much harder to do implicit fitting = much easier to resist the temptation

Fit ***allocation*** using real data: Some account curves



Fit ***allocation*** using real data: Some account curves

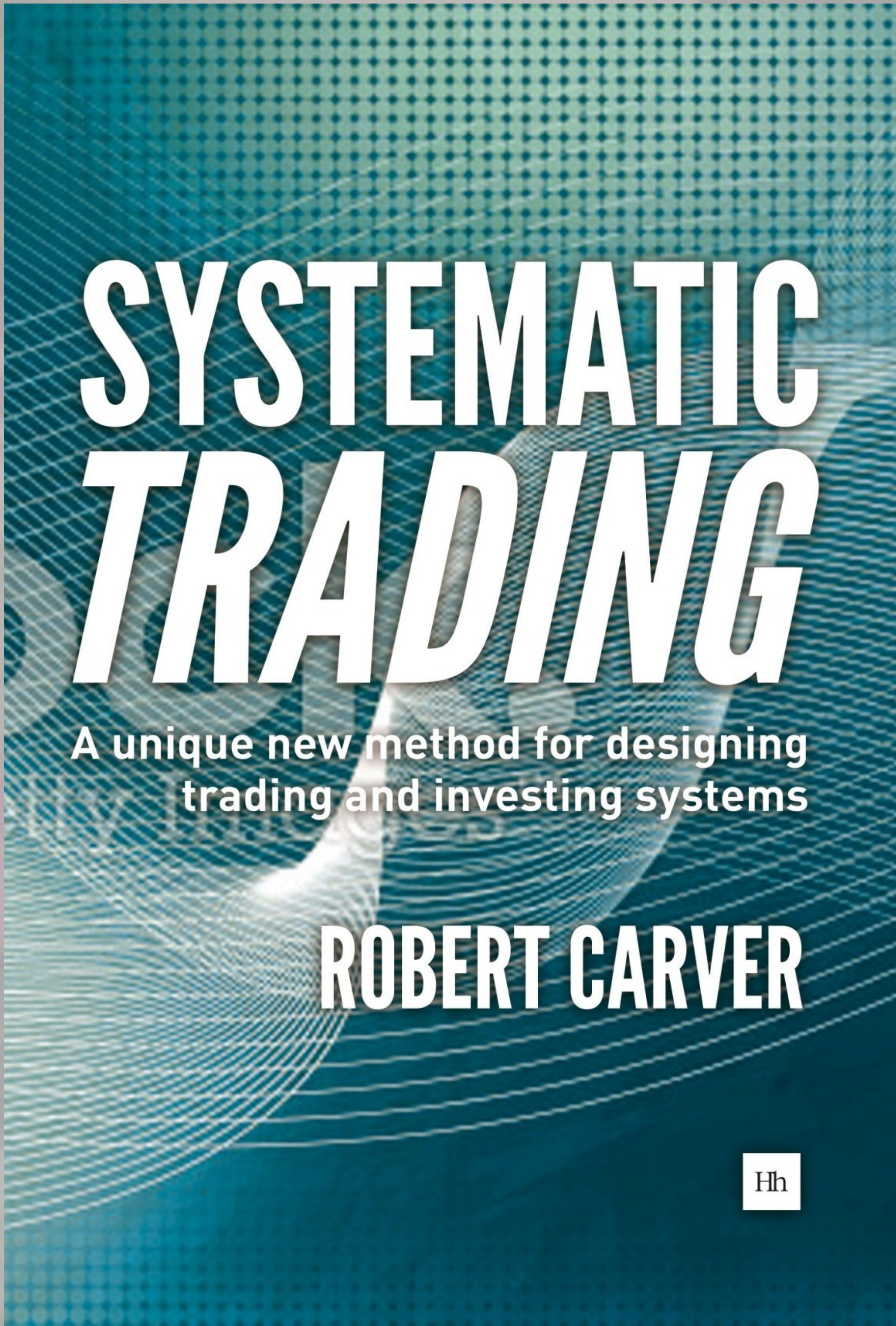


Summary

- Three types of over fitting: tacit, implicit, explicit.
- You can’t get around tacit knowledge.
- Use tacit knowledge to **design** trading strategies.

Design process:

- Start with a sound framework which imposes some conditions
- Come up with the idea
- Use some random data or single scenario of real data plus theory / common sense to develop algo
- Use fake data to “fit” algo
- Real data for parameter sensitivity check / sense check
- Fit *allocation* using real data (out of sample, robust optimisation)



My first book:
systematictrading.org

My second book:
TBC

My blog:
qoppac.blogspot.com

Some python:
github.com/robcarver17/

Twittering:
@investingidiocy