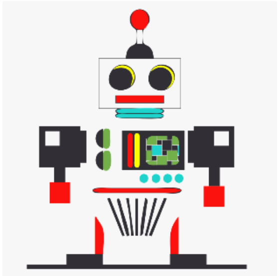
Predicting the Stock Market with Genetic Programming

David Moskowitz, Ph.D.

Infoblazer LLC



Data Scientists in ~~Stamford~~ Westport

January 25, 2016

Disclaimer

- The following is my opinion only
- It is not the opinion of my employer
- It is not related to any work done at my employer

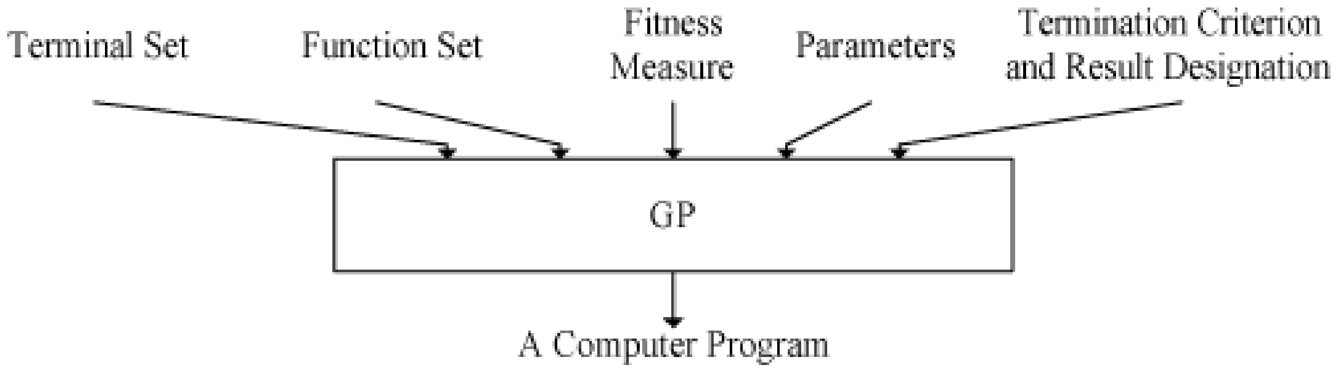
You should not make any decision, financial, investments, trading or otherwise, based on any of the information presented without undertaking independent due diligence and consultation with a professional broker or competent financial advisor. You understand that you are using any and all information presented at your own risk.

Agenda

- What is Genetic Programming?
- Time Series Prediction
- Stock Market Prediction
- Other Issues
  - Modularity
  - Linear GP
  - Genetic Algorithms
- Demonstrations

What is Genetic Programming?

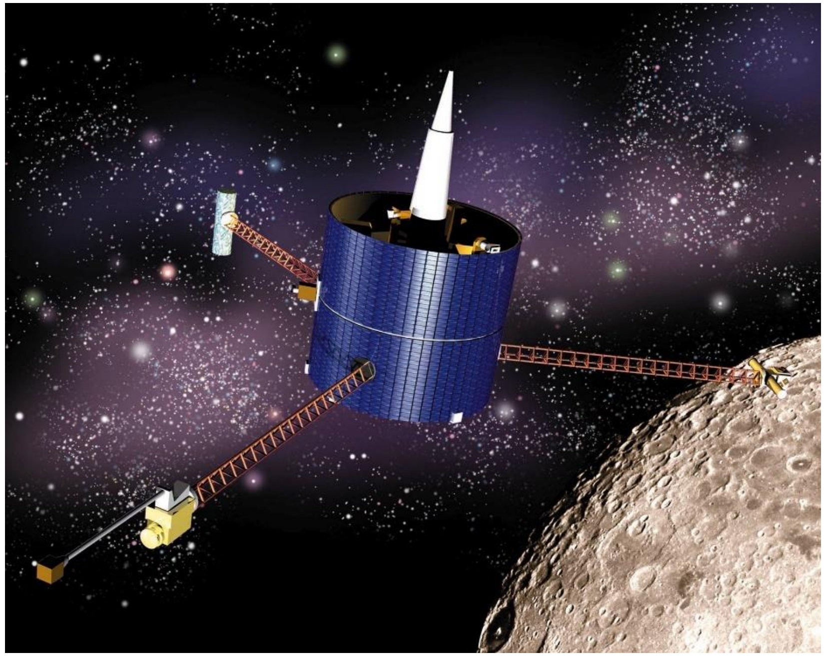
- Get a computer to do something without telling it how to do it
- Breeds a population of computer programs that improve over time
- Evolution
  - Genetic Operators
  - Survival of the Fittest
- Stochastic component
  - Non-Greedy
  - Creativity
  - Novel solutions
- Size and shape of solution unknown
- Limited only by what can be represented as a computer program

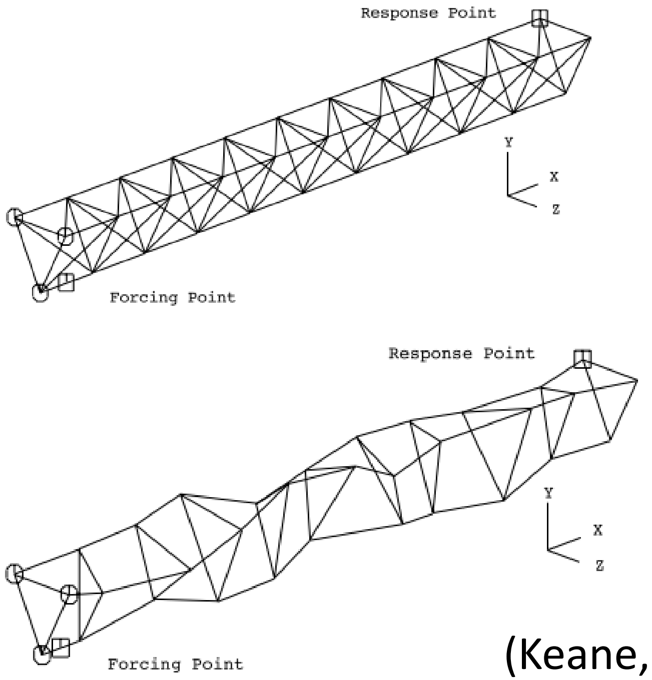


(Koza et al., 2006, p. 11)

Example: Design of a Satellite Boom

- Designed using a genetic algorithm
- 20,000+% improvement in frequency averaged energy levels





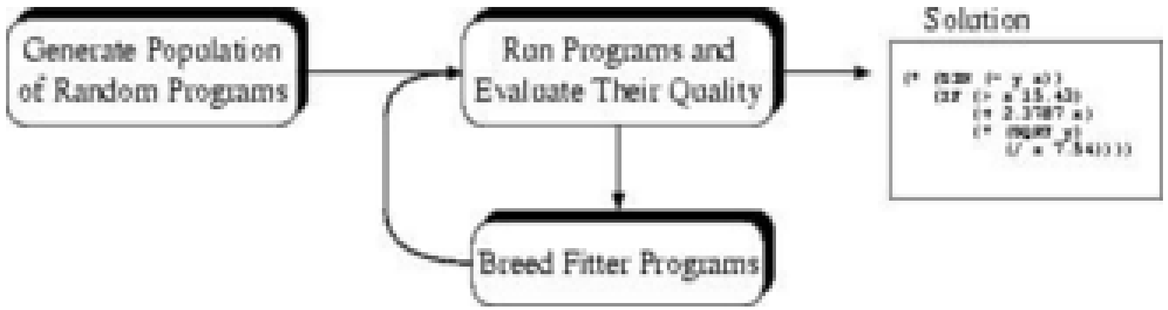
(Keane, 1996)

