Northfield's 31st Annual Research Conference
Washington DC

Machine Learning for Algorithmic Trading and Trade Schedule Optimization

Robert Kissell, PhD

October 27, 2019

RKissell@Molloy.edu
Molloy College, School of Business

Robert.Kissell@KissellResearch.com
Kissell Research Group

"Molloy Means Business"

Machine Learning for Algorithmic Trading and Trade Schedule Optimization

Robert Kissell, PhD
Molloy College
Kissell Research

October 27, 2019

Abstract:
We present a machine learning technique that can be used in conjunction with multi-period trade schedule optimization in program trading. The technique is based on an artificial neural network (ANN) that determines a better starting solution for the non-linear optimization routine. Unlike many current industry approaches that use heuristics and numerical approximation, our machine learning approach solves for the exact problem and provides a dramatic improvement in calculation times. As such, we believe this this technique represents a material advance in the ability of asset managers to more efficiently manage trade execution for large transactions.

Research Topic

Previous Research and Preliminary Findings were published in:

"Machine Learning for Algorithmic Trading and Trade Schedule Optimization,"
Robert Kissell and Jungsun "Sunny" Bae
Journal of Trading, Fall 2018 Volume 13, No. 4, Pg. 138-147.

"Algorithmic Trading Methods: Applications using Advanced Statistics, Optimization, and Machine Learning Techniques," Second Edition
Robert L. Kissell, Ph.D.
Elsevier/Academic Press, 2019.

Presentation Outline

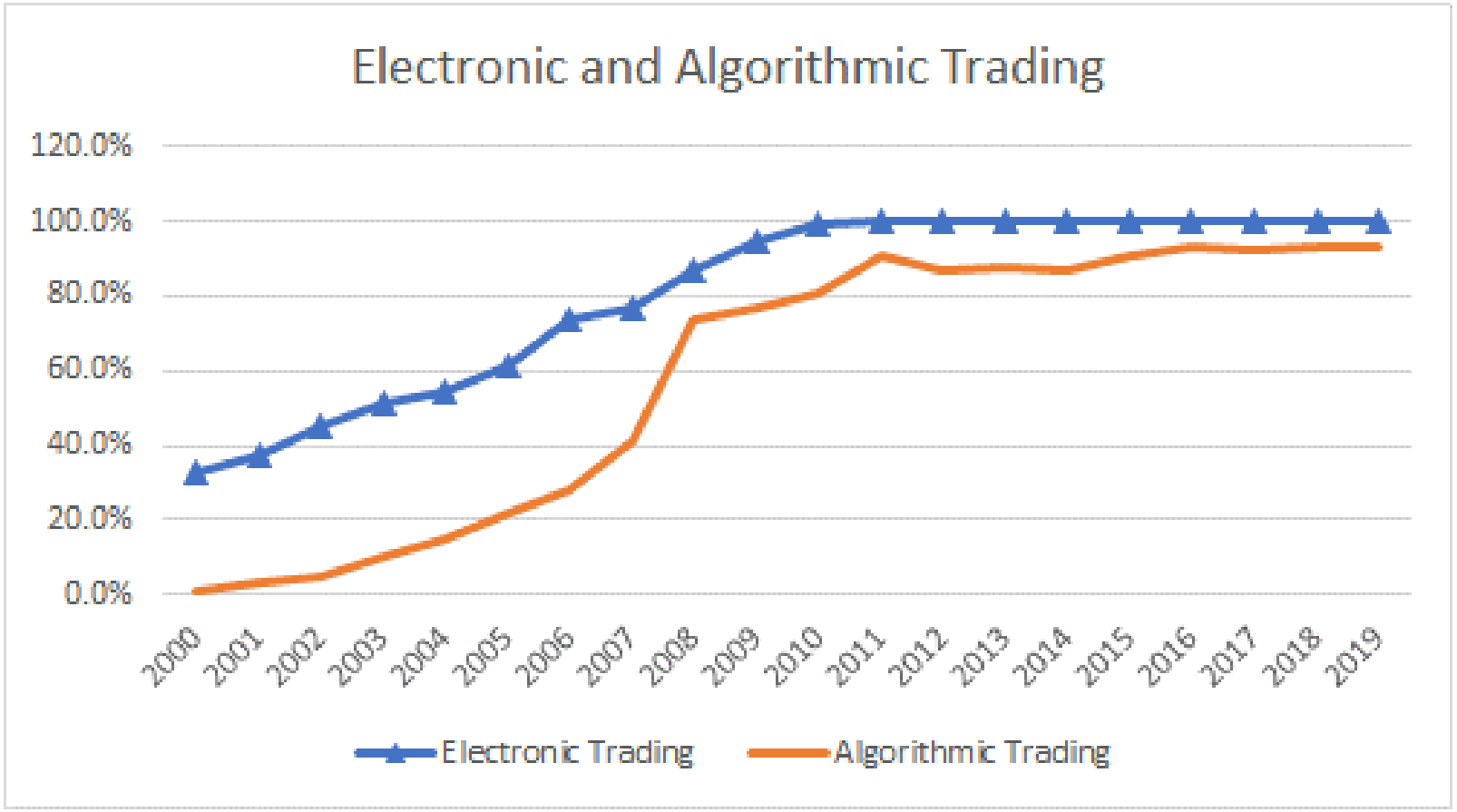
- Algorithmic Trading
- Multi-Period Trade Schedule Optimization
  - Market Impact Model, Trader's Dilemma
- Machine Learning & Data Analytics
  - Descriptive, Predictive, Prescriptive
- Convergence Algorithms
  - Newtons Method, Gradient Descent, Neural Networks
- Optimization Experiment
  - Data Inputs, Stepwise Regression, Best Variables
- Results/Conclusions

What do we want to do?

