Algorithmic Trading as a Science
Speaker Profile

- Haksun Li, **Numerical Method Inc.**
- Quantitative Trader
- Quantitative Analyst
- PhD, Computer Science, University of Michigan Ann Arbor
- M.S., Financial Mathematics, University of Chicago
- B.S., Mathematics, University of Chicago
Definition

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

- Assumptions cannot be quantified
- No model validation against the current regime
- Cannot explain winning and losing trades
- *Cannot be analyzed (systematically)***
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)
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 models work or fail
  - 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 $Z_t$

$Z_t = 0$
DOWN TREND

$Z_t = 1$
UP TREND

$q$

$p$

$1-q$

$1-p$
Knight-Satchell-Tran Process

\[ R_t = \mu_l + Z_t \varepsilon_t - (1 - Z_t) \delta_t \]

- \( \mu_l \): long term mean of returns, e.g., 0
- \( \varepsilon_t, \delta_t \): positive and negative shocks, non-negative, i.i.d

\[ f_\varepsilon(x) = \frac{\lambda_1 \alpha_1 x^{\alpha_1 - 1}}{\Gamma(\alpha_1)} e^{-\lambda_1 x} \]
\[ f_\delta(x) = \frac{\lambda_2 \alpha_2 x^{\alpha_2 - 1}}{\Gamma(\alpha_2)} e^{-\lambda_2 x} \]
How Signal Do We Use?

- Let’s try Moving Average Crossover.
Moving Average Crossover

- Two moving averages: slow \( (n) \) and fast \( (m) \).
- Monitor the crossovers.

\[
B_t = \left( \frac{1}{m} \sum_{j=0}^{m-1} P_{t-j} \right) - \left( \frac{1}{n} \sum_{j=0}^{n-1} P_{t-j} \right), \quad n > m
\]

- 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?
GMA(n, 1)

- $B_t \geq 0$ iff $P_t \geq \left( \prod_{j=0}^{n-1} P_{t-j} \right)^{\frac{1}{n}}$
  
  - $R_t \geq - \sum_{j=1}^{n-2} \frac{n-(j+1)}{n-1} R_{t-j}$ (by taking log)

- $B_t < 0$ iff $P_t < \left( \prod_{j=0}^{n-1} P_{t-j} \right)^{\frac{1}{n}}$
  
  - $R_t < - \sum_{j=1}^{n-2} \frac{n-(j+1)}{n-1} R_{t-j}$ (by taking log)
What is $n$?

- $n = 2$
- $n = \infty$
Assume the long term mean is 0, $\mu_l = 0$.

- $(B_t \geq 0) \equiv (R_t \geq 0) \equiv (Z_t = 1)$
- $(B_t < 0) \equiv (R_t < 0) \equiv (Z_t = 0)$
Naïve MA Trading Rule

- Buy when the asset return in the present period is positive.
- Sell when the asset return in the present period is negative.
How Much Money Will I Make?

- $T$ Period Return:
  - $RR_T = \sum_{t=1}^{T} R_t \times I_{\{B_{t-1} \geq 0\}}$

\[\begin{array}{c}
\text{hold} \\
0 \downarrow \\
1 \downarrow \\
\vdots \downarrow \\
B_T < 0 \downarrow \\
T \downarrow \\
\text{Sell at this time point}
\end{array}\]
Expected Holding Time

- $P(N = T)$
  - $= P(B_T < 0, B_{T-1} \geq 0, \ldots, B_1 \geq 0, B_0 \geq 0)$
  - $= P(Z_T = 0, Z_{T-1} = 1, \ldots, Z_1 = 1, Z_0 = 1)$
  - $= P(Z_T = 0, Z_{T-1} = 1, \ldots, Z_1 = 1|Z_0 = 1)P(Z_0 = 1)$
  - $= \left\{ \begin{array}{ll}
  \Pi \rho^{T-1} (1 - p), & T \geq 1 \\
  1 - \Pi, & T = 0 
  \end{array} \right.$

- Stationary probabilities
  - $\Pi = \frac{1 - q}{2 - p - q}$
My Returns Distribution (1)