History

- Visionaries
  - Samuel 1959 – Goal of AI
  - Turing 1948 – Evolutionary search, gene combination, survival of the fittest
- Evolutionary Algorithms , 1962-
  - mutation , populations,
- Genetic Algorithms, 1973-
  - John Holland
  - Crossover
- Genetic Programming, 1989-
  - John Koza
  - Best way to represent a computer program is a computer program

How GP Works

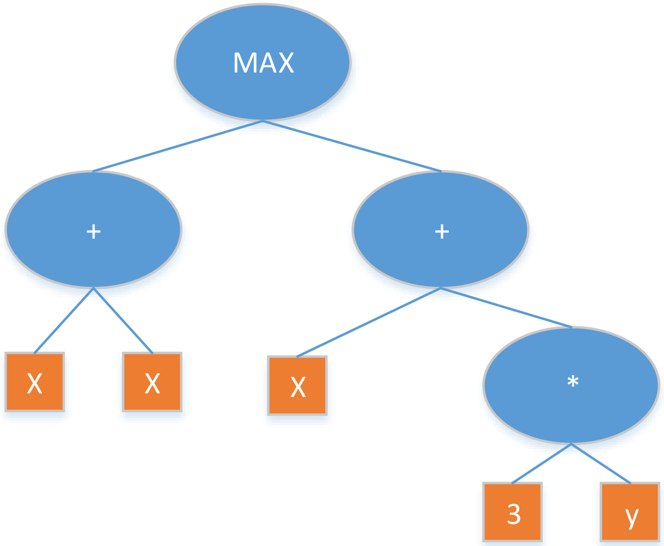
- Preparatory Steps
  - Primitives
  - Fitness Function(s)
- Initialize Population
- Evolve Population
  - Calculate population fitness
  - Select next generation
- Termination Condition



(Poli et al.,2008, p. 2)

GP Representation

- LISP
- (max (+ x x) (+ x (* 3 y)))
- max(x+x,x+3*y)



GP Operations

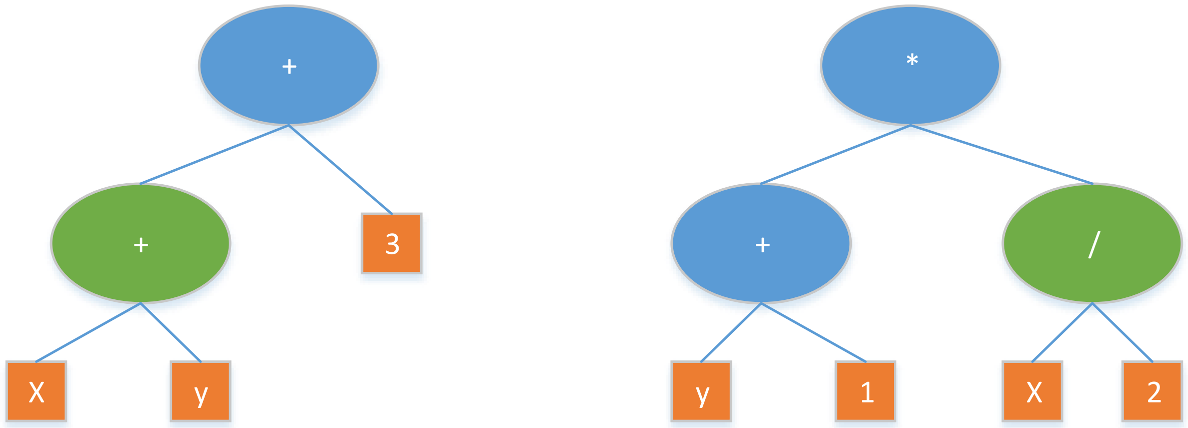
- Probabilistically select an operation
- Crossover
  - Switch two nodes on different individuals
- Mutation
  - Randomly modify an individual (node)
- Reproduction
  - Copy parent, as is, to next generation

GP Selection

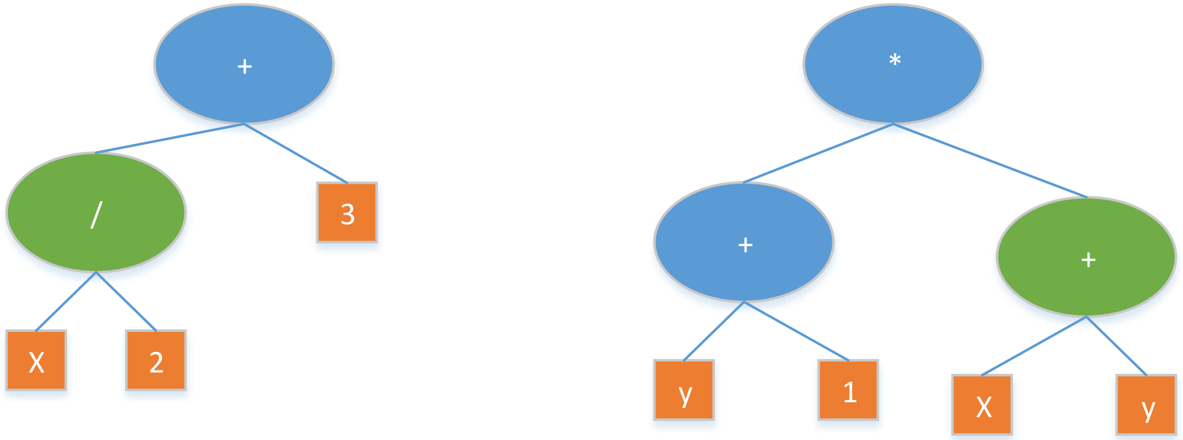
- Need to select one or two individuals for genetic operations
- Selection is probabilistic
- Fitness Proportional Selection
- Tournament Selection