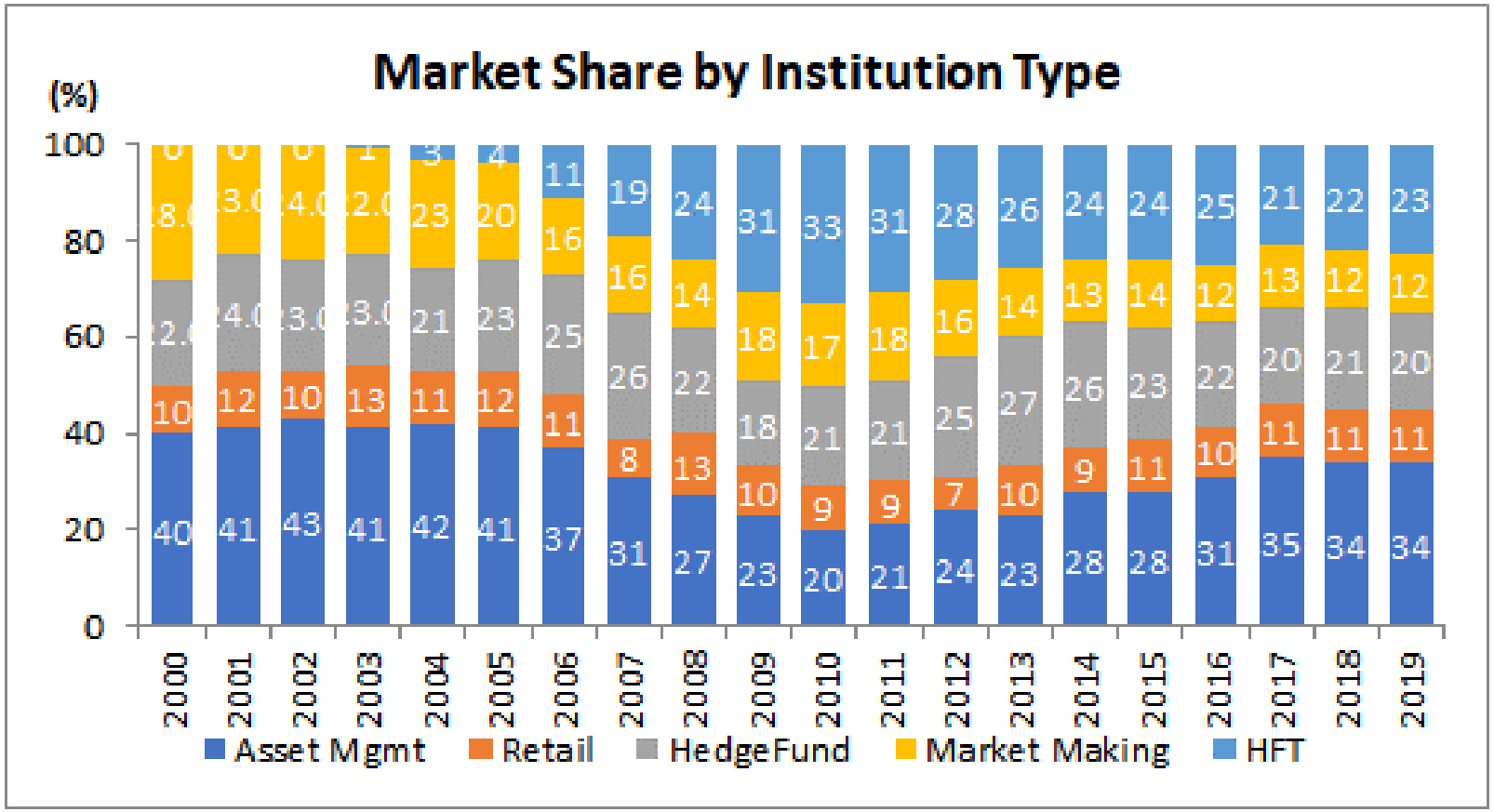
- Solve Real-Time Algorithmic Optimization Problems Faster.
- Take advantage of Profiting Opportunities when they arise, and before other market participants uncover the opportunity.
- Make Money!

Algorithmic Trading Trends

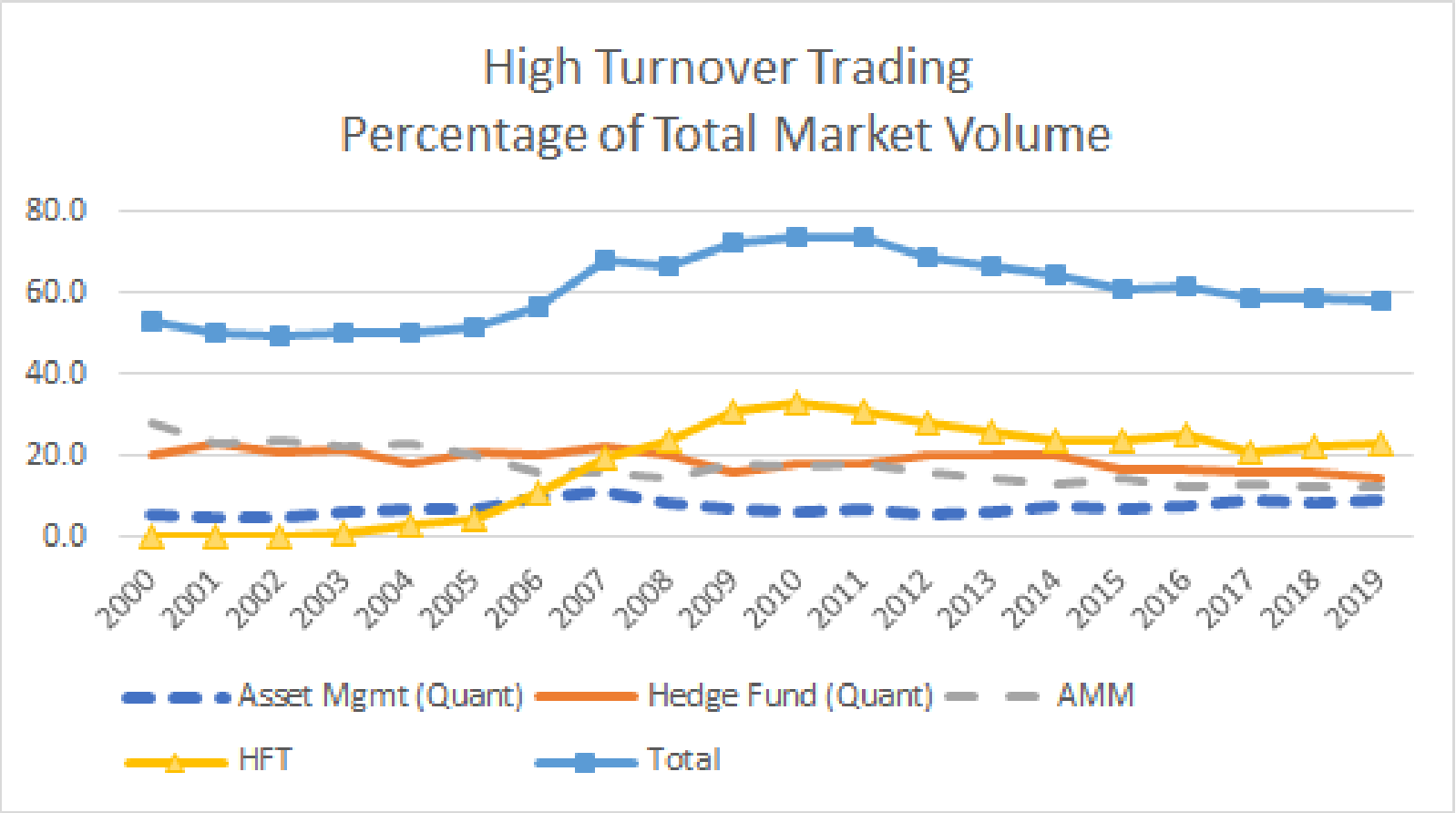
Electronic and Algorithmic Trading Volume



