THE MACHINE DESIGN OF TRADING SYSTEMS
YOUR PRESENTER: MIKE BARNA

- Founder and President, Trading System Lab
- Sr.VP Regency Stocks and Commodities Fund, LP, LLC, QEP, CPO, CTA
- BS Mathematics, Arizona State University
- MS Astronautical and Aeronautical Engineering, Stanford University
- Systems Authored or Co-Authored: TSL MACHINE GENERATED SYSTEMS, R-MESA, BIGBLUE, MESA BONDS, MESA NOTES, SIERRA HOTEL
- Former Defense Industry Rocket-Ramjet, Laser and Guidance Engineer
- Star Wars Research and Development Management Engineer
- Series 3, Series 30
- 12 FAA pilot certificates or ratings

Contact: www.tradingsystemlab.com    mike@tradingsystemlab.com
OUR TEAM

• Mike Barna: Trading System Lab-Silicon Valley Based trading research and development company with a team of international and domestic programmers, third party developers and testers. Developed the First Commercially available Machine Designed Trading Systems Platform that requires no programming from the user.
  www.tradingsystemlab.com

• Frank Francone: Register Machine Learning, Inc.-US Based company with a team of international and domestic machine learning scientists, IP attorneys, statisticians and programmers. Involved in government contracts. Produces the LAIMGP licensed exclusively to TSL. Authored the leading University Textbook on GP. 1600 citations.
  www.rmltech.com
WHAT IS TSL?

- TSL is a platform for the Machine Design of Trading Strategies
- Linear Automatic Induction of Machine Code with Genetic Programming (LAIMGP) (algorithms are nonlinear)
- Code is exported in different languages
- Strategies are tested OOS “during” design
- Patented and Trademarked
- Single Market Systems: HFT, MFT, LFT
- Daytrading
- Pairs
- Portfolios
- Options
OUR PRODUCT:
The TSL Platform

- Unlimited orthogonal return streams
- Use any data: No Programming Required
- Any time frame
- Very fast
- Code is exported in different languages
- Anti-curve fitted and pre-tested OOS “during” design
- Any trading tactic: Pairs, Portfolios, Options, Daytrading, HFT
TSL CLIENTS AND TRADERS
TSL’s JOB IS TO PROVIDE TSL TO CLIENT TRADERS

• Major Wall Street Investment Bank >$100M
• Small and Mid size CTA’s: $10M-$100M
• Proprietary Trading Firms: $5M-$50M
• Individual Traders < $5M
• International Traders and Funds
• Strategy Development Engineers
• Beginner to PhD
HYPOTHETICAL PERFORMANCE RESULTS HAVE MANY INHERENT LIMITATIONS, SOME OF WHICH ARE DESCRIBED BELOW. NO REPRESENTATION IS BEING MADE THAT ANY ACCOUNT WILL OR IS LIKELY TO ACHIEVE PROFITS OR LOSSES SIMILAR TO THOSE SHOWN.

IN FACT, THERE ARE FREQUENTLY SHARP DIFFERENCES BETWEEN HYPOTHETICAL PERFORMANCE RESULTS AND THE ACTUAL RESULTS ACHIEVED BY ANY PARTICULAR TRADING PROGRAM. ONE OF THE LIMITATIONS OF HYPOTHETICAL PERFORMANCE RESULTS IS THAT THEY ARE GENERALLY PREPARED WITH THE BENEFIT OF HINDSIGHT. IN ADDITION, HYPOTHETICAL TRADING DOES NOT INVOLVE FINANCIAL RISK, AND NO HYPOTHETICAL TRADING RECORD CAN COMPLETELY ACCOUNT FOR THE IMPACT OF FINANCIAL RISK IN ACTUAL TRADING. FOR EXAMPLE, THE ABILITY TO WITHSTAND LOSSES OR TO ADHERE TO A PARTICULAR TRADING PROGRAM IN SPITE OF TRADING LOSSES ARE MATERIAL POINTS WHICH CAN ALSO ADVERSELY AFFECT ACTUAL TRADING RESULTS.