Crossover

Parents

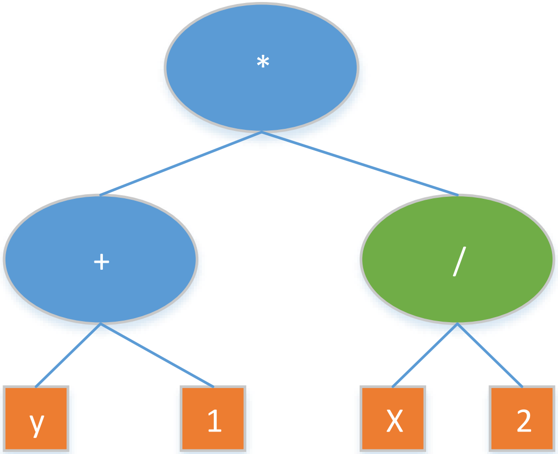


Children

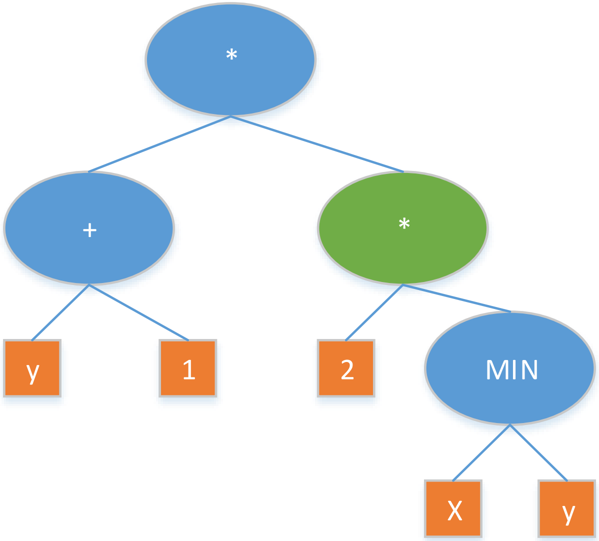


Mutation

Parent

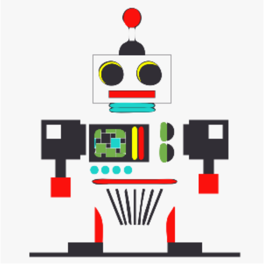


Child

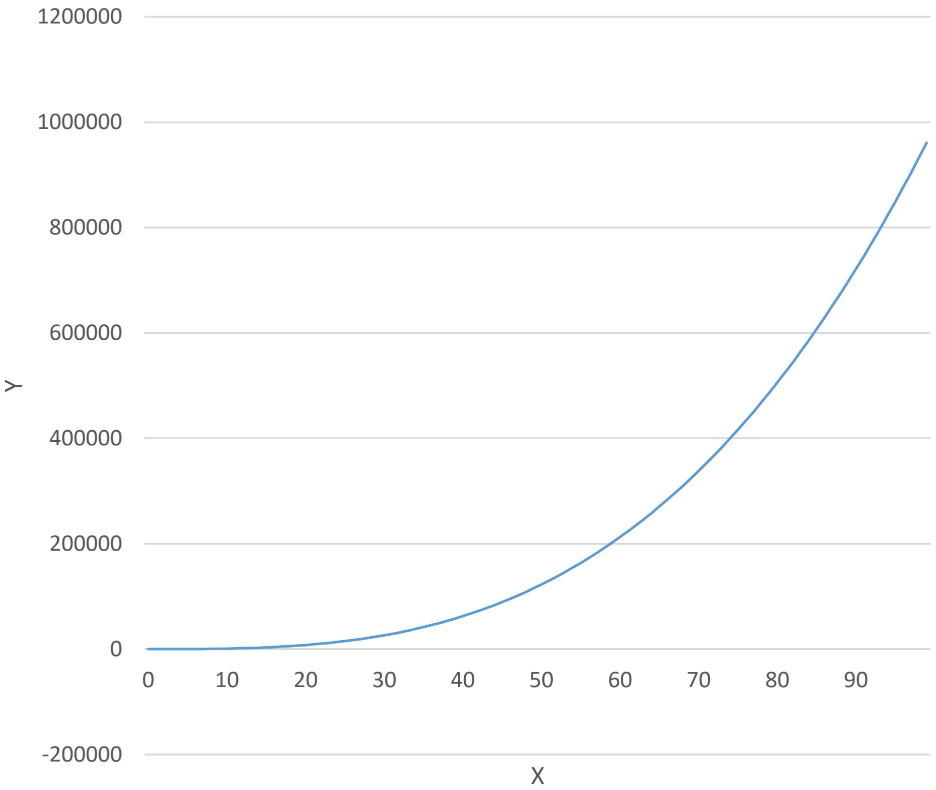


Demo – Symbolic Regression

- Curve Fitting
- $x^3 - x^2 + x - 4$
- Prove that GP works

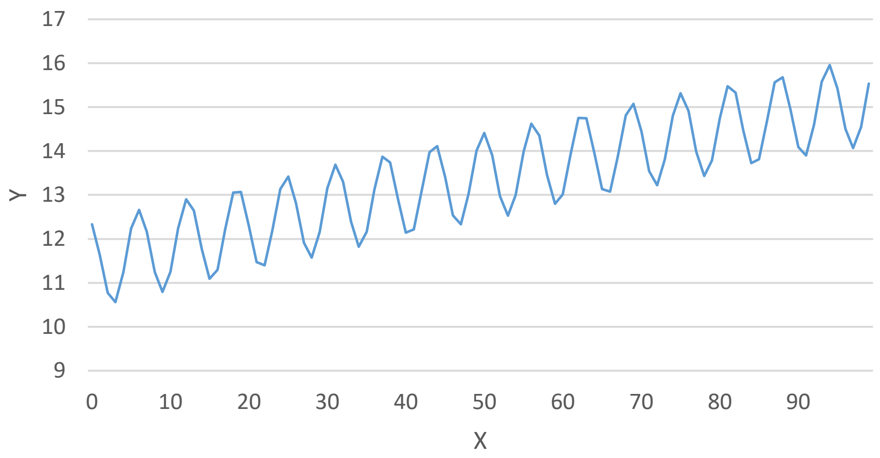


Demo a, b

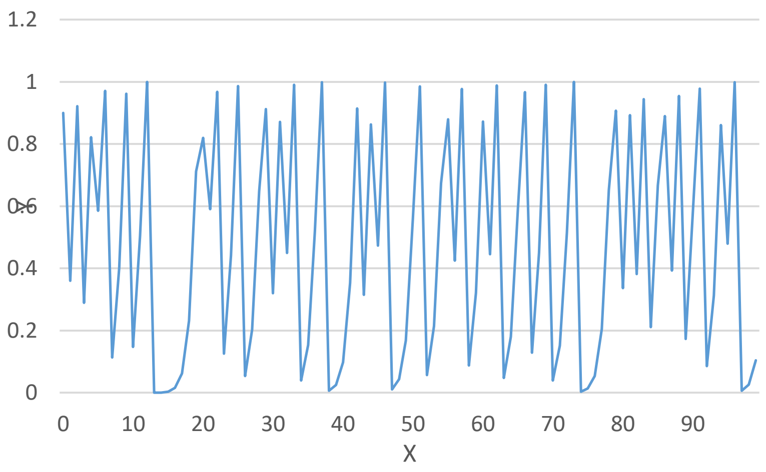


Chaotic Series

- Look random, but are deterministic
- Highly dependent on initial conditions
  - Difficult to predict