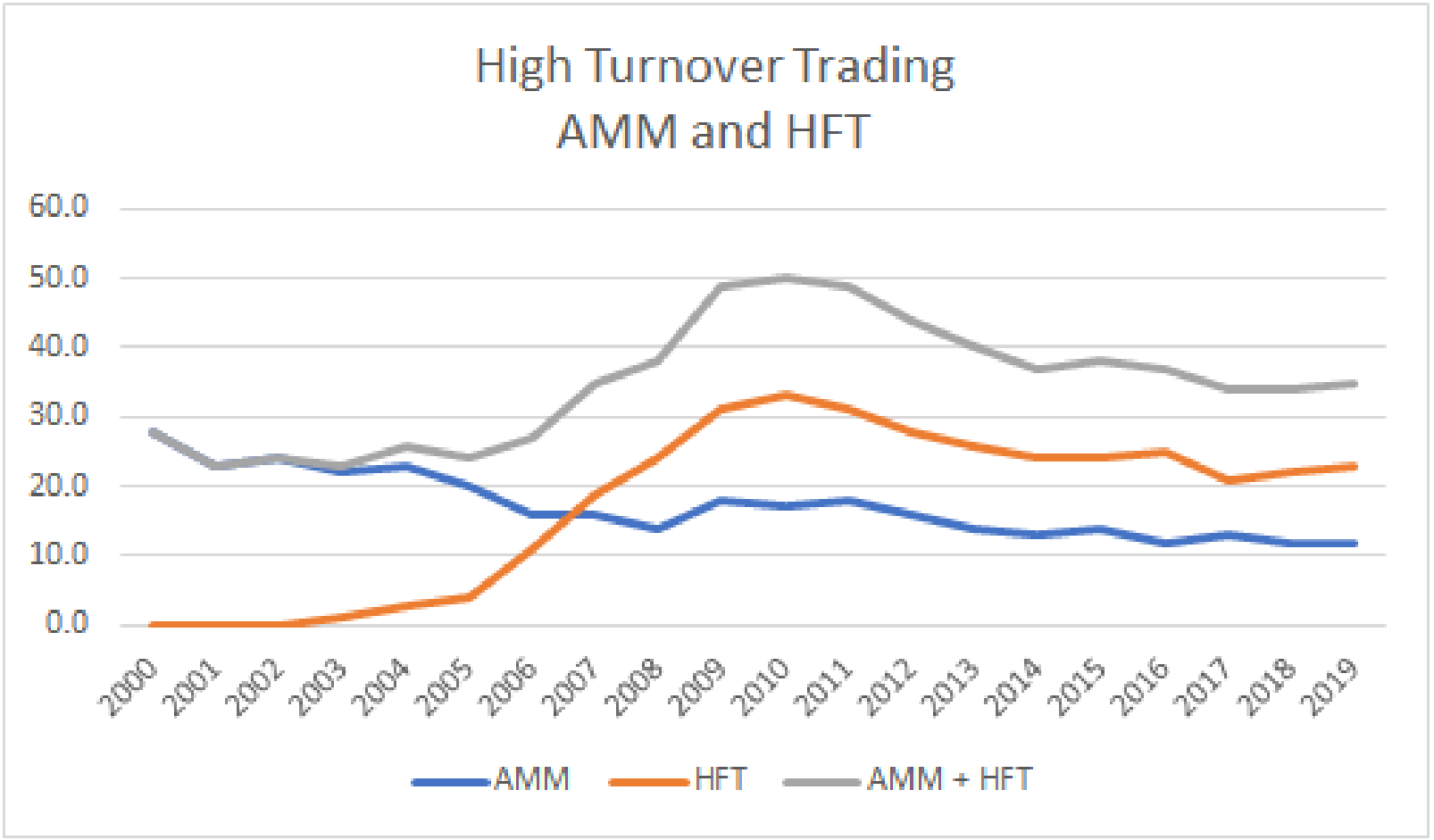
Market Share by Institution Type



High Turnover Trading



Market Share by Institution Type



Market Impact Model

I-Star Market Impact Model – POV Strategy

$$I^* = a_1 \cdot \left(\frac{X}{ADV}\right)^{a4} \cdot \sigma^{a3} \cdot P^{a5}$$

$$MI^* = b_1 \cdot I^* \cdot POV^{a4} + (1 - b_1) \cdot I^*$$

$$TR = \sigma \cdot \sqrt{\frac{1}{3} \cdot \frac{1}{250} \cdot \frac{X}{ADV} \cdot \frac{(1 - POV)}{POV}} \cdot 10^{-4}$$

Variables:

X = Shares to Trade

ADV = Average Daily Volume

σ = annualized volatility (expressed as a decimal)

P = Price

POV = percentage of volume (expressed as a decimal)

a_1, a_2, a_3, a_4, b_1 = model parameters, $a_k > 0$; $0 \leq b_1 \leq 1$

"Molloy Means Business"

Source: Science of Algorithmic Trading (Kissell, 2013)

Trader's Dilemma

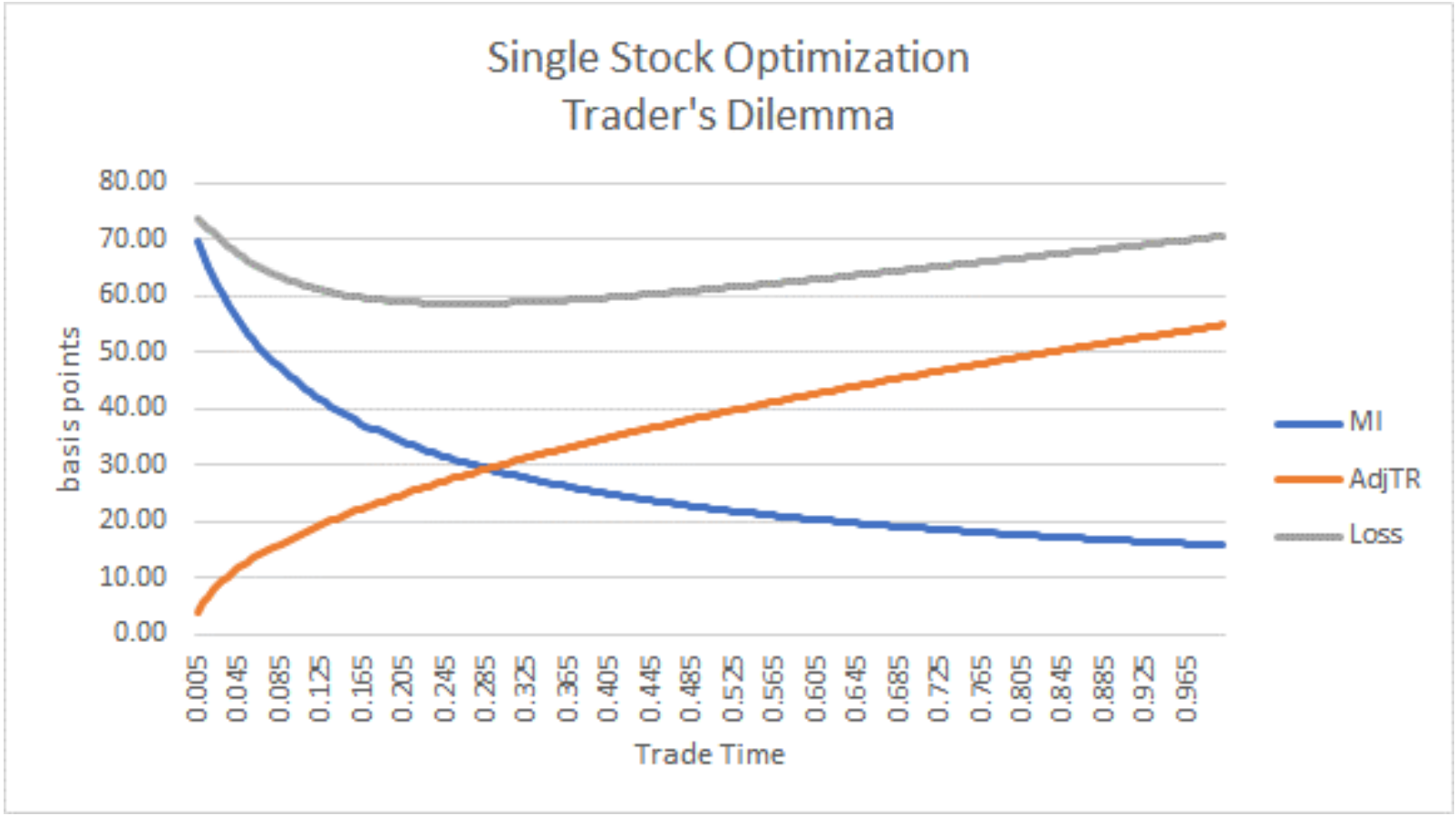
$$Min \quad MI + \lambda \cdot TR$$

Trader's Dilemma:

- Trading fast incurs high market impact cost. Trading slow exposes the trade to timing risk due to price volatility and volume uncertainty.
- Traders want to balance the trade-off between market impact cost and timing risk.

