Introduction to Algorithmic Trading Strategies
Lecture 1

Overview of Algorithmic Trading

Haksun Li
haksun.li@numericalmethod.com
www.numericalmethod.com

Outline

- Definitions
- IT requirements
- Back testing
- Scientific trading models

Lecturer Profile

- Dr. Haksun Li
- CEO, Numerical Method Inc.
- (Ex-) Adjunct Professors, Advisor with the National University of Singapore, Nanyang Technological University, Fudan University, etc.
- Quantitative Trader/Analyst, BNPP, UBS
- PhD, Computer Sci, University of Michigan Ann Arbor
- M.S., Financial Mathematics, University of Chicago
- B.S., Mathematics, University of Chicago

Numerical Method Incorporated Limited

- A consulting firm in mathematical modeling, esp. quantitative trading or wealth management
- Products:
  - SuanShu
  - AlgoQuant
- Customers:
  - brokerage houses and funds all over the world
  - multinational corporations
  - very high net worth individuals
  - gambling groups
  - academic institutions

Overview

- Quantitative trading is the systematic execution of trading orders decided by quantitative market models.
- It is an arms race to build
  - more reliable and faster execution platforms (computer sciences)
  - more comprehensive and accurate prediction models (mathematics)

Market Making

- Quote to the market.
- Ensure that the portfolios respect certain risk limits, e.g., delta, position.
- Money comes mainly from client flow, e.g., bid-ask spread.
- Risk: market moves against your position holding.

Statistical Arbitrage

- Bet on the market direction, e.g., whether the price will go up or down.
- Look for repeatable patterns.
- Money comes from winning trades.
  - Risk: market moves against your position holding (guesses).