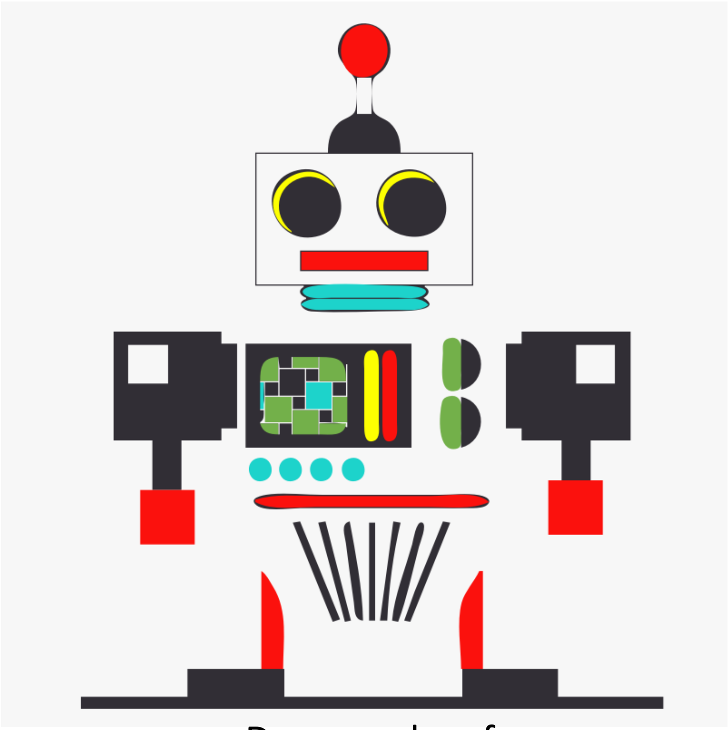
$$Y_t = sin(x - 130) + \sqrt{x + 130}$$



x=0: $Y_{(0)} = 0.9$

X>0: $Y_{(t+1)} = 4Y_t(1 - Y_t)$

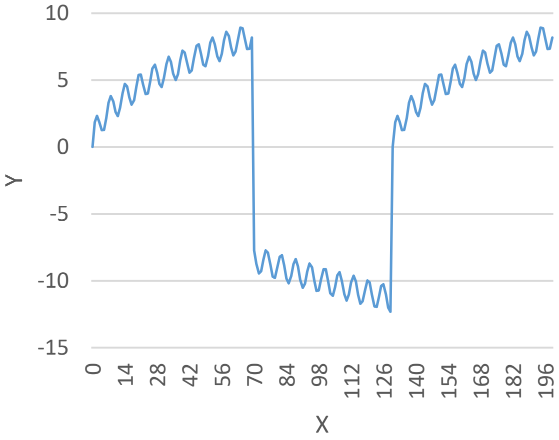
Demo- Chaotic Series Symbolic Regression



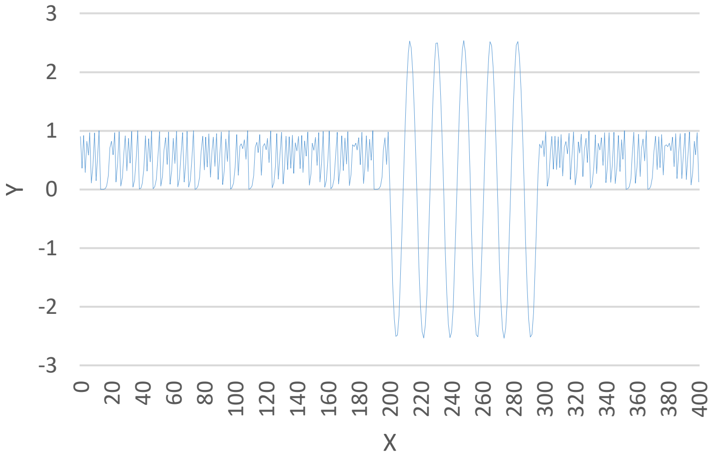
Demo c, d, e, f

Regime Change

- Goal is to uncover underlying data generating process
- This can change over time



0<=x<70: $Y_t = sin(x) + \sqrt{x}$
70<=x<130: $Y_t = cos(x) - \sqrt{x}$
130<=x<200: $Y_t = sin(x - 130) + \sqrt{x - 130}$

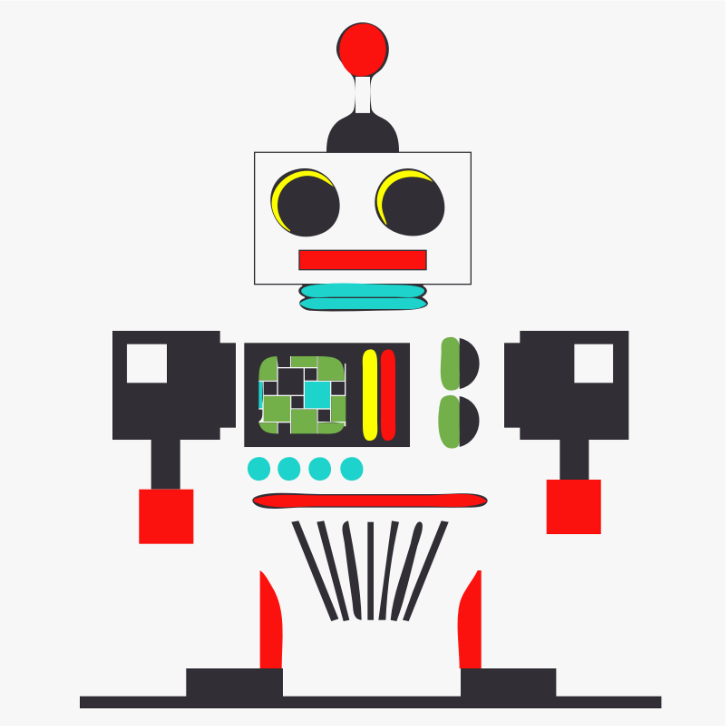


x<200, x>=300: $Y_{(t+1)} = 4Y_t(1 - Y_t)$
200<=x<300: $Y_{(t+1)} = 1.8708Y_t - Y_{t-1}$

S&P 500 index close price during the stock market crash of 2008 (Yahoo, 2013).

$Y_t = f(WTF)$?