Source: Science of Algorithmic Trading (Kissell, 2013)

Single Stock Optimization: Trader's Dilemma



"Molloy Means Business"

Source: Science of Algorithmic Trading (Kissell, 2013)

I-Star Market Impact Model – Trade Schedule

$$\underset{POV}{Min} \; L(\$) = \sum \left((b_1 \cdot I_i^* \cdot POV^{a4} + (1 - b_1) \cdot I_i^*) \cdot X_i \cdot P_i \right)$$

$$+\lambda \cdot \sqrt{trace(R'CR)}$$

where,

$$R = f(POV)$$

s.t.

i) $\min POV \leq POV \leq maxPOV$

ii) $Cash\ Balancing$

Variables:

I^* = Instantaneous Impact

X = Shares to Trade

R= Matrix of Residuals

C= Covariance Matrix, One-Sided, ($/Share)^2

POV=Percentage of Volume (Trade Rate)

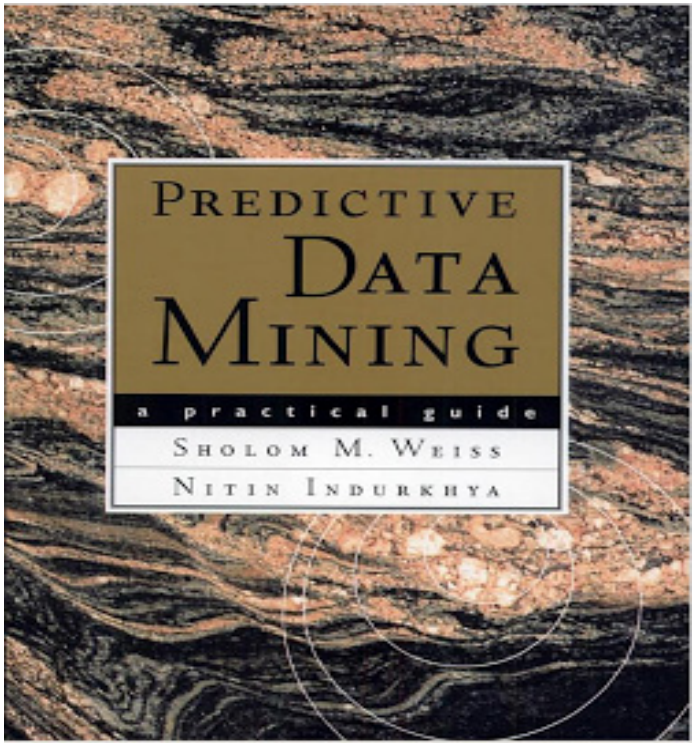
Source: Science of Algorithmic Trading (Kissell, 2013)

Machine Learning

Machine Learning

- Machine Learning gives computers the ability to learn without being explicitly programmed
  - (Arthur Samuel, 1959).
- Machine Learning Algorithms:
  - Neural Networks, Genetic Algorithms, Cluster Analysis, Decision Trees, Classification, Pattern Recognition, K-Means, Newton's Method, Gradient Descent

Predictive Data Mining (1998)



Machine Learning

- Cluster Analysis
- Classification
- Regression

Machine Learning

- Unsupervised
- Supervised

Machine Learning

- Descriptive
- Predictive
- Prescriptive

Machine Learning

- Descriptive
- Predictive
- Prescriptive

- “Semi-Prescriptive”

Fun Fact #1

- When did Sports Analytics Begin?
  - Bill James, Sabermetrics
  - Billy Bean, Oakland A's
  - Michael Lewis, Money Ball

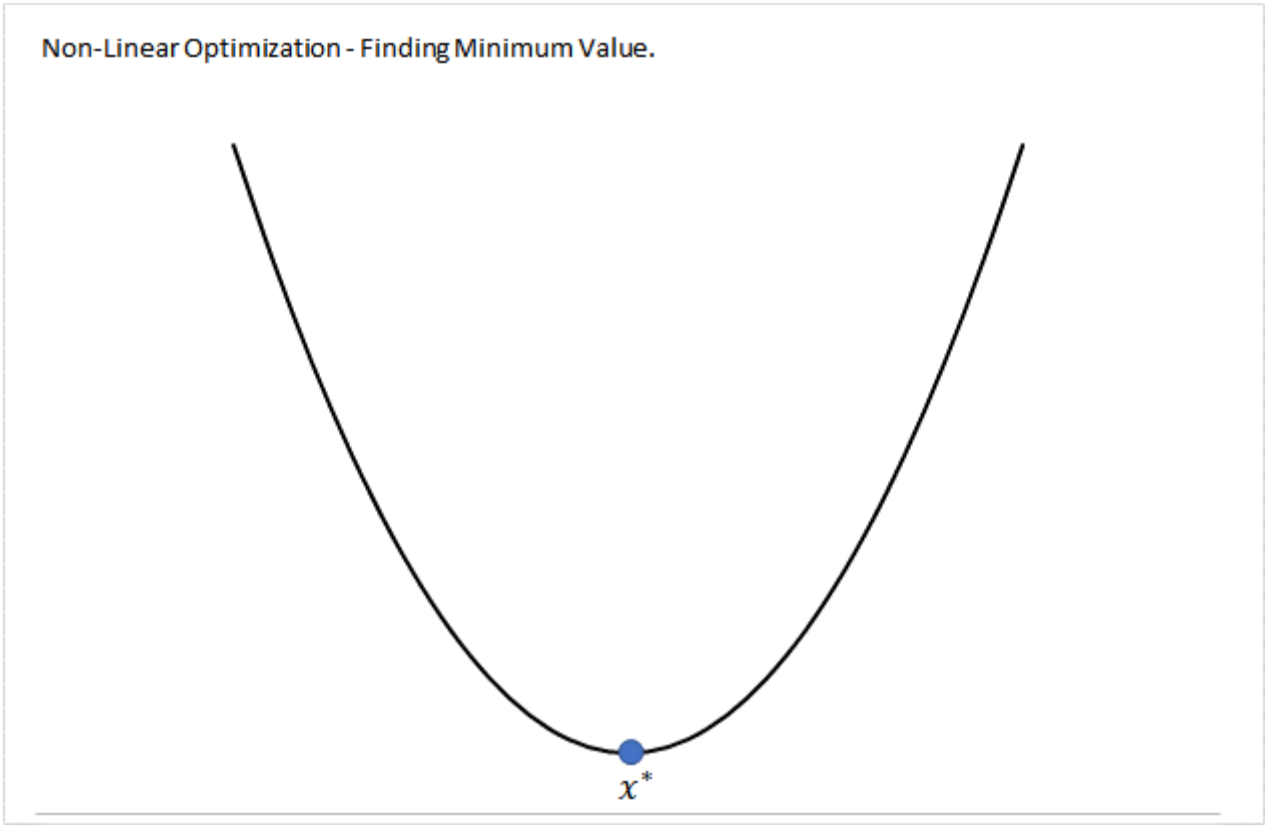
Fun Fact #1

- When did Sports Analytics Begin?
  - Bill James, Sabermetrics
  - Billy Bean, Oakland A's
  - Michael Lewis, Money Ball