THERE ARE NUMEROUS OTHER FACTORS RELATED TO THE MARKETS IN GENERAL OR TO THE IMPLEMENTATION OF ANY SPECIFIC TRADING PROGRAM WHICH CANNOT BE FULLY ACCOUNTED FOR IN THE PREPARATION OF HYPOTHETICAL PERFORMANCE RESULTS AND ALL OF WHICH CAN ADVERSELY AFFECT ACTUAL TRADING RESULTS.
Ballmer says machine learning will be the next era of computer science

Former Microsoft CEO Steve Ballmer. Credit: Reuters/2013 file photo

Former Microsoft CEO makes donation to expand Harvard's computer science department

By Sharon Gaudin  FOLLOW  
Computerworld  |  Nov 13, 2014 4:02 AM PT
MACHINES “CRUSHED” HUMAN RIVALS

Hedge fund robots crushed human rivals in 2014

Lawrence Delevingne | @ldelevingne
Monday, Jan 5, 2015 10:43 AM ET

NetNet

Adam Jeffery | CNBC
David Winton Harding, founder and president of Winton Capital Management
WHAT IS A TRADING SYSTEM EQUITY CURVE?

MACHINE PRODUCED OR HUMAN DESIGNED <ALGORITHMS>

S&P Futures 1982-2013

Release Date
1996 knee
SAMPLE COUNTERTRENDING TRADING SYSTEM

Weak Equity Curve

inputs: Length1(10), threshold1(0.2), threshold2(0.2);
vars: atr(0);
atr = average(truerange, length1);
if marketposition=0 then buy next bar at low - threshold1*atr limit;
if marketposition>0 then sell next bar at high + threshold2*atr limit;
HUMAN DESIGNED STRATEGIES

COST

TIME

TECHNICAL
MACHINE DESIGNED STRATEGIES

LOWER COST PER STRATEGY
REDUCED TIME TO IMPLEMENTATION
MANY TECHNICAL CAPABILITIES
• Brokerage/Software companies do not or cannot provide you with the most important item you need to be successful in the markets with trading systems:
  Robust Trading Strategies that are easy to create

• TSL is interested only in:
  Robust Trading Strategies that are easy to create
WHAT IS THE PROBLEM TO BE SOLVED?

The problem to be solved is how to construct a Machine that automatically writes better Systems faster and less expensively than humans can create using manual techniques.
SYSTEMATIC VERSES DISCRETIONARY CTA VAMI
1987-2015

MECHANICAL SYSTEMS

DISCRETIONARY

457 programs, $296B

137 programs, $18B

Source: BarclayHedge
SYSTEMATIC VERSES DISCRETIONARY
CTA MUM, $B 1999 to 2014

Systematic
(At least 95% Systematic)

Discretionary
(At least 65% Discretionary or Judgmental)

Source: BarclayHedge

Trading System Lab®
TYPES OF SYSTEMATIC TRADING SYSTEM DESIGN

• If-Then Human (manual) constructs
• Predictive Modeling (ML)
• Self-Evolving Strategy Structure (ML-TSL)

  1. Meta-Heuristic Simulation Based
  2. Supervised Learning-No Supervisory Signal
  3. Single and Multi-Objective
  4. Reinforcement Learning

PREDICTION APPROACH

DATA → MODEL → PREDICTION → EXPECTATION

NOT GOOD ENOUGH?

EXPECTATION APPROACH

DATA → EXPECTATION → MODEL

GOOD ENOUGH?

IMPLEMENT
PREDICTION VERSES EXPECTATION

• In the Prediction Approach, Prices or Volatility are forecasted n-steps out. Equity Curves are then generated as an additional step.

  Good Trading Systems may exhibit poor $R^2$
  Standard prediction models may require further work to generate good Equity Curves.

• In the Expectation Approach, Systems are viewed as Objects and Equity Curves are generated and improved through System Metric Targeting and Objective Function Optimization.