Demo- Symbolic Regression Regime Change

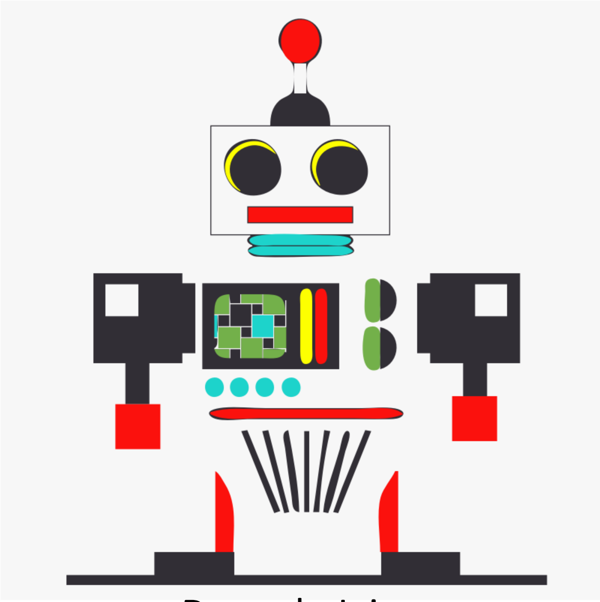


Demo g

Time Series Prediction

- Train on past values
- Predict future values
- Retrain periodically

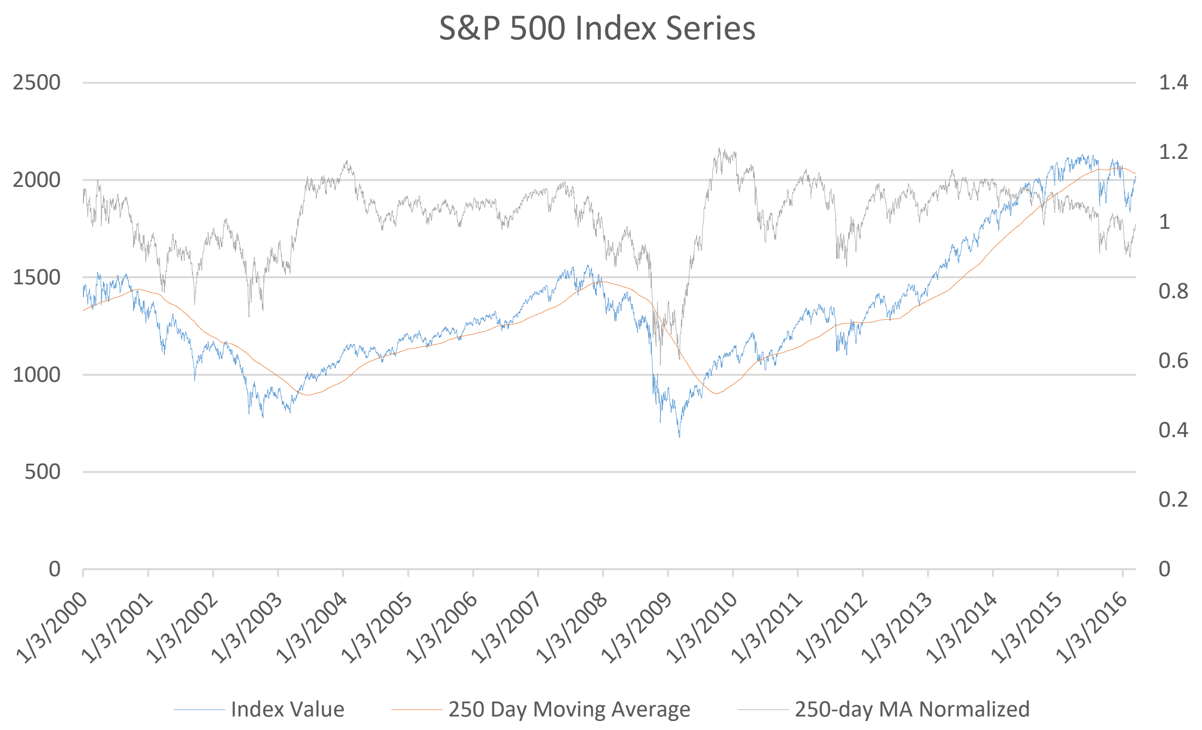
Demo- Chaotic Series Prediction



Demo h, I, j

Market Prediction

- S&P 500 Long-Flat (Invest-don't invest)
- Ignore transaction costs
- Ignore out of market returns
- Predictors
  - S&P 500 Price
  - S&P 500 Volume
  - 250-day MA normalized



GP is a Perfect Match for Market Prediction

- “The interrelationships among the relevant variables is unknown or poorly understood (or where it is suspected that the current understanding may possibly be wrong).”
- “Finding the size and shape of the ultimate solution is a major part of the problem.”
- “Significant amounts of test data are available in computer-readable form.”
- “There are good simulators to test the performance of tentative solutions to a problem, but poor methods to directly obtain good solutions.”
- “Conventional mathematical analysis does not, or cannot, provide analytic solutions”
- “An approximate solution is acceptable (or is the only result that is ever likely to be obtained)”
- “Small improvements in performance are routinely measured (or easily measurable) and highly prized.”

(Poli et al.,2008, pp. 111-113)

Primitive Set

- Hundreds of indicators
  - Ex. (http://www.investopedia.com/active-trading/technical-indicators/)
- Include common technical analysis indicators
  - Momentum- compare to recent average
  - Breakout- compare to recent minimum/maximum
  - Ex. Buy if current price risen by 2% over minimum price last 30 days
- Prefer low level functions
- Better results possible with higher level , packaged indicators?

Primitive set

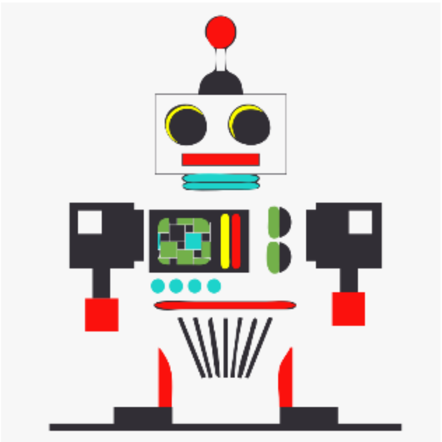
- Functions
  - Add
  - Subtract
  - Multiply
  - Divide
  - Gt
  - Lt
  - And
  - Or
  - Not
  - offsetValue
  - ifElseBoolean
  - movingAverage
  - periodMaximum
  - periodMinimum
  - AbsoluteDifference

- Terminals
  - randomInteger(low high)
  - randomDouble(low high)
  - True
  - False
  - offsetValue(0)

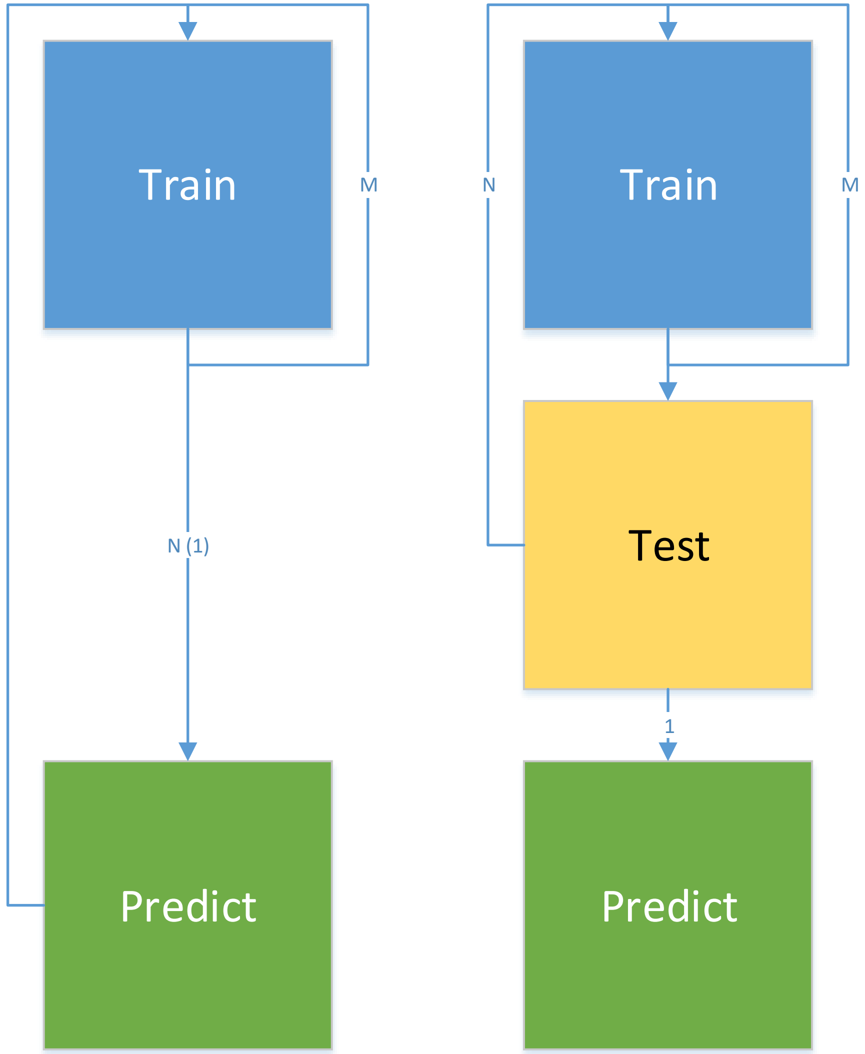
- Hundreds of other technical indicators (http://www.investopedia.com/active-trading/technical-indicators/)