- \( \Phi_{RRT|N=T}(s) \)
- \( = \mathbb{E} \left[ e^{\left\{ i \left[ \sum_{t=1}^{T} R_t \times I\{B_{t-1} \geq 0\} \right] s \right\}} \bigg| N = T \right] \)
- \( = \mathbb{E} \left[ e^{\left\{ i \left[ \sum_{t=1}^{T} R_t \times I\{B_{t-1} \geq 0\} \right] s \right\}} \bigg| B_T < 0, B_{T-1} \geq 0, \ldots, B_0 \geq 0 \right] \)
- \( = \mathbb{E} \left[ e^{\left\{ i \left[ \sum_{t=1}^{T} R_t \right] s \right\}} \bigg| Z_T = 0, Z_{T-1} = 1, \ldots, Z_1 = 1 \right] \)
- \( = \mathbb{E} \left[ e^{\{i[\varepsilon_1 + \cdots + \varepsilon_{T-1} - \delta_T]s\}} \right] \)
- \( = \begin{cases} \Phi_{\varepsilon}^{T-1}(s) \Phi_{\delta}(-s), & T \geq 1 \\ \Phi_{\delta}(-s), & T = 0 \end{cases} \)
My Returns Distribution (2)

\[ \Phi_{RR_T}(s) = \]

\[ \sum_{T=0}^{\infty} E \left[ e^{i \left[ \Sigma_{t=1}^{T} R_t \times I_{\{B_{t-1} \geq 0}\} \right]} \right| N = T \right] P(N = T) \]

\[ = \sum_{T=1}^{\infty} \Pi p^{T-1}(1 - p) \Phi_{\epsilon}^{T-1}(s) \Phi_{\delta}(-s) + (1 - \Pi) \Phi_{\delta}(-s) \]

\[ = (1 - \Pi) \Phi_{\delta}(-s) + \Pi(1 - p) \frac{\Phi_{\delta}(-s)}{1 - p \Phi_{\epsilon}(s)} \]
Expected P&L

\[ E(\mathcal{R}_T) = -i \Phi_{\mathcal{R}_T}'(0) \]

\[ = \frac{1}{1-p} \{ \Pi p \mu_\epsilon - (1-p)\mu_\delta \} \]
When Will My Strategy Make Money?

- The expected return is positive when
  - $\mu_\epsilon \geq \frac{1-p}{\prod p} \mu_\delta$, shock impact
  - $\mu_\epsilon \gg \mu_\delta$, shock impact
  - $\prod p \geq 1 - p$, if $\mu_\epsilon \approx \mu_\delta$, persistence
What About GMA(∞, 1)

- Repeat the steps above.
- $E(RR_T) = -[1 - p(1 - \Pi)][\mu_\varepsilon + \mu_\delta]$
When Will GMA(∞,1) Make Money?
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 2 states sufficient?
  - prediction variance?
- We can justify take-profit and stop-loss based on trader utility function.
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 stop-loss.
Some Performance Statistics

- p&l
- mean, stdev, corr
- Sharpe ratio
- confidence intervals
- max drawdown
- breakeven ratio
- biggest winner/loser
- breakeven bid/ask
- slippage
Omega

\[ \Omega(L) = \frac{\int_{L}^{b}[1-F(x)]dx}{\int_{a}^{L}F(x)dx} = \frac{C(L)}{P(L)} \]

- The higher the ratio, the better
- Ratio of the probability of having a gain by the probability of having a loss
- Do not assume Normality
- Use the whole returns distribution
Performance on MSCI Singapore
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).
Fake Data
Returns: AR(1)

\[ X_t = \alpha X_{t-1} + \varepsilon_t \]

- Auto-correlation is required to be profitable.
- The smaller the order, the better. (quicker response)
Returns: AR(1)
Returns: ARMA(1, 1)

- \((X_t - \mu) - p(X_{t-1} - \mu) = \epsilon_t - q\epsilon_{t-1}\)
- Prices tend to move in one direction (trend) for a period of time and then change in a random and unpredictable fashion.
Returns: ARMA(1, 1)

Yearly Expected Rule Returns
Price-trend model without drift

- Momentum
- SMAV
- WMAV

Optimal order

No systematic winner
Returns: ARIMA(o, d, o)

- $\nabla^d (X_t - \mu) = e_t$
- Irregular, erratic, aperiodic cycles.
Returns: ARIMA(0, d, 0)
ARCH + GARCH

- The presence of conditional heteroskedasticity, if unrelated to serial dependencies, may be neither a source of profits nor losses for linear rules.
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
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.
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
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. \( d(\text{parameter}) \) is a good visual aid to determine robustness.
- We look for plateaus.
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......
Implementation

- Connectivity to exchanges
  - e.g., ION, RTS
- Platform dependent APIs
- Programming languages
  - Java, C++, C#, VBA, Matlab
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.