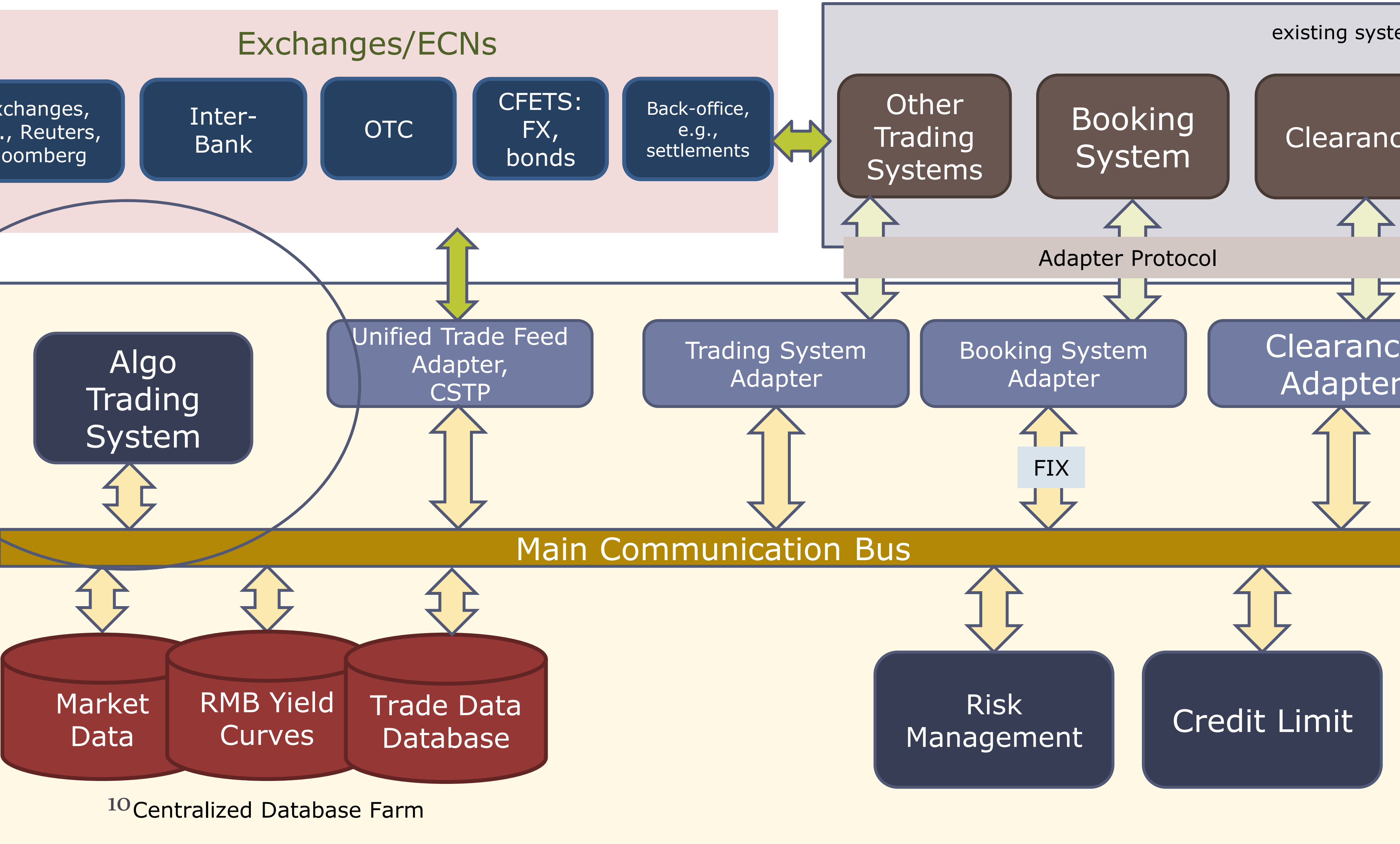
Prerequisite

- Build or buy a trading infrastructure.
  - many vendors for Gateways, APIs
  - Reuters Tibco
- Collect data, e.g., timestamps, order book history, numbers, events.
  - Reuters, EBS, TAQ, Option Metrics (implied vol),
- Clean and store the data.
  - flat file, HDF5, Vhayu, KDB, One Tick (from GS)

Trading Infrastructure

- Gateways to the exchanges and ECNs.
  - ION, ECN specific API
  - Aggregated prices
- Communication network for broadcasting and receiving information about, e.g., order book, events and order status.
- API: the interfaces between various components, e.g., strategy and database, strategy and broker, strategy and exchange, etc.

STP Trading Architecture Example



The Ideal 4-Step Research Process

- Hypothesis
  - Start with a market insight
- Modeling
  - Translate the insight in English into mathematics in Greek
- Model validation
  - Backtesting
- Analysis
  - Understand why the model is working or not

The Realistic Research Process

- Clean data
- Align time stamps
- Read Gigabytes of data
  - Retuers' EURUSD, tick-by-tick, is 1G/day
- Extract relevant information
  - PE, BM
- Handle missing data
- Incorporate events, news and announcements
- Code up the quant. strategy
- Code up the simulation
  - Bid-ask spread
  - Slippage
  - Execution assumptions
- Wait a very long time for the simulation to complete
- Recalibrate parameters and simulate again
- Wait a very long time for the simulation to complete
- Recalibrate parameters and simulate again
- Wait a very long time for the simulation to complete
- Debug
- Debug again
- Debug more
- Debug even more
- Debug patiently
- Debug impatiently
- Debug frustratingly
- Debug furiously
- Give up
- Start to trade

Research Tools – Very Primitive

- Excel
- Matlab/R/other scripting languages…
- MetaTrader/Trade Station
- RTS/other automated trading systems…

Matlab/R

- They are very slow. These scripting languages are interpreted line-by-line. They are not built for parallel computing.
- They do not handle a lot of data well. How do you handle two year worth of EUR/USD tick by tick data in Matlab/R?
- There is no modern software engineering tools built for Matlab/R. How do you know your code is correct?
- The code cannot be debugged easily. Ok. Matlab comes with a toy debugger somewhat better than gdb. It does not compare to NetBeans, Eclipse or IntelliJ IDEA.

R/scripting languages Advantages

- Most people already know it.
  - There are more people who know Java/C#/C++/C than Matlab, R, etc., combined.
- It has a huge collection of math functions for math modeling and analysis.
  - Math libraries are also available in SuanShu (Java), Nmath (C#), Boost (C++), and Netlib (C).

R Disadvantages

- TOO MANY!