Training Approaches

- Train-Predict-Retrain
- Train-Test-Predict
- Multiple Runs

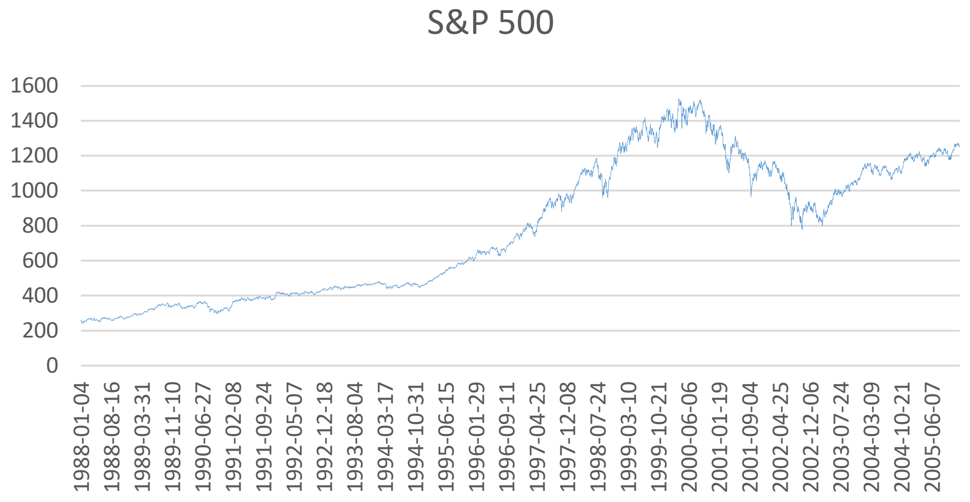


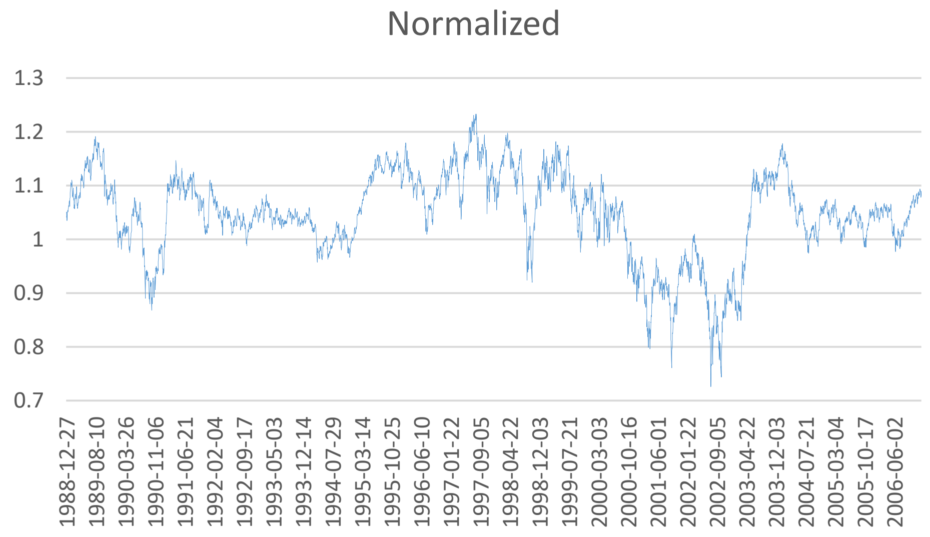
Demo k, l, m



Experiment- Market Prediction

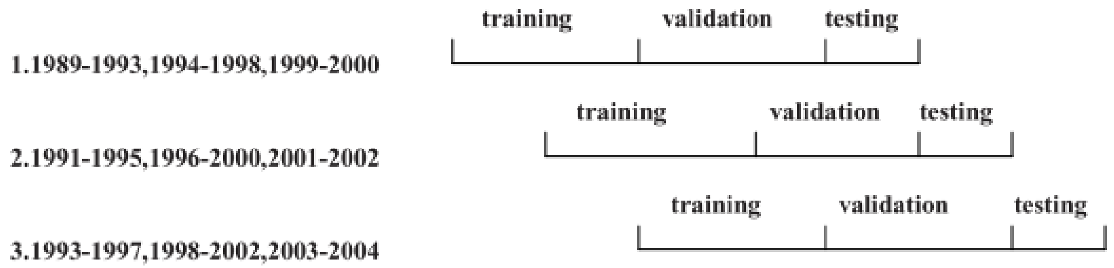
- Investment decisions in S&P 500 Index
- Modeled after (Chen et al., 2008), 1988-2004
- Long-Flat decisions
- Normalized by 250-day moving average
- Fitness = investment gain





Experiment- Market Prediction

- Training-validation-prediction approach (T-V-P)



(Chen et al., 2008, p. 110)

- Training-prediction approach (T-P)
  - Training - 1989-1998
  - Prediction- 1999-2004

Results T-V-P w/Trans Cost

Method	Mean	Std. Dev.	Min	Max	95% CI	# beating benchmark
1999-2000						
Buy & Hold	0.0751					
GP	0.0434	0.0664	-0.1917	0.1197	[0.0250 ... 0.0618]	5/50
ADF	0.0309	0.0798	-0.3054	0.0845	[0.0088 ... 0.0530]	3/50
ADT	0.0510	0.0519	-0.1974	0.1042	[0.0366 ... 0.0654]	5/50
2001-2002						
Buy & Hold	-0.3144					
GP	-0.3693	0.1306	-0.8087	-0.2885	[-0.4055 ... -0.3331]	1/50
ADF	-0.3347	0.0887	-0.7290	-0.1777	[-0.3593 ... -0.3102]	2/50
ADT	-0.3697	0.1390	-0.7450	-0.0134	[-0.4082 ... -0.3312]	1/50
2003-2004						
Buy & Hold	0.3332					
GP	0.2945	0.0497	0.1432	0.3291	[0.2807 ... 0.3083]	0/50
ADF	0.3139	0.0390	0.1170	0.3539	[0.3031 ... 0.3247]	1/50
ADT	0.3247	0.0150	0.2349	0.3522	[0.3205 ... 0.3289]	2/50

(Moskowitz, 2016, p. 119)

Results T-V-P wo/Trans Cost

Method	Mean	Std. Dev.	Min	Max	95% CI	# beating benchmark
1999-2000						
Buy & Hold	0.0751					
GP	0.1494	0.1088	-0.0438	0.4525	[0.1192 ... 0.1795]	35/50
ADF	0.1418	0.1238	-0.0399	0.5112	[0.1075 ... 0.1761]	35/50
ADT	0.1567	0.1099	-0.0068	0.4796	[0.1262 ... 0.1871]	37/50
2001-2002						
Buy & Hold	-0.3144					
GP	-0.3121	0.0573	-0.4081	-0.0348	[-0.3280 ... -0.2962]	17/50
ADF	-0.3023	0.0848	-0.5153	0.0196	[-0.3258 ... -0.2788]	18/50
ADT	-0.2843	0.0635	-0.3924	-0.1245	[-0.3020 ... -0.2667]	32/50
2003-2004						
Buy & Hold	0.3332					
GP	0.3045	0.0929	0.0463	0.5045	[0.2788 ... 0.3303]	15/50
ADF	0.3395	0.1171	-0.0016	0.5597	[0.3070 ... 0.3719]	22/50
ADT	0.3329	0.1202	0.0775	0.6443	[0.2996 ... 0.3663]	29/50