  Good Trading System Equity Curves are thus directly evolved, leveraging the cross metric effect characteristic of Trading Systems and eliminating the need for additional steps.
THE FAILURE OF BACKTESTS

- Are not proof of Robustness
- High Potential for Over-Fitting
- False sense of returns
- Reinforces bad design approaches
- Like trying to find a needle in a haystack

WHAT CAN I DO ABOUT THIS ISSUE?
- Sequestered Data (Tests conducted in the Future)
- Out Of Sample Testing
- Walk Forward Testing
- Walk Backwards Testing
- Differential Market Testing
- Stress and Parametric Testing
- Distribution and Matched Pairs Testing (Null)

Reference: ”Pseudo-Mathematics and Financial Charlatanism:
The Effects of Backtest Overfitting on Out-of-Sample Performance, Marcos Lopez de Prado and 3 others.”
WHERE IS THE ALPHA?
Machine Designed Trading Strategies can operate in all categories

TRADING FREQUENCY

LFT
Passive
Long Holding Periods
Index Pegged
Long Term Directional
Hedged
Expiration
Derivative
Portfolio Dynamics
Breakout
Macro Pattern
Global Macro
Event Driven
Relative Value

MFT
Short Holding Periods
Short Term Patterns
Trend/Countertrend
Mean Reversion
Intraday Momentum
Cointegration
Cycle analysis
Volatility Breakout
Volume analysis
Daytrading
Swing Trading
Spreads
Options and combinations
Portfolio of Market-Models

HFT
Order Book Plays
Special Order Types
Hide and Light
Queue Jumping
Spam and Cancel
Top of Book
ARB
ISO
Spreads
Best Execution
Options surface
MACHINE BASED STRATEGY DESIGNS

• Operator does not need to be a programmer
• Allows the machine to explore a wide range
• Not limited to existing theory
• Will quickly find what does and does not work
• May be redirected quickly
• May be reengineered easily
• Human controlled and configured
TSL’s CHALLENGE

IN 2007 WE CHALLENGED ANY MANUALLY DESIGNED SYSTEM TO BEAT TSL’s MACHINE DESIGNED SYSTEMS IN COMPETITIVE THIRD PARTY RATINGS EVALUATED ON SEQUESTERED DATA
HOW IS THE SEQUESTERED DATA COMPETITION PERFORMED?

5/6/2015

TRAINING DATA

OOS DATA

COMPETITION PHASE BEGINS: DESIGN FROZEN

OVERFIT STRATEGY WILL FAIL

2008 WORLD FINANCIAL DISASTER
WE DID IT!

In 2008, and again in 2010, TSL submitted several frozen “Machine Designed” Strategies to Futures Truth. These strategies were initially held for over 18 months, then tested on Sequestered Data, compared and ranked against approximately 700 submissions from over 80 worldwide strategy designers. These systems have not been touched since and reporting on these “Machine Designed” Strategies continues through 2015.
THE RESULTS?
MACHINE CREATED WITH
NO PROGRAMMING REQUIRED

2014 Reports
700+ systems, 80+ vendors

TSL SP on ES
THE RESULTS?
MACHINE CREATED WITH
NO PROGRAMMING REQUIRED
2015 Reports

700+ systems, 80+ vendors
The table below shows the ratings of systems designed in 2007 and held for 18 months, with controlled sequestered testing and re-optimization not allowed. The highest position in all categories was held by SP, NG systems designed in 2007 and held for 18 months. The systems are applied to different symbols over time, with unfavorable bias, variance tradeoff, and retraining needed. The table includes data from 2009 to 2015.

### Table: TSL Futures Truth Ratings Over Time

<table>
<thead>
<tr>
<th>YEAR</th>
<th>SP1z</th>
<th>SP1</th>
<th>US1</th>
<th>US2</th>
<th>NG1</th>
<th>DX1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2010</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2011</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2012</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2013</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>2014</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2015</td>
<td>4-ES</td>
<td>5-ES</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>&lt;10</td>
</tr>
</tbody>
</table>

SP pit closed. Systems now applied to different symbol.

Note: 700+ systems and 80+ developer in competition.
TSL MAIN COMPONENTS

LEARNING MACHINE
### WHAT IS THE BEST LEARNING ALGORITHM?