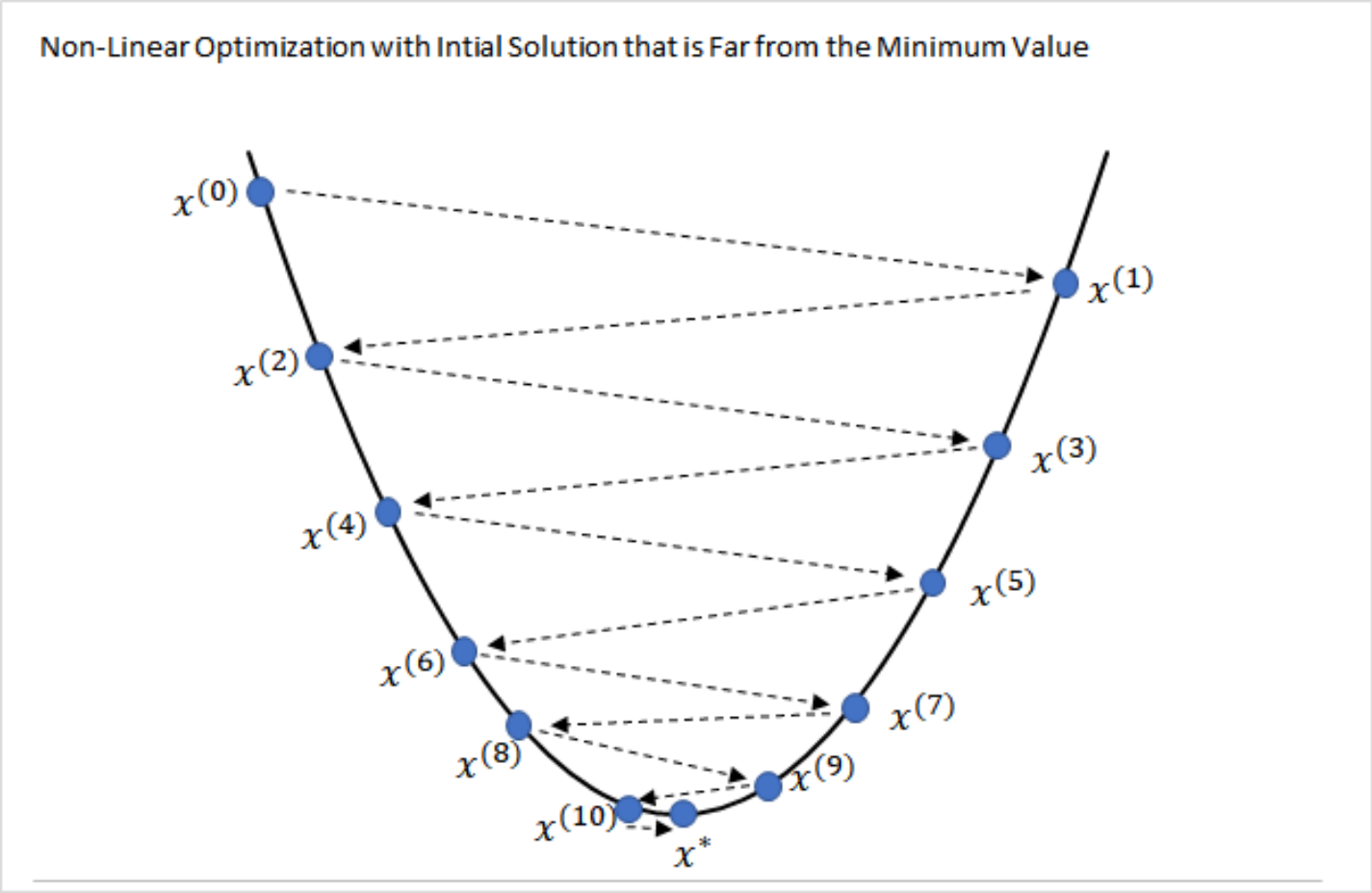
- Answer: Branch Rickey, 1950's
  - GM/Owner, Brooklyn Dodgers
  - Hired Allan Roth, Mathematician
  - Team of Mathematicians working out of a secret lab somewhere in Brooklyn.

Machine Learning Algorithms (Refresher)