(Moskowitz, 2016, p. 122)

Results T-P w/Trans Cost

Method		Mean	Std. Dev.	Min	Max	95% CI	# beating benchmark
		1999-2000					
Buy & Hold		0.0634					
ADT		0.0018	0.1372	-0.4290	0.2388	[-0.0362 ... 0.1010]	17/50
DyFor GP		-0.0157	0.1101	-0.2690	0.1426	[-0.0463 ... 0.0639]	15/50
		2001-2002					
Buy & Hold		-0.3339					
ADT		-0.1364	0.1014	-0.2964	0.1601	[-0.1645 ... -0.0631]	50/50
DyFor GP		-0.1018	0.0819	-0.2810	0.0817	[-0.1245 ... -0.0426]	50/50
		2003-2004					
Buy & Hold		0.2970					
ADT		0.1035	0.0653	-0.0603	0.2529	[0.0854 ... 0.1507]	0/50
DyFor GP		0.0489	0.0723	-0.1780	0.2156	[0.0289 ... 0.1012]	0/50
		1999-2004					
Buy & Hold		-0.0189					
ADT		-0.0349	0.1933	-0.5395	0.4592	[-0.0884 ... 0.0187]	24/50
DyFor GP		-0.0698	0.1413	-0.3597	0.2136	[-0.1089 ... -0.0306]	15/50

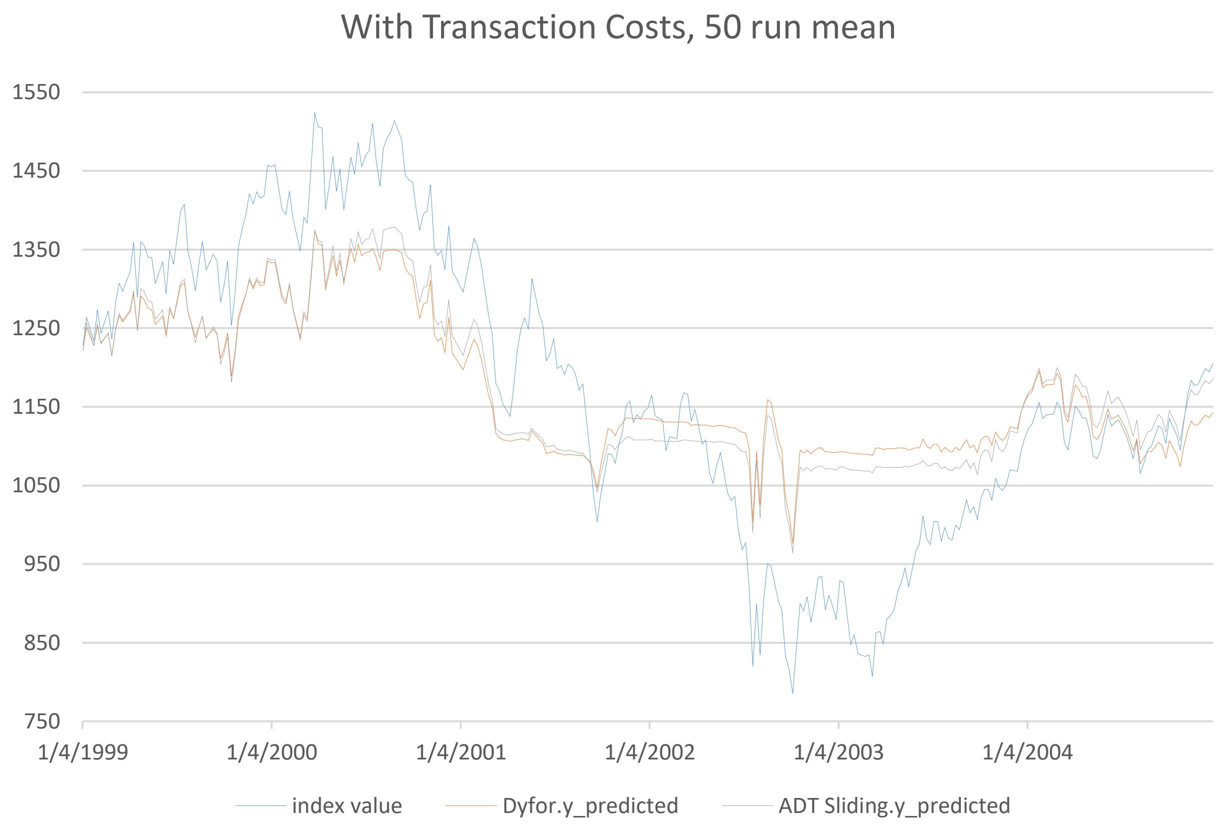
(Moskowitz, 2016, p. 120)

Results T-P wo/Trans Cost

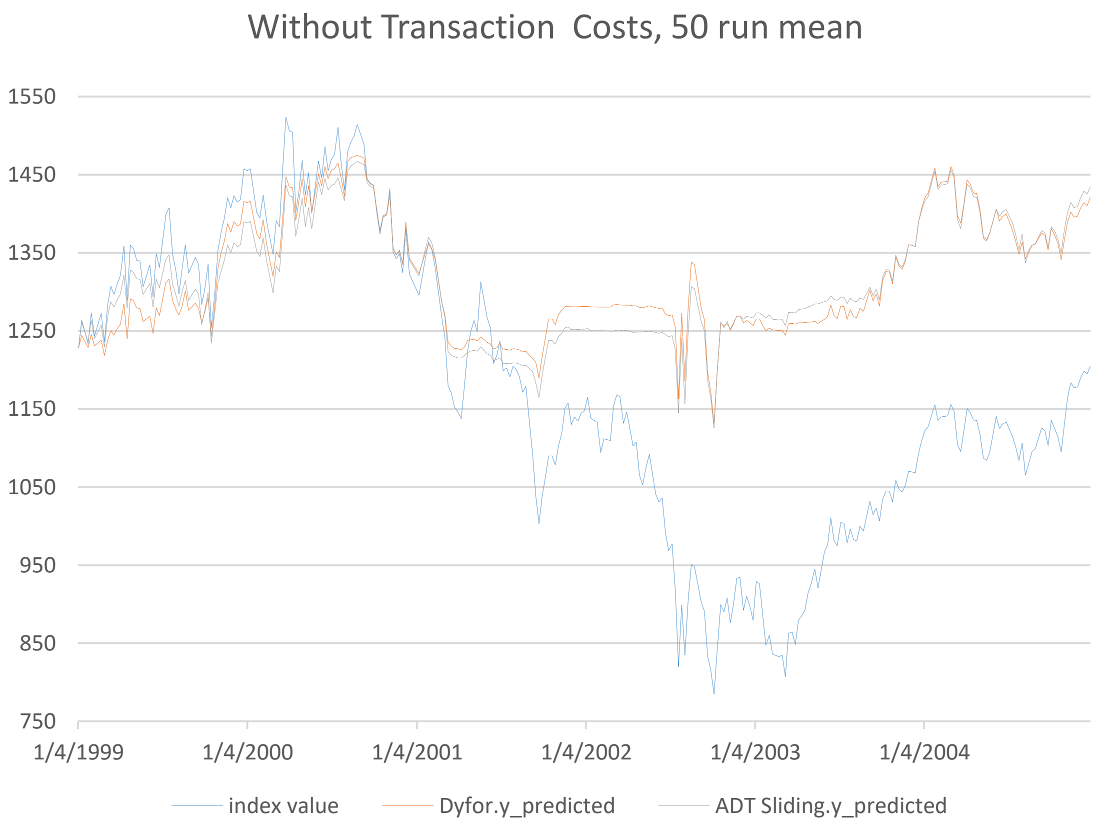
Method	Mean	Std. Dev.	Min	Max	95% CI	# beating benchmark
1999-2000						
Buy & Hold	0.0634					
ADT	0.0788	0.1071	-0.1106	0.3576	[0.0491 ... 0.1562]	27/50
DyFor GP	0.0807	0.1323	-0.1904	0.3408	[0.0440 ... 0.1763]	26/50
2001-2002						
Buy & Hold	-0.3339					
ADT	-0.0524	0.1026	-0.2674	0.1521	[-0.0808 ... 0.0218]	50/50
DyFor GP	-0.0594	0.0862	-0.2314	0.1020	[-0.0833 ... 0.0029]	50/50
2003-2004						
Buy & Hold	0.2970					
ADT	0.1246	0.0782	-0.0132	0.3739	[0.1029 ... 0.1811]	2/50
DyFor GP	0.1233	0.0702	-0.0297	0.2783	[0.1038 ... 0.1740]	0/50
1999-2004						
Buy & Hold	-0.0189					
ADT	0.1683	0.2005	-0.1946	0.6959	[0.1128 ... 0.2239]	39/50
DyFor GP	0.1568	0.1887	-0.2618	0.5762	[0.1045 ... 0.2091]	41/50