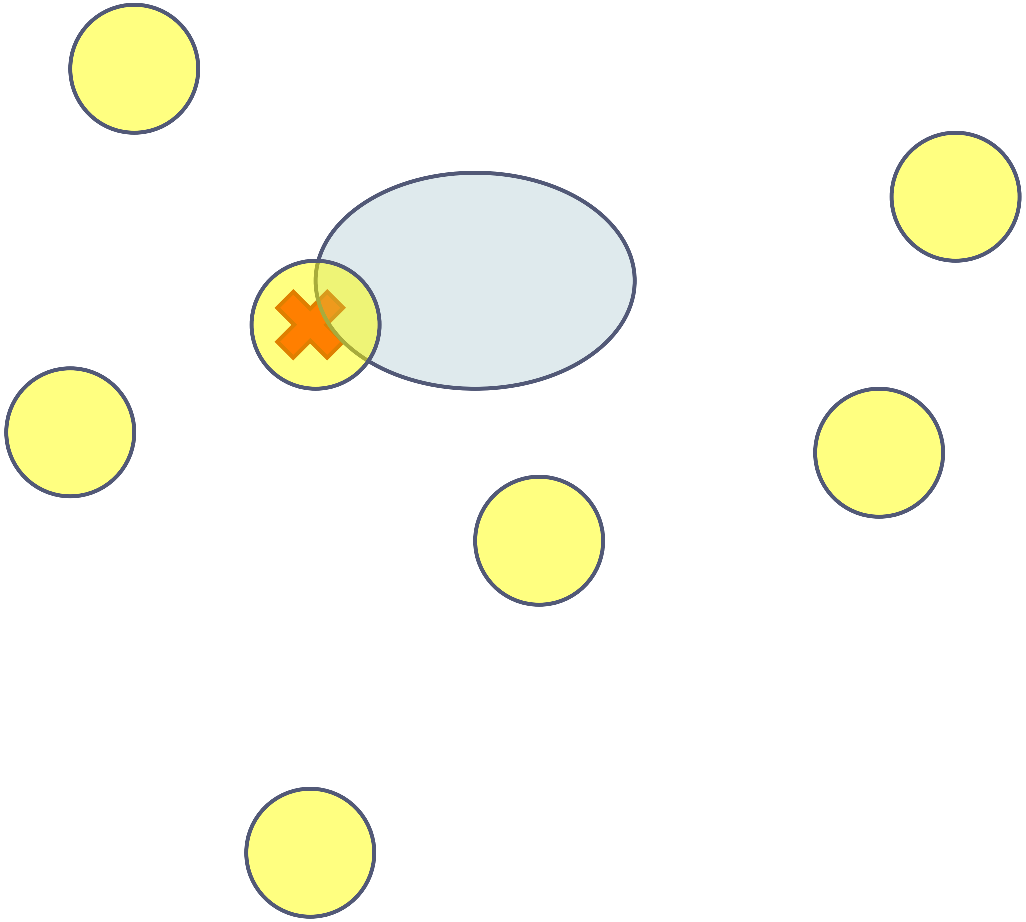
Some R Disadvantages

- Way too slow
  - Must interpret the code line-by-line
- Limited memory
  - How to read and process gigabytes of tick-by-tick data
- Limited parallelization
  - Cannot calibrate/simulate a strategy in many scenarios in parallel
- Inconvenient editing
  - No usage, rename, auto import, auto-completion
- Primitive debugging tools
  - No conditional breakpoint, disable, thread switch and resume
- Obsolete C-like language
  - No interface, inheritance; how to define $f(x)$?

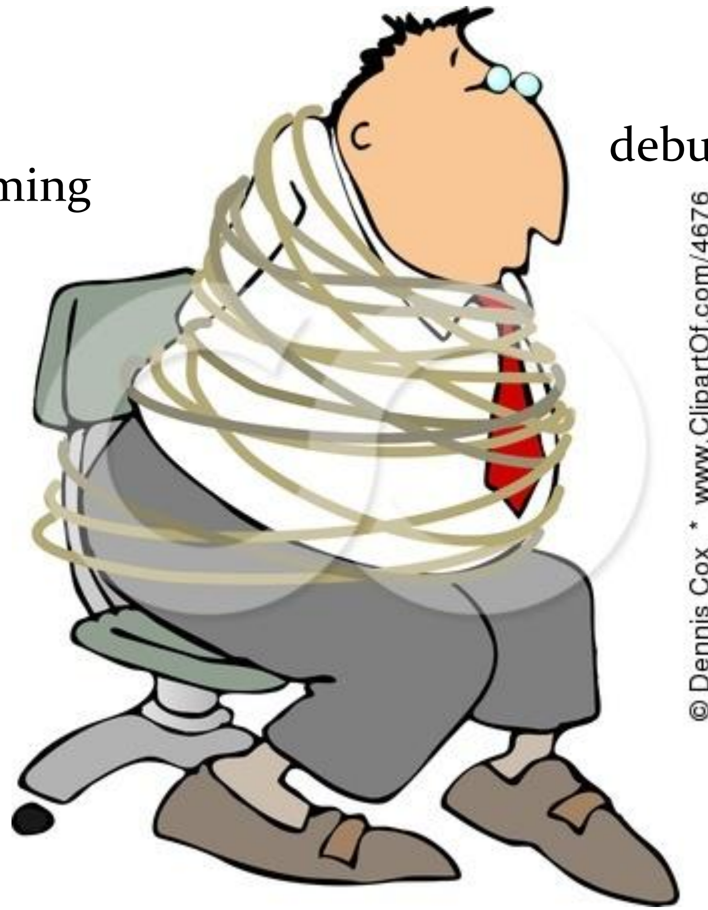
R's Biggest Disadvantage

- You cannot be sure your code is right!