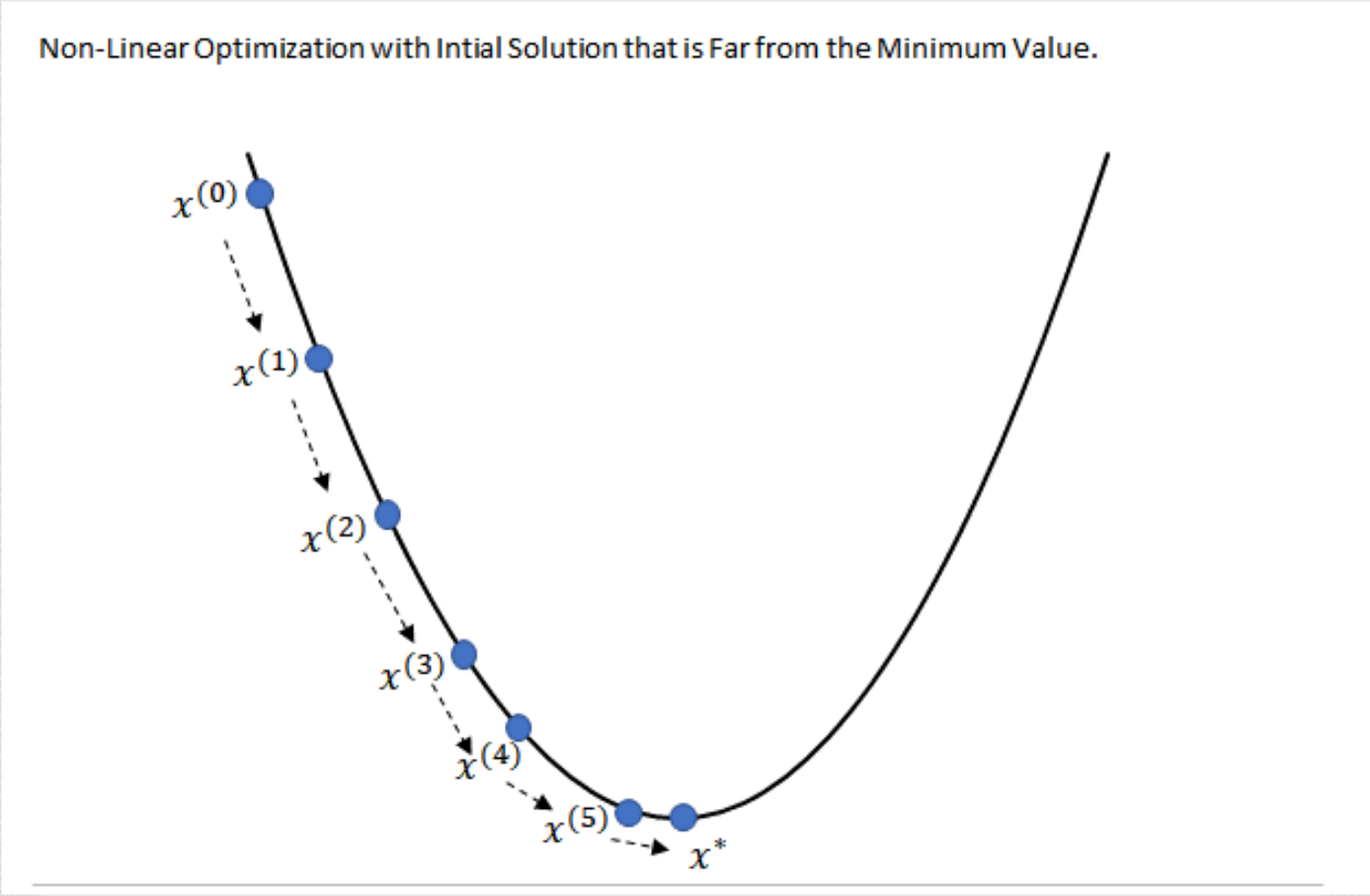
Machine Learning Algorithms



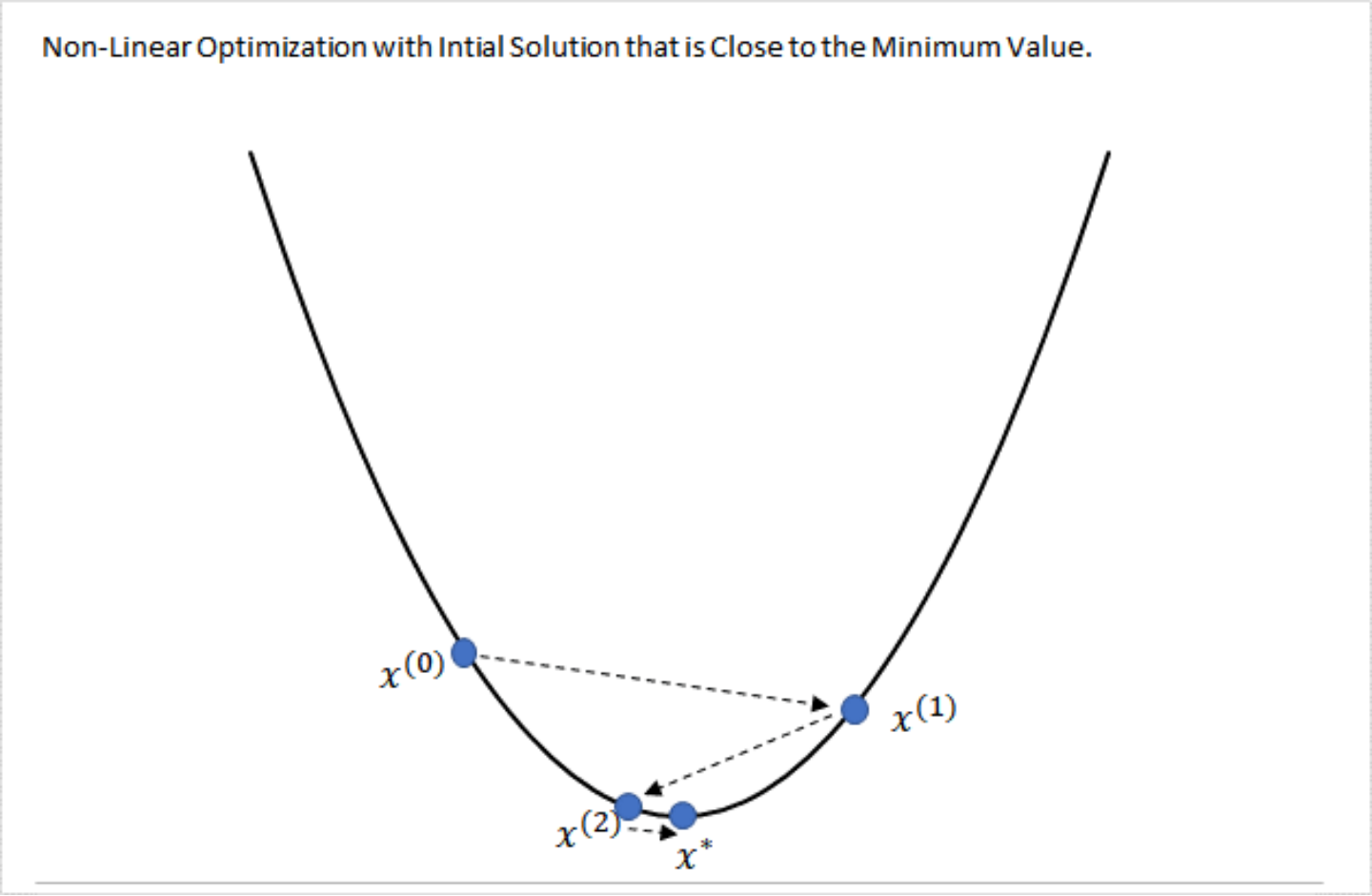
Machine Learning Algorithms



Machine Learning Algorithms



Machine Learning Algorithms



Machine Learning Algorithms

- Newton's Method
- Steepest Gradient Descent

Newton's Method

Find the zeros of a function:

$$f(x) = 0$$

Updating Function:

$$x^{i+1} = x^i - \frac{f(x^i)}{f'(x^i)}$$

Solve for x:

$$x = \sqrt[3]{73}$$

Rewriting:

$$x^3 - 73 = 0$$

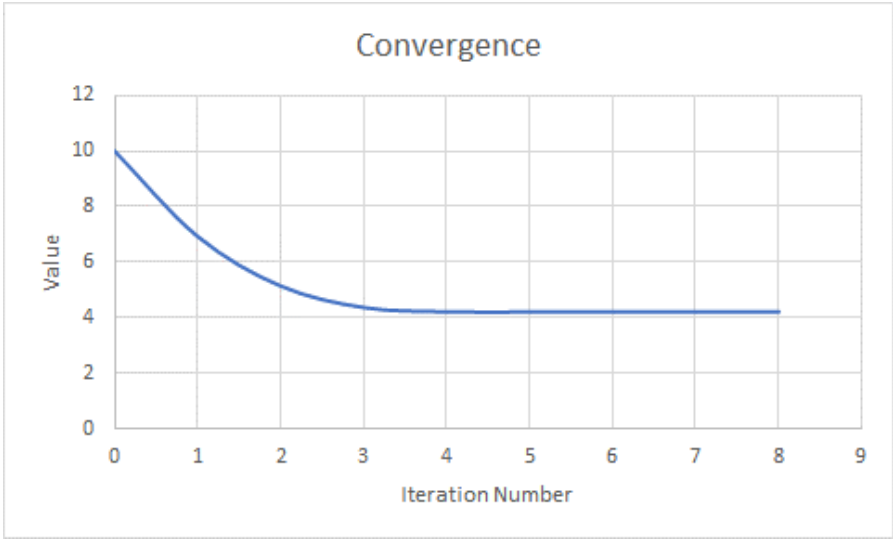
Functions:

$$f(x) = x^3 - 73$$

$$f'(x) = 3x^2$$

Machine Learning: Newton's Method (Cubed Root of 73)

Iteration	x	f(x)	f'(x)	Chg
0	10	927	300	
1	6.91	256.93937	143.24430	-3.09
2	5.11628555	60.92582	78.52913	-1.7937145
3	4.34044834	8.77184	56.51848	-0.7758372
4	4.18524527	0.30992	52.54883	-0.1552031
5	4.17934753	0.00044	52.40084	-0.0058977
6	4.17933920	0.00000	52.40063	-8.331E-06
7	4.17933920	0.00000	52.40063	-1.661E-11
8	4.17933920	0.00000	52.40063	0



Gradient Descent Method

Simple Linear Regression

$$\hat{y}_i = b_0 + b_1 x_i$$

Loss Function:

$$L = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2$$

Partial Derivatives:

$$\frac{\partial L}{\partial b_0} = \frac{1}{n} \cdot \sum_{i=1}^{n} -2 \cdot (y_i - \hat{y}_i) = \frac{-2}{n} \cdot \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)$$

$$\frac{\partial L}{\partial b_1} = \frac{1}{n} \cdot \sum_{i=1}^{n} -2 \cdot (y_i - \hat{y}_i) \cdot x_i = \frac{-2}{n} \cdot \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i) \cdot x_i$$

Machine Learning Update Function:

$$b_k^{i+1} = b_k^i - \alpha \cdot \frac{1}{n} \cdot \frac{\partial L}{\partial b_k}$$

$\alpha = learning\ rate$

Update Function #1:

$$b_0^{i+1} = b_0^i - \alpha \cdot \frac{1}{n} \sum_{i=1}^{n} -2 \cdot (y_i - \hat{y}_i)$$

$$b_1^{i+1} = b_1^i - \alpha \cdot \frac{1}{n} \sum_{i=1}^{n} -2 \cdot (y_i - \hat{y}_i) \cdot x_i$$

Update Function #2:

$$b_0^{i+1} = b_0^i + \alpha \cdot \frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)$$

$$b_1^{i+1} = b_1^i + \alpha \cdot \frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i) \cdot x_i$$