**Supervised learning**
- AODE
- Artificial neural network
- Backpropagation
- Autoencoders
- Hopfield networks
- Boltzmann machines
- Restricted Boltzmann Machines
- Spiking neural networks

**Unsupervised learning**
- EM
- OLS
- KRR
- PCA
- Probably approximately correct learning (PAC)
- Ripple down rules, a knowledge acquisition methodology
- Symbolic machine learning algorithms
- Subsymbolic machine learning algorithms
- Support vector machines
- Random Forests
- Ensembles of classifiers
  - Bootstrap aggregating (bagging)
  - Boosting (meta-algorithm)
- Ordinal classification
- Regression analysis
- Information fuzzy networks (IFN)
- Conditional Random Field
- Statistical classification
  - ANOVA
  - Linear classifiers
  - Fisher’s linear discriminant
  - Logistic regression
  - Multinomial logistic regression
  - Naïve Bayes classifier
  - Perceptron
  - Support vector machines
- Quadratic classifiers
- k-nearest neighbor
- Boosting
- Decision trees
- C4.5
- Random forests
- Bayesian networks
- Hidden Markov models

**Supervised learning**
- AODE
- Artificial neural network
- Backpropagation
- Autoencoders
- Hopfield networks
- Boltzmann machines
- Restricted Boltzmann Machines
- Spiking neural networks

**Unsupervised learning**
- EM
- OLS
- KRR
- PCA
- Probably approximately correct learning (PAC)
- Ripple down rules, a knowledge acquisition methodology
- Symbolic machine learning algorithms
- Subsymbolic machine learning algorithms
- Support vector machines
- Random Forests
- Ensembles of classifiers
  - Bootstrap aggregating (bagging)
  - Boosting (meta-algorithm)
- Ordinal classification
- Regression analysis
- Information fuzzy networks (IFN)
- Conditional Random Field
- Statistical classification
  - ANOVA
  - Linear classifiers
  - Fisher’s linear discriminant
  - Logistic regression
  - Multinomial logistic regression
  - Naïve Bayes classifier
  - Perceptron
  - Support vector machines
- Quadratic classifiers
- k-nearest neighbor
- Boosting
- Decision trees
- C4.5
- Random forests
- Bayesian networks
- Hidden Markov models

### WHAT ARE THE APPLICABLE STATISTICAL TESTS?

- Kolmogorov-Smirnov
- Shapiro-Wilk
- Anderson-Darling
- Dickey-Fuller
- Wolcoxon
- Wald-Wolfowitz
- Kruskal-Wallis
- ANOVA
- Median test
- Q-statistic
- Sign test
- Friedman
- Cochran Q test
- McNemar test
- Kendall coefficient of concordance
- Spearman rank order R
- Chi-square
- V-square statistic
- Phi
- Gamma
- Sommer’s d
- Paired t-test
- Man-Whitney
- Bootstrap test
- CVAR
- Monte-Carlo Permutation Tests
WHAT IS THE BEST ML SOFTWARE SUITE?

Ayasdi
Angoss KnowledgeSTUDIO
Apache Mahout
Gesture Recognition Toolkit
IBM SPSS Modeler
KNIME
KXEN Modeler
LIONsolver
MATLAB
Mathematica
mlpy
MLPACK library
MCMLL
OpenCV
dlib
Oracle Data Mining
Orange
Discipulus *
Python scikit-learn
R
RapidMiner
Salford Predictive Modeler
SAS Enterprise Miner
Shogun toolbox
STATISTICA Data Miner
Weka

---------Libraries and Tools---------
Accord.NET
ILNumerics
Math.NET Numerics
Wintellect Power Collections
QLNet
Noda Time
R.NET
ALGLIB
LIBSVM

Ref: http://en.wikipedia.org/wiki/Machine_learning
WHAT ARE THE BEST APPLICABLE LANGUAGES OR LIBRARIES?

- AXUM
- F#
- RUBY
- SCALA
- ERLANG
- HASKELL
- PYTHON
- JAVA
- ASSEMBLER
- VB
- C#
- C
- C++
- C++.NET
- C++ AMP
- OPEN MP
- EL
- PL
- WLS
- FORTRAN
- VERILOG
- VHDL
- CUDA
- OPEN CL
- CILK
- CLOURE
- HTML5
- R
REGISTER GENETIC PROGRAMMING

• Based loosely on biological models of evolution and eucaryotic* sexual reproduction
• Simulates the path a biological species goes through as it evolves:
  -Starts off simple
  -Adapts to hostile environment
  -Strong Parents give birth to strong children
  -Random mutations may help
• Works at the FAST CPU Register Level, not high level code
• Fast, Accurate, and Writes Code
• Different from GA and Tree Based GP