Productivity



Free the Trader!



programming

debugging

calibrating

data extracting

data cleaning

waiting

backtesting

Industrial-Academic Collaboration

- Where do the building blocks of ideas come from?
  - Portfolio optimization from Prof. Lai
  - Pairs trading model from Prof. Elliott
  - Optimal trend following from Prof. Dai
  - Moving average crossover from Prof. Satchell
  - Many more……

Backtesting

- Backtesting simulates a strategy (model) using historical or fake (controlled) data.
- It gives an idea of how a strategy would work in the past.
  - It does not tell whether it will work in the future.
- It gives an objective way to measure strategy performance.
- It generates data and statistics that allow further analysis, investigation and refinement.
  - e.g., winning and losing trades, returns distribution
- It helps choose take-profit and stoploss.

A Good Backtester (1)

- allow easy strategy programming
- allow plug-and-play multiple strategies
- simulate using historical data
- simulate using fake, artificial data
- allow controlled experiments
  - e.g., bid/ask, execution assumptions, news

A Good Backtester (2)

- generate standard and user customized statistics
- have information other than prices
  - e.g., macro data, news and announcements
- Auto calibration
- Sensitivity analysis
- Quick

Iterative Refinement

- Backtesting generates a large amount of statistics and data for model analysis.
- We may improve the model by
  - regress the winning/losing trades with factors
  - identify, delete/add (in)significant factors
  - check serial correlation among returns
  - check model correlations
  - the list goes on and on……

Some Performance Statistics

- pnl
- mean, stdev, corr
- Sharpe ratio
- confidence intervals
- max drawdown
- breakeven ratio
- biggest winner/loser
- breakeven bid/ask
- slippage

Omega

- $\Omega(L) = \frac{\int_L^b [1-F(x)]dx}{\int_L^b [F(x)]dx} = \frac{C(L)}{P(L)}$
- The higher the ratio; the better.
- This is the ratio of the probability of having a gain to the probability of having a loss.
- Do not assume normality.
- Use the whole returns distribution.

Bootstrapping

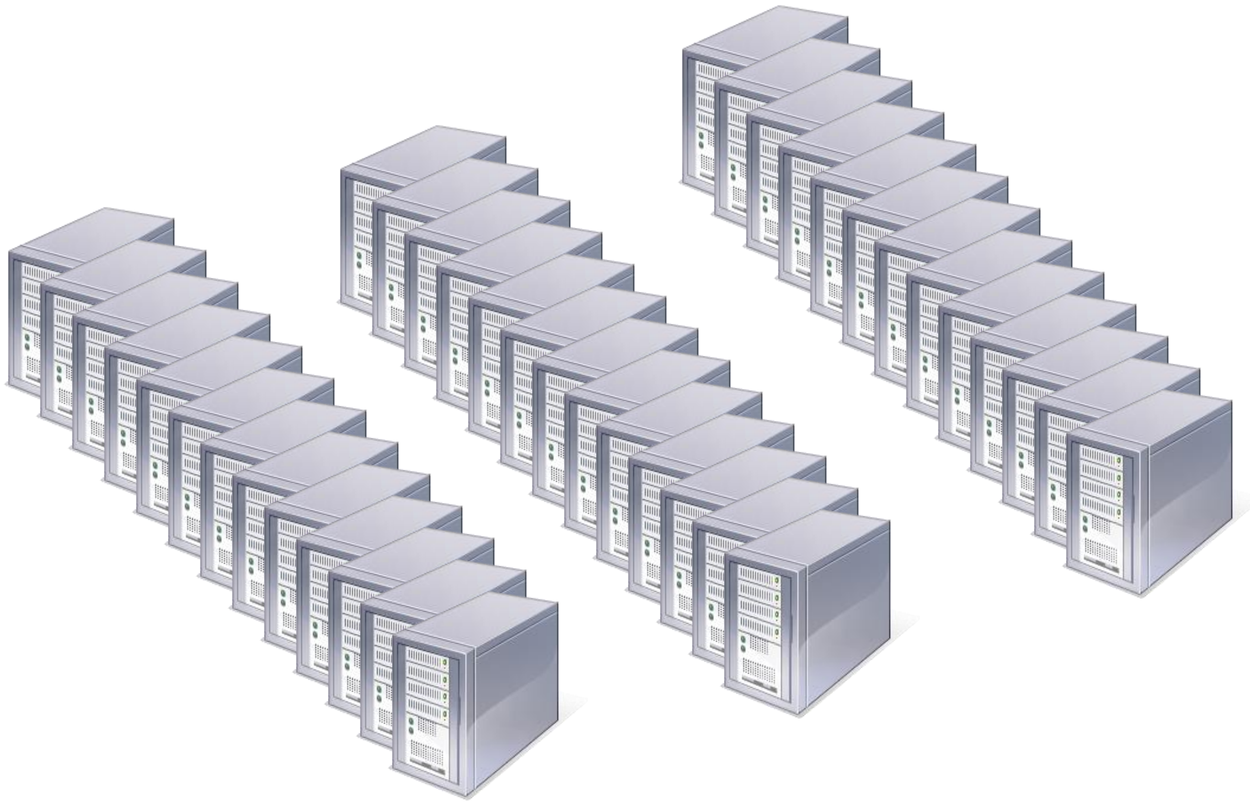
- We observe only one history.
- What if the world had evolve different?
- Simulate “similar” histories to get confidence interval.
- White's reality check (White, H. 2000).