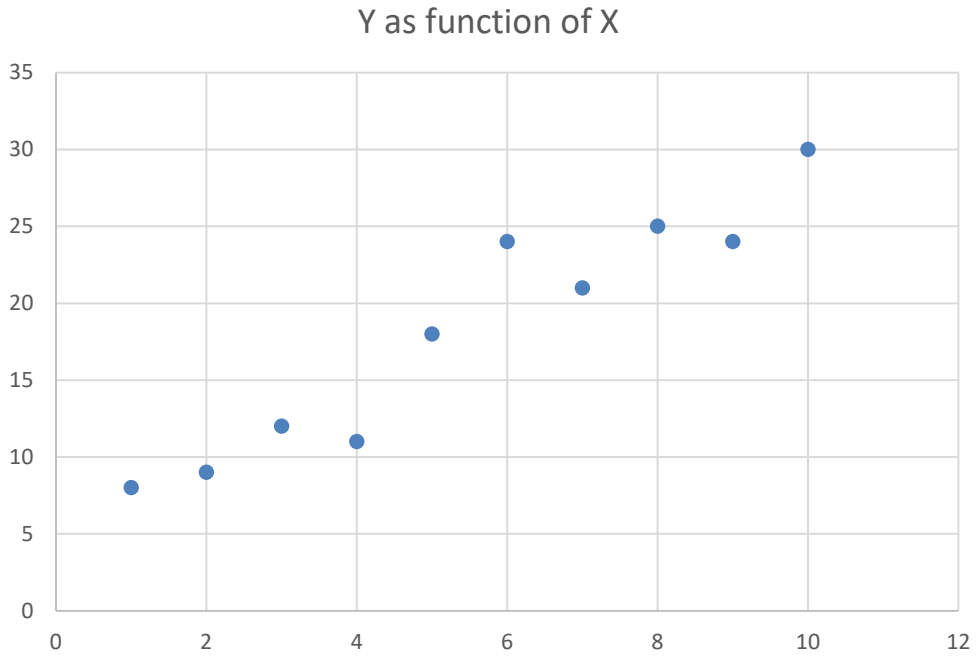
Update Function #3:

$$b_0^{i+1} = b_0^i + \alpha \cdot \frac{2}{n} \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)$$

$$b_1^{i+1} = b_1^i + \alpha \cdot \frac{2}{n} \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i) \cdot x_i$$

Gradient Descent Method: Example

Linear Regression Data		
obs	Y	X
1	8	1
2	9	2
3	12	3
4	11	4
5	18	5
6	24	6
7	21	7
8	25	8
9	24	9
10	30	10



Update Function:

$$b_0^{i+1} = b_0^i + 0.01 \cdot \frac{2}{10}\sum_{i=1}^{10}\left(y_i - b_0^i - b_1^i x_i\right)$$

$$b_1^{i+1} = b_1^i + 0.01 \cdot \frac{2}{10}\sum_{i=1}^{10}\left(y_i - b_0^i - b_1^i x_i\right) \cdot x_i$$

Starting Solution:

$b_0 = 2.5$

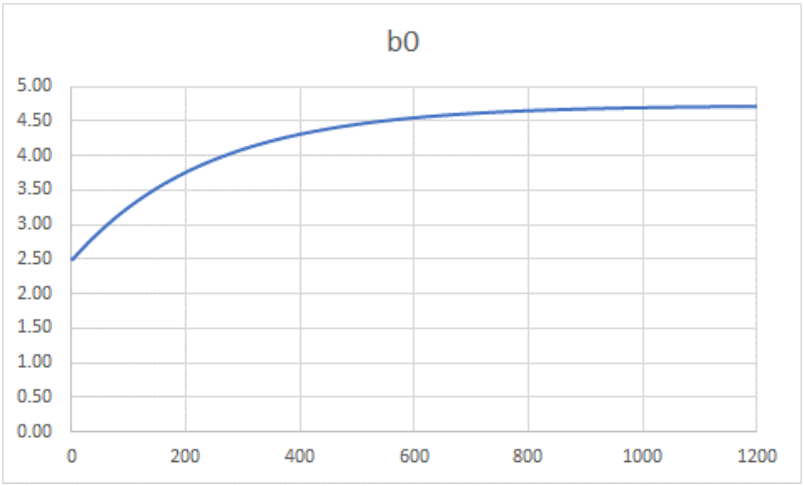
$b_1 = 3.0$

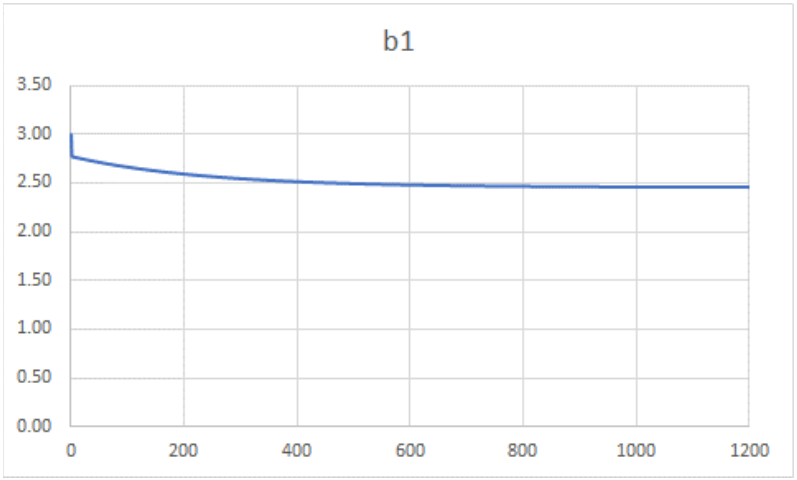
Learning Rate:

$\alpha = 0.01$

"Molloy Means Business"