*Based on complex cells with membranes


5/6/2015
TSL GP LEARNING

- Supervised Learning. No supervisory Signal.
- Population is initialized
- Trading Strategies are initialized with random signals
- Tournament is run within population applied to the trading simulator
- **Mutation** causes random changes in winners
- **Crossover** exchanges DNA between winners
- **Reproduction** is applied on remainder
- **Demes** enhance genetic diversity
- Parsimony Pressure favor simpler solutions
- If n GWI occur then run restarts
- New trading algorithms emerge and improve based on the error
- Algorithms learn to trade better as they trade in simulation
- After x runs or user termination, all runs stop
- Finally, code is exported, translated and ported to a Trading OMS/EMS
EVOLUTIONARY BASED INDUCTION OF MACHINE CODE
TSL’s Patented GP is 60-200 times faster than other algorithms.

Past: Slow

Present: Fast

Step 1. Convert to machine code

Step 2. Evaluation

Direct Execution. No intermediate steps.

LAIMGP REPRODUCTIVE Crossover
Homologous and Non-Homologous crossover

Trading Algorithm

Trading System can vary its size during evolution

FUNCTION SETS: DNA

More Function Sets allow deeper and wider ranges of solutions to be explored

TSL’s GP is 60-200 times faster than other Algorithms
TSL uses 34 Function Sets including +,-,*,/
MACHINE EVOLVED AND WRITTEN CORE
LOGIC OF YOUR TRADING SYSTEM

Translation Path:
Machine Code  ->  Core Logic C Code  ->  C#, EasyLanguage and others

C#, EL, PL, BLOX or many other platforms

Note only 7 inputs are used here out of the Initial 56 fact Terminal Set available

```c
long double f[8];
long double tmp = 0;
int cflag = 0;


L0:  f[0] = v[25];
L1:  f[0] += v[43];
L2:  f[0] = fabs(f[0]);
L3:  f[0] = v[13];
L4:  f[0] = v[49];
L5:  f[0] = v[41];
L6:  f[0] = v[41];
L7:  f[1] = f[0];
L8:  f[0] = v[22];
L9:  tmp = f[1];
  f[1] = f[0];
  f[0] = tmp;
L10: cflag = (f[0] < f[2]);
L11: f[0] = v[39];

if (!finite(f[0])) f[0] = 0;
return f[0];
```
RML’s Discipulus, used exclusively in TSL, outperformed existing published results.

UXO Discrimination Tests
TSL MAIN COMPONENTS

LEARNING MACHINE

TRADING SIMULATOR
A TRADING SYSTEM MAPS DATA TO EQUITY CURVES

The resultant equity stream net profit $np[n]$ is given by:

$$np[n] = \sum_{t=1}^{n} (tp[t]) + opp$$

The resultant net profit at $t$ is given by:

$$np[t] = np[t - 1] + tp[t] + opp$$
TSL INPUT PREPROCESSING
10 Built In PP’s. Open Code-Fully customizable. 56 Inputs available

Classical and Non-Classical Patterns
- 1, 2 and more bar patterns
- Momentum Patterns
- Counter-trend Patterns
- Trend Patterns
- Gaps and variations
- Adaptive boolean patterns
- Adaptive numeric pattern relationships
- Support and Resistance, adaptations and variations
- Detrended pattern effects and variations

Classical and Non-Classical Indicators
- Normalized variables
- Transforms
- Standard Deviation and variations
- Averages and variations
- Volatility, Volatility Ratios and variations
- Adaptive Channels
- Regressions and variations
- Oscillators and variations
- Detrended prices, oscillators and variations

Other DNA:
- Intermarket data
- Fundamental data
- COT
- Machine readable news
- Social Media
- Exogeneous Data
- Order Book Bid/Ask & Size
- Order Book Movement
- Order Book Stats
SOCIAL MARKET DATA
TSL AND EOTPRO DEVELOPING TECHNOLOGIES

Without Social Media Data
http://www.eotpro.com/

With Social Media Data

5/6/2015
TSL SIMULATION ROUTINES

• 25 Trade Types including multi-systems
• 40 Fitness Functions - External FF API DLL
• 56 Inputs
• 8 Outputs
• 11 Risk, Size, Stops, Targets
• 5 Preprocessors + 5 ID
• EVORUN™
THE STUDY OF ENTRY TYPES

Each order type has many variations. Which one has the best EV for your Market under Study?