Calibration

- Most strategies require calibration to update parameters for the current trading regime.
- Occam's razor: the fewer parameters the better.
- For strategies that take parameters from the Real line: Nelder-Mead, BFGS
- For strategies that take integers: Mixed-integer non-linear programming (branch-and-bound, outer-approximation)

Global Optimization Methods

$$f$$



Sensitivity

- How much does the performance change for a small change in parameters?
- Avoid the optimized parameters merely being statistical artifacts.
- A plot of measure vs. d(parameter) is a good visual aid to determine robustness.
- We look for plateaus.

Summary

- Algo trading is a rare field in quantitative finance where computer sciences is at least as important as mathematics, if not more.
- Algo trading is a very competitive field in which technology is a decisive factor.

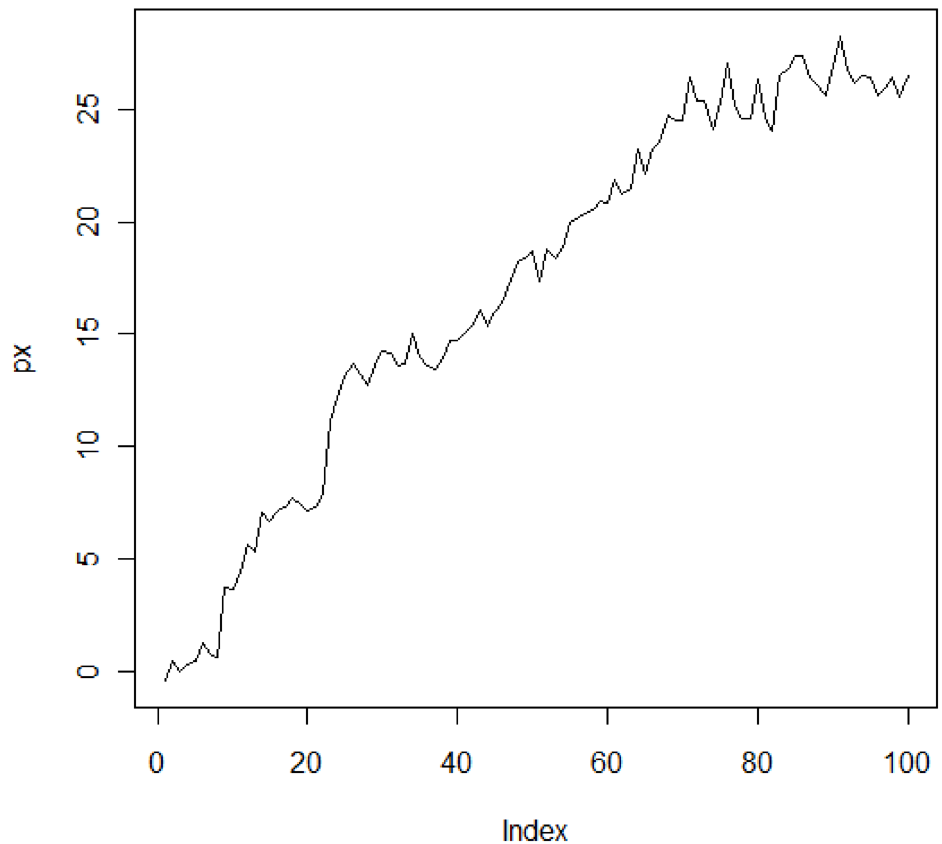
Scientific Trading Models

- Scientific trading models are supported by logical arguments.
  - can list out assumptions
  - can quantify models from assumptions
  - can deduce properties from models
  - can test properties
  - can do iterative improvements

Superstition

- Many “quantitative” models are just superstitions supported by fallacies and wishful-thinking.