(Moskowitz, 2016, p. 124)

ADT vs DyFor GP vs Buy and Hold



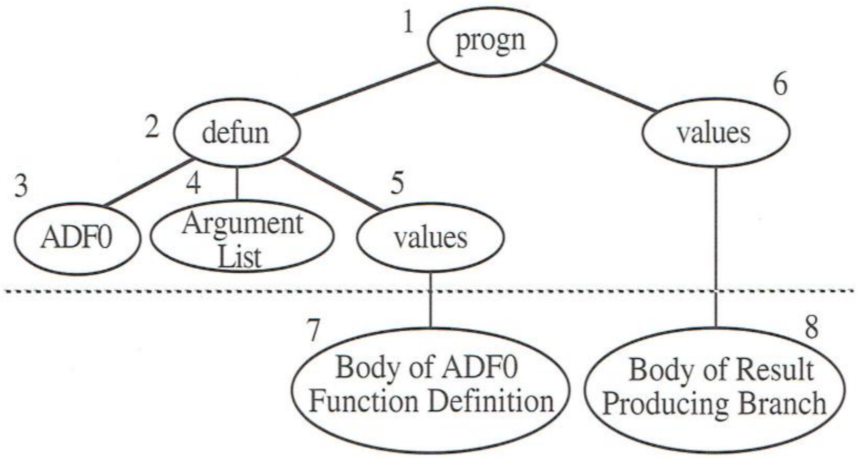
ADT: -0.349%
DyFor GP: -0.698%
B&H: -0.0189%
(Moskowitz, 2016, p. 208)



ADT: +0.1683%
DyFor GP: +0.1568%
B&H: -0.0189%
(Moskowitz, 2016, p. 213)

Advanced GP

- Modularity
  - Automatically Defined Functions
- Strongly-typed GP
  - Closure
- Advanced techniques
  - Looping
  - Memory store
  - Lambdas
  - Recursion
  - Time series regimes (Moskowitz, 2016)
  - Design patterns (Moskowitz, 2016)



(Koza, 1994, p. 74)

Genetic Algorithms

- Non-differentiable / nonlinear optimization problem
- Search for parameters, rules
- Size and shape prescribed
- Bit, Numeric, or other representations
- Ex. Minimize $x^2 - 50y + z^3$, x={0-31},y={0-31},z={0-15}

10011 11000 1011 $19^2 - 50 * 24 + 11^3$=492
00110 11010 1100 $6^2 - 50 * 26 + 12^3$=464

10010 11010 1100 $18^2 - 50 * 26 + 12^3$=752
00111 11000 1011 $7^2 - 50 * 24 + 11^3$=180

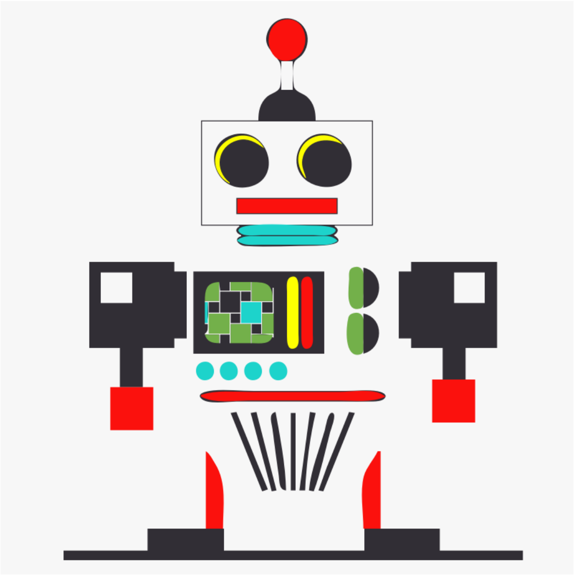
Linear GP

- Sequence of imperative instructions
- Register-based operations
- Machine code, GPU Instructions

(Poli et al.,2008, pp. 61-65)

Demo- Symbolic Regression Regime Change

- Regime determining branch
- Regime specific functions
- Implements template method design pattern



Demo o

Next Steps

- MATLAB (GA only)
- JGAP (Java)
- DEAP (Python)
- Roll your own
- Evolutionary Signals

Not So Shameless Plug

- Evolutionary Signals
  - Develop Buy/Sell signals using genetic programming
  - Minimal financial knowledge and no programming experience required
  - Goals
    - Much larger number of predictor series
    - Crowdsource models
  - Earn revenue from high performing models
  - Open beta testing
  - Visit www.gpsignals.com for more information

Questions?

- Thank you!

- Contact info:

  LinkedIn: infoblazer

  @infojester

References

- Chen, S. H., Kuo, T. W., & Hoi, K. M. (2008). Genetic Programming and Financial Trading: How Much About "What We Know." In Handbook of financial engineering (pp. 99–154). Springer US. doi:10.1007/978-0-387-76682-9
- Keane, A. J. (1996). THE DESIGN OF A SATELLITE BOOM WITH ENHANCED VIBRATION PERFORMANCE USING GENETIC ALGORITHM TECHNIQUES. The Journal of the Acoustical Society of America, 99(4), 2599–2603.
- Koza, J. R. (1994). Genetic programming II: automatic discovery of reusable programs. MIT press.
- Koza, J. ., Keane, M. A., Streeter, M. J., Mydlowec, W., Yu, J., & Lanza, G. (2006). Genetic programming IV: Routine human-competitive machine intelligence (Vol. 5). Springer.
- Moskowitz, D. (2016). Automatically Defined Templates for Improved Prediction of Non-stationary, Nonlinear Time Series in Genetic Programming. Doctoral dissertation. Nova Southeastern University. Retrieved from http://nsuworks.nova.edu/gscis_etd/953.
- Poli, R., Langdon, W. B., McPhee, N. F., & Koza, J. R. (2008). *A field guide to genetic programming*. Lulu. com.