Counter Trend
Breakout
Market Order
(TSL has 25 Trade Types)
MARKETS HAVE DIFFERENT DESCRIPTIVE STATISTICS

So Why Design Symmetrical Systems?

TSL will design systems within systems

Power Spectral Density
Indicator Serial Correlation
Random Trend

CME:E-MINI S&P
CBOT:WHEAT
EVORUN™ ON SETUPS

- **(TT) Trade Types/Trading Tactics** are entry or order techniques. 
  Example: Enter on Limit

- **(FF) Fitness Functions** are "Targets" that TSL attempts to design to. 
  Example: NetProfit/Max Drawdown

- **(PP) Preprocessors** are Patterns, Indicators or other facts used as DNA in TSL. 
  Example: Close>Close[1]

- **(TTPD) Trades Per Day.** Determines efficient intra day trading frequency.

- **(BS) Bar Size.** Optimum bar size needs to be stochastically determined.

- There are 25 TT’s, 40 FF’s, 10 PP’s many TTPD’s and BS allowing millions of possible setups to be tested, each allowing millions of systems to be generated for each setup.

- **Clearly there is a need for simplicity and runs reduction**
EVORUN™
WHAT IS THE BEST BAR SIZE?

ES 15 Min?
30 Min?
45 Min?
60 Min?
WHAT IS EVORUN™?

EVORUN is a TSL multi run iterator:
1. Trade Type
2. Fitness Function
3. Preprocessor
4. Bar Size
5. Max Trades per Day
TSL MAIN COMPONENTS

LEARNING MACHINE  TRADING SIMULATOR

FITNESS EVALUATOR
FITNESS CAN BE MULTI GOAL

Machine Design Allows Us to Adjust Critical System Metrics
as Targeted Fitness Function

Net Profit
Drawdown
Percent Accuracy
Profit Factor
Average Trade

Example fitness:

\[ AMR = \left( \frac{1}{n} \right) \sum_{i=1}^{n} POS(MCE(i) - TE(i)) \]
SYSTEMS AS OBJECTS: FITNESS AS EXPRESSIONS

Notice drawdown improved even though DD was not part of fitness
FITNESS GAMES

Increasing Run Time

Dynamic Fitness or DNA injection

Code begins to use different Dominant DNA sets

Better OOS Better TRN Approach #2 Emerges. Comparable to Intermediate strategies

Best OOS Best TRN Alternative Approaches #3 Emerges. Comparable to advanced strategies

Alternative DNA sets emerge into selection

REWARD TO RISK

Increasing Run Time

Code uses Dominant DNA set 1

Good OOS Good TRN Approach #1 Emerges. Comparable to basic strategies

Design Time Generations

How, what, where, when, why it learns

5/6/2015
TSL MAIN COMPONENTS

9 Languages, > 1 million lines of code, 2 companies, 10+ years in development

- EXCHANGE DATA
- EXOGENOUS DATA
  - LEARNING MACHINE
  - TRADING SIMULATOR
  - FITNESS EVALUATOR

- CODE GENERATOR
  Machine Code to C to C#, EL, etc.

- RT PERFORMANCE
  - EXCHANGE
  - DATA
  - ORDERS
  - OMS
  - EMS

- MACHINE DESIGNED TRADING ALGORITHMS
IMPORTANCE OF DATA LENGTH

OOS verses TRN Net Profit
ES Long/Short GWI = 5, TT5, PP1, FF1
1994 to 2014 Back OOS 10%
107,464 matched pairs

R² = 0.2723

OOS verses TRN Net Profit
ES Long/Short GWI = 5, TT5, PP1, FF1
1982 to 2014 Back OOS 10%
107,464 matched pairs

R² = 0.3972
A DISTRIBUTION OF UNIQUE SYSTEMS FROM TSL

Systems are unique and novel. Evolves different systems for each user even with same setup due to Stochastic nature of process.