Let's Play a Game



Impostor Quant. Trader

- Decide that this is a bull market
  - by drawing a line
  - by (spurious) linear regression
- Conclude that
  - the slope is positive
  - the t-stat is significant
- Long
- Take profit at 2 upper sigmas
- Stop-loss at 2 lower sigmas

Reality

- r = rnorm(100)
- px = cumsum(r)
- plot(px, type='l')

Mistakes

- Data snooping
- Inappropriate use of mathematics
  - assumptions of linear regression
    - linearity
    - homoscedasticity
    - independence
    - normality
- Ad-hoc take profit and stop-loss
  - why 2?
- How do you know when the model is invalidated?

Extensions of a Wrong Model

- Some traders elaborate on this idea by
  - using a moving calibration window (e.g., Bands)
  - using various sorts of moving averages (e.g., MA, WMA, EWMA)

Fake Quantitative Models

- Data snooping
- Misuse of mathematics
- Assumptions cannot be quantified
- No model validation against the current regime
- Ad-hoc take profit and stop-loss
  - why 2?
- How do you know when the model is invalidated?
- Cannot explain winning and losing trades
- Cannot be analyzed (systematically)

A Scientific Approach

- Start with a market insight (hypothesis)
  - hopefully without peeking at the data
- Translate English into mathematics
  - write down the idea in math formulae
- In-sample calibration; out-sample backtesting
- Understand why the model is working or not
  - in terms of model parameters
  - e.g., unstable parameters, small p-values

MANY Mathematical Tools Available

- Markov model
- co-integration
- stationarity
- hypothesis testing
- bootstrapping
- signal processing, e.g., Kalman filter
- returns distribution after news/shocks
- time series modeling
- The list goes on and on……

A Sample Trading Idea

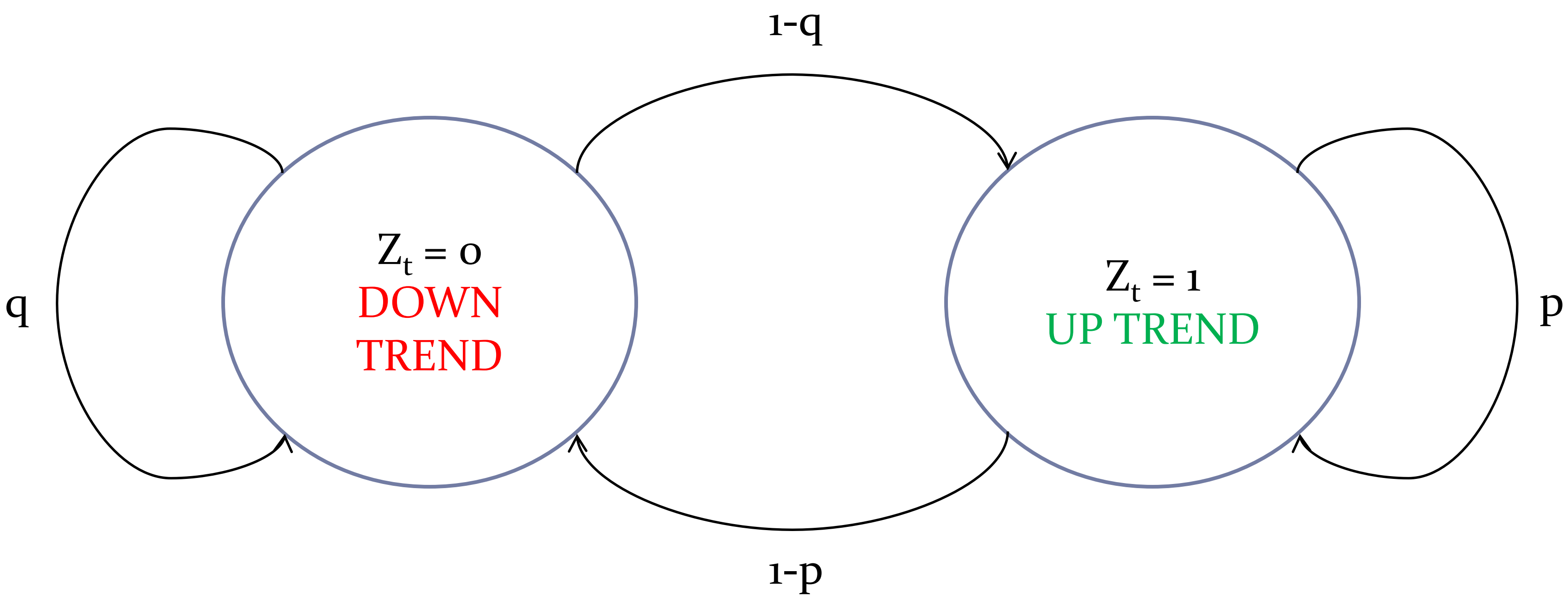
- When the price trends up, we buy.
- When the price trends down, we sell.

What is a Trend?

An Upward Trend

- More positive returns than negative ones.
- Positive returns are persistent.

Knight-Satchell-Tran Z_t



Knight-Satchell-Tran Process

- $R_t = \mu_l + Z_t\varepsilon_t - (1 - Z_t)\delta_t$
  - μ_l: long term mean of returns, e.g., 0
  - ε_t, δ_t: positive and negative shocks, non-negative, i.i.d
- $f_\varepsilon(x) = \frac{{\lambda_1}^{\alpha_1} x^{\alpha_1 - 1}}{\Gamma(\alpha_1)} e^{-\lambda_1 x}$
- $f_\delta(x) = \frac{{\lambda_2}^{\alpha_2} x^{\alpha_2 - 1}}{\Gamma(\alpha_2)} e^{-\lambda_2 x}$

What Signal Do We Use?

- Let's try Moving Average Crossover.

Moving Average Crossover

- Two moving averages: slow (n) and fast (m).
- Monitor the crossovers.
- $B_t = \left(\frac{1}{m}\sum_{j=0}^{m-1} P_{t-j}\right) - \left(\frac{1}{n}\sum_{j=0}^{n-1} P_{t-j}\right), n > m$
- Long when $B_t \geq 0$.
- Short when $B_t < 0$.

How to choose n and m?

- For most traders, it is an art (guess), not a science.
- Let's make our life easier by fixing $m = 1$.
  - Why?

What is n?

- $n = 2$
- $n = \infty$

Expected P&L

- GMA(2,1)
  - $\mathrm{E}(RR_T) = \frac{1}{1-p}\{\Pi p\mu_\varepsilon - (1-p)\mu_\delta\}$
- GMA(∞)
  - $\mathrm{E}(RR_T) = -[1 - p(1 - \Pi)][\mu_\varepsilon + \mu_\delta]$

Model Benefits (1)

- It makes “predictions” about which regime we are now in.
- We quantify how useful the model is by
  - the parameter sensitivity
  - the duration we stay in each regime
  - the state differentiation power

Model Benefits (2)

- We can explain winning and losing trades.
  - Is it because of calibration?
  - Is it because of state prediction?
- We can deduce the model properties.
  - Are 3 states sufficient?
  - prediction variance?
- We can justify take profit and stoploss based on trader utility function.

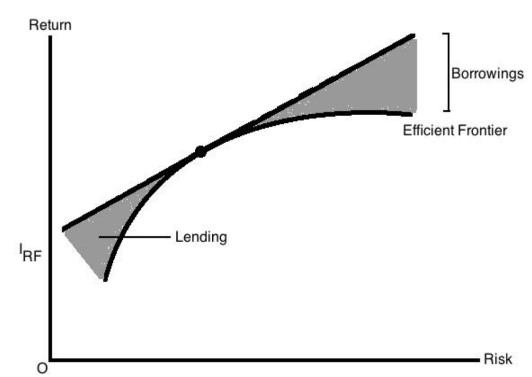
Limitations

- Assumptions are not realistic.
  - Classical example: Markowitz portfolio optimization
  - http://www.numericalmethod.com:8080/nmj2ee-war/faces/webdemo/markowitz.xhtml
- Regime change.
- IT problems.
- Bad luck!
  - Variance

Markowitz's Portfolio Selection

- For a portfolio of m assets:
  - expected returns of asset i = μ_i
  - weight of asset i = w_i such that $\sum_i^m w_i = 1$
- Given a target return of the portfolio μ_*, the optimal weighting w_{eff} is given by

$$w_{eff} = \arg\min_w w^T \Sigma w \text{ subject to } w^T \mu = \mu_*, w^T 1 = 1, w \geq 0$$