OOS verses TRN Net Profit
ES Long
TT5, PP1, FF1
1982 to 2014 20% FWD OOS
114,150 matched pairs

\[ y = 1.4047x + 22323 \]
\[ R^2 = 0.7232 \]
A WORD ON SLIPPAGE AND COMMISSION IN TSL-GP

Average Trade is often the limiting issue as trading frequency increases. Using S&C at the start of the run prevents potential good material from evolving into very acceptable end solutions and structures, so apply S&C at the end of the run.

Evolution cannot overcome initial S&C debit. With S&C, acceptable solutions are achieved more efficiently but final solution is acceptable with S&C.
ROBUSTNESS
(Over Fit Avoidance)

- Forward and Back OOS Testing (Walk either)
- Run Path Logs (Path intelligence)
- Unbiased Terminal Set (Directionless inputs)
- Multi-Run, Randomized Criteria (Global optimum)
- Zero Point Origin (No predefined initial point)
- Parsimony Pressure (Occam’s razor)
- Stat Tests-Distribution is exported (Reject Null)
- TTPR (Degrees of Freedom)
- Data duration and choice (More is better)
- Post Design/Post OOS tests (Second Blind)
- Sequestered Data Testing (Extreme testing)
A FEW OMS/EMS THAT CAN HOST TSL TRADING STRATEGIES

- TradeStation (EL)
- MultiCharts (PL)
- Deltix (C#)
- QuantHouse (C#)
- SmartQuant (C#)
- Systemathics (C#)
- LightSpeed (C#)
- OneMarketData (C++)
- NYSE API (C++)
- Mantara (C#)
- WaveRules (C++)
- AB2000 (C++)
- Trading Blox (Blox)
- Ninja (C#)
- WealthLab (WLS)
- Others via native languages or TSL DLL
TRADING STRATEGY DESIGN
REDUCED TO 3 SIMPLE STEPS
No Programming Required

1. Preprocess
2. Evolve
3. Translate
The world is getting smaller and interconnected
- Inter-market Correlations are increasing
- Co-holding risk is largely hidden, expensive to hedge and difficult to diversify against
- The markets keep changing dynamically while systems are not adaptive enough, difficult and expensive to create
- Returns are elusive and require deeper mining
- The worst risks are potentially unknown and unknowable
- Money Managers are typically slow to adapt
TSL MAJOR PROJECTS

- TSL - Current Commercial Platform Product
- Limited External Strategic Consulting
- Internal R&D supporting:
  - Quant Systems Lab (TSL Next Gen)
WHAT IS THE BOTTOM LINE?

There is no way a Human can design as many unique and novel Trading Strategies as TSL
…and then have the code written for you
CONCLUSION

MACHINE LEARNING IN TRADING WILL CONTINUE TO “EVOLVE”

www.tradingsystemlab.com
408-356-1800

Check out the Kindle Book: Best Trading Strategies and our section on Machine Designed Trading Strategies

Join our Silicon Valley Machine Learning for Trading Strategies MeetUp Group
TSL RECENT ADDITIONS

• New Fitness Functions:
  Net Profit/Average Max DD
  Net Profit/Average Trade Duration
  External API User Defined e.g.:
  If Fitness Calls<100000 Fitness = Net Profit/ Avg Max DD
  else Fitness = NetProfit

• Multi Asset Robo Advisor:
  Long Only Portfolio
  Short Only Portfolio
  with Constant Dollars

• GUI Enhancements:
  Quick Save/Quick Load Settings
  Save/Load Any Settings
TSL RECENT ADDITIONS

• EVORUN:
  Performs run iteration on:
  BarSize
  TradeTypes
  PreProcessor
  Fitness Function
  Max Trades Per Day

• SOFT Fitness Targets:
  Average Trade
  MAX Drawdown

• QUANT SYSTEMS LAB
SOLUTION SPACE

What do you do for each of these cases?

• Poor TRN, Poor OOS
• Good TRN, Poor OOS
• Poor TRN, Good OOS
• Good TRN, Good OOS
WHAT IS TSL’S MACHINE DOING NOW?

- TSL is Learning to trade better as it is trading in the Simulator
- Systems are being tested OOS DURING Evolution
- Parameter reduction is automatic
- Strategies are being simplified due to Parsimony Pressure
- Equations are being written and manipulated
- Strategies are improving
- Machine code blocks are being manipulated in FPU’s
- Finally code is up translated from register machine code.