Stochastic Optimization Approach

- Consider the more fundamental problem:
  - Given the past returns $r_1, \dots, r_n$
    $$\max\{E(w^T r_{n+1}) - \lambda Var(w^T r_{n+1})\}$$
  - λ is regarded as a ***risk-aversion index*** (user input)
- Instead, solve an equivalent stochastic optimization problem

$$\max_{n}\{E[w^T(\eta)r_{n+1} - \lambda Var[w^T(\eta)r_{n+1}]\}$$

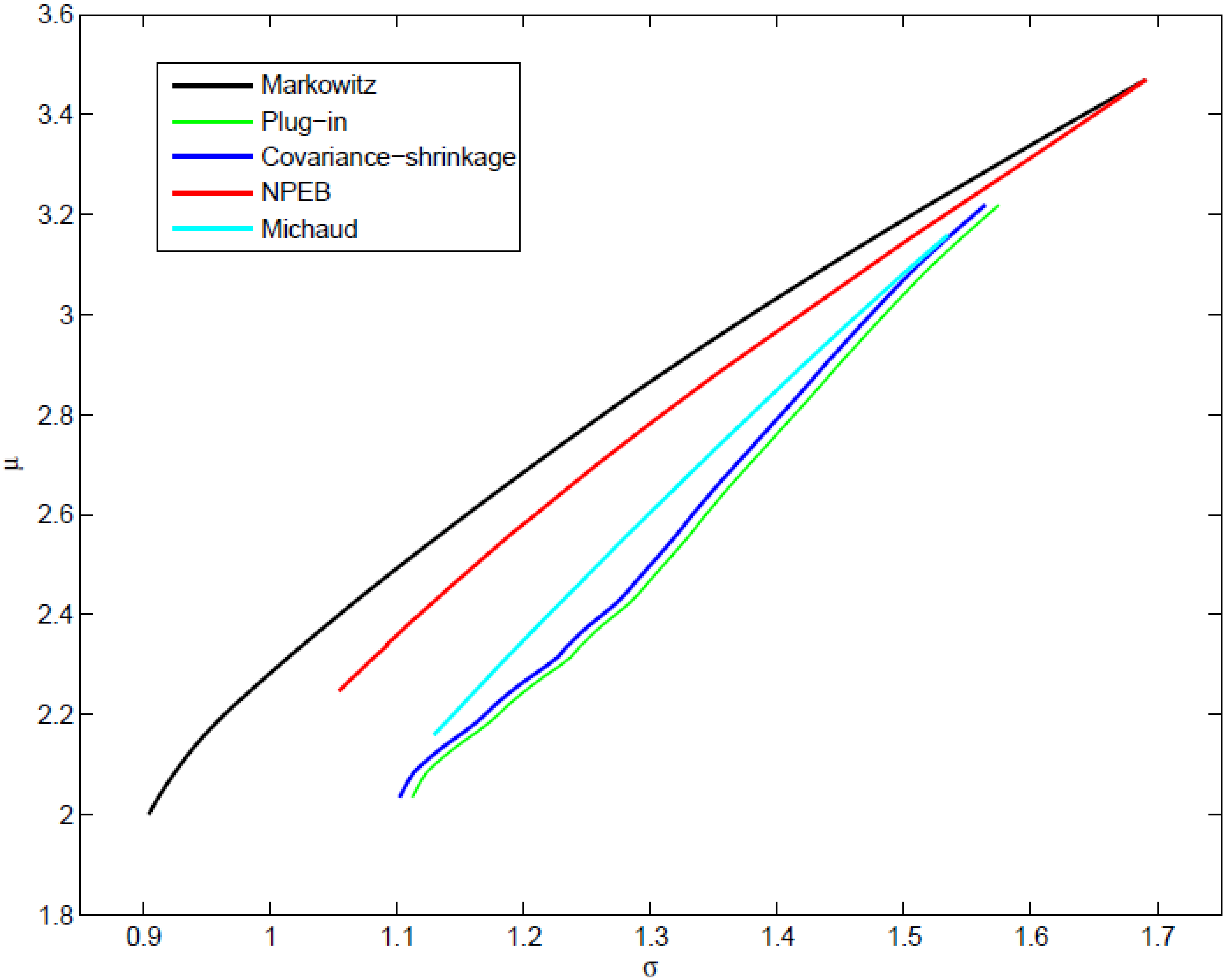
where

$$w(\eta) = \arg\min_{w}\{\lambda E[(w^T r_{n+1})^2] - \eta E(w^T r_{n+1})\}$$

and

$$\eta = 1 + 2\lambda E(W_B)$$

Mean-Variance Portfolio Optimization when Means and Covariances are Unknown



Summary

- Market understanding gives you an intuition to a trading strategy.
- Mathematics is the tool that makes your intuition concrete and precise.
- Programming is the skill that turns ideas and equations into reality.

AlgoQuant Demo