Gradient Descent Method: Example



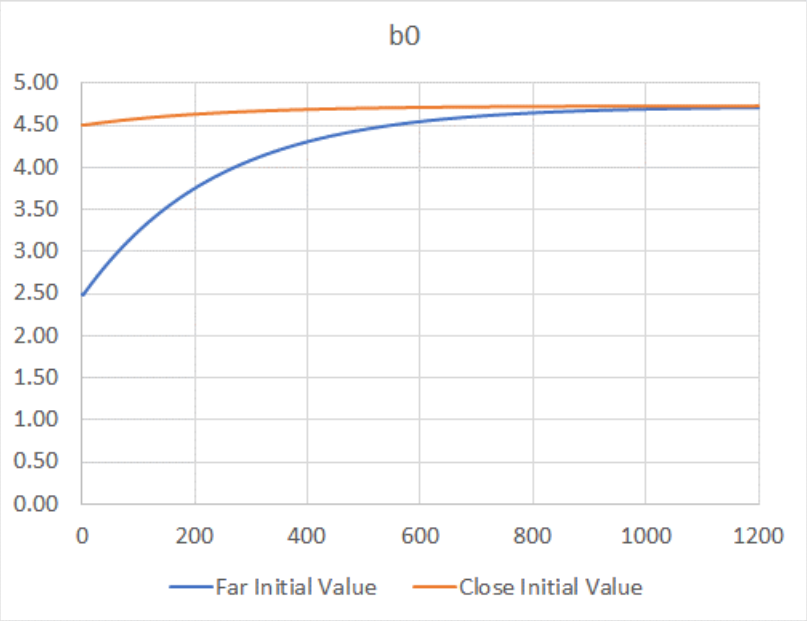


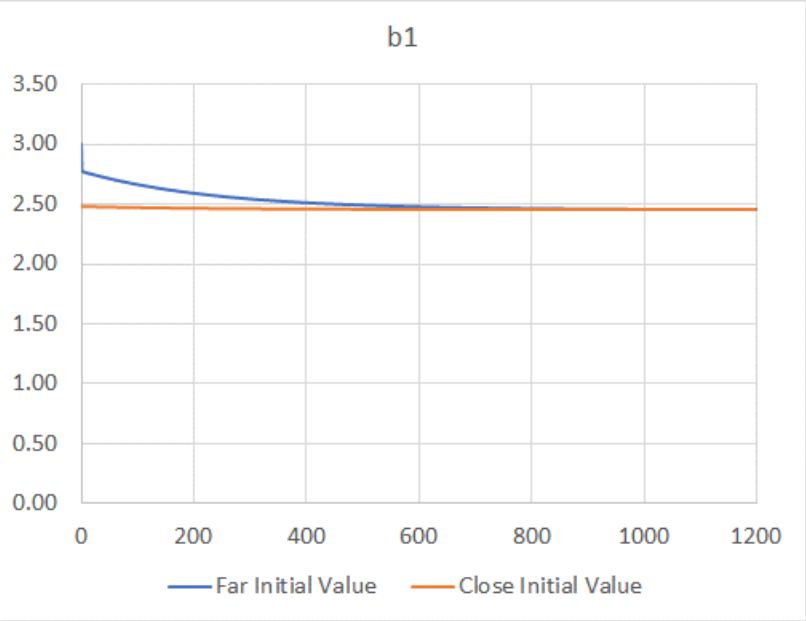
Iterative Result Using Gradient Descent Method

Iteration	b0	b1		Iteration	b0	b1
0	2.5000000	3.0000000		1190	4.7182513	2.4506512
1	2.4880100	2.7815900		1191	4.7183147	2.4506421
2	2.4962749	2.7720846		1192	4.7183777	2.4506331
3	2.5054201	2.7689892		1193	4.7184406	2.4506241
4	2.5147229	2.7672713		1194	4.7185031	2.4506151
5	2.5240286	2.7658529		1195	4.7185654	2.4506061
6	2.5333042	2.7645030		1196	4.7186274	2.4505972
7	2.5425428	2.7631722		1197	4.7186892	2.4505883
8	2.5517430	2.7618499		1198	4.7187507	2.4505795
9	2.5609046	2.7605338		1199	4.7188119	2.4505707
10	2.5700278	2.7592233		1200	4.7188729	2.4505620

	b0	b1
Initial Value	2.50	3.00

Gradient Descent Method: Example

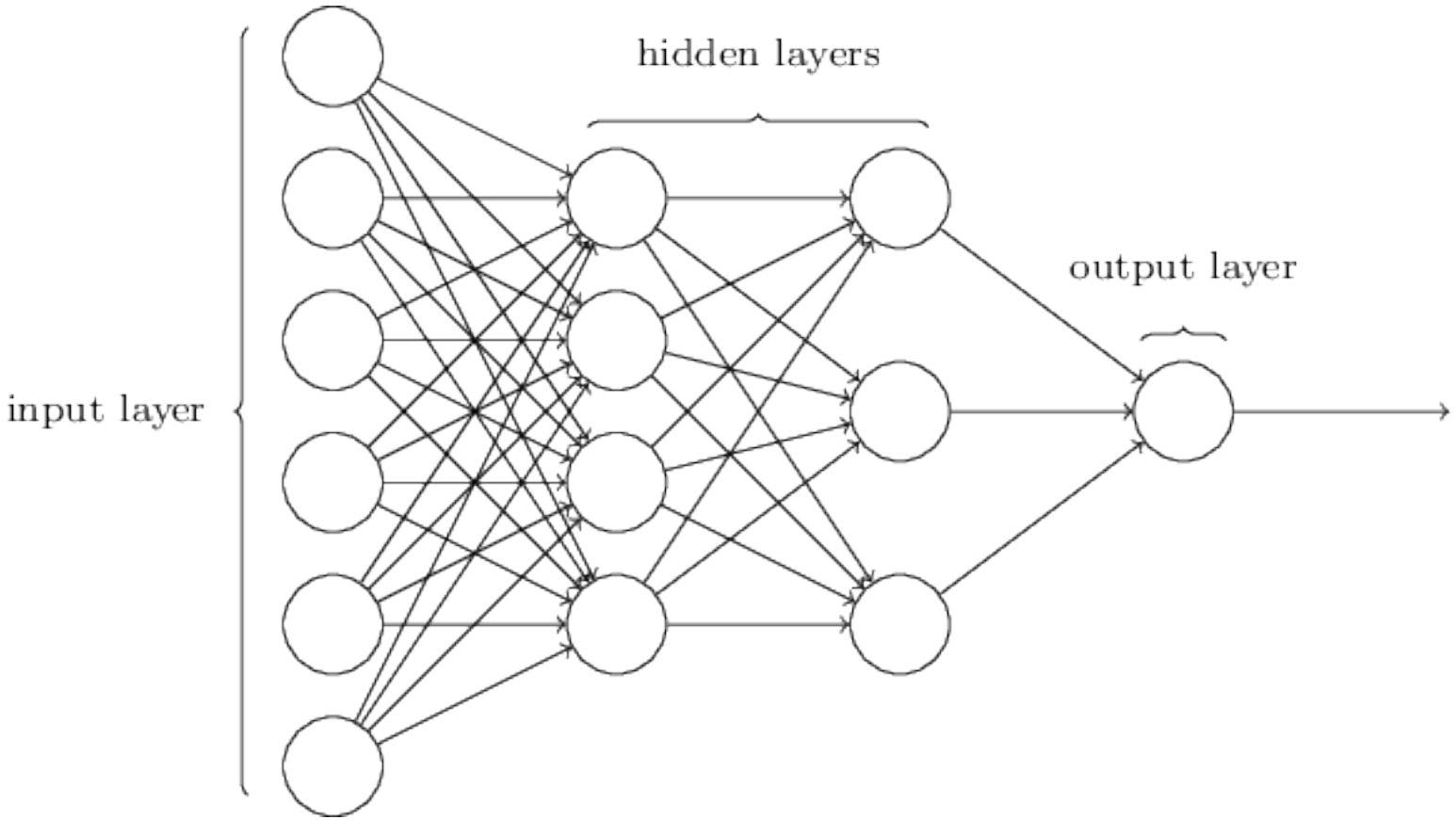




	b0	b1
Far Initial Value	2.50	3.00
Close Initial Value	4.50	2.50

Neural Networks

Neural Network



Experiment:

Multi-Period Trade Schedule Optimization using Machine Learning

Question:

Question:

- Can Machine Learning (Neural Networks) be used to find a better initial starting solution for Non-Linear Optimization?

Question:

Question:

- Can Machine Learning (Neural Networks) be used to find a better initial starting solution for Non-Linear Optimization?

Answer:

- Yes We Can!

Experiment:

Data Generating Process:

- 21 Basket Sizes ($n = 10, 25, 50, 75, \ldots, 450, 475, 500$).
- 100 Simulated Trades for each.
- 2,100 different baskets - 21 different basket sizes and 100 optimizations.
- Total of 526,000 "x-input" samples.
  - Each of the 21 baskets had a different number of stocks.

- Orders Sizes: 0.01% ADV to 50% ADV.
- One-Sided and Two-Sided Baskets
  - Two-Sided: 10% Buy and 90% Sell,…., 90% Buy and 10% Sell

- Compiled statistics for (18) different "x-variables" for each stock.

- Solved Multi-Period Trade Schedule Optimizer
  - Solved for Optimal POV Rate
  - This Optimal POV Rate is then used in the NNET Training

Experiment:

Stepwise Regression Approach:

- Step 1: select one of the fifteen variables to include in the NNET training
- Step 2: determine the best NNET structure, e.g., determine number of layers and nodes that provide the best results for the input variables.
- Step 3: record the error term, e.g., the standard deviation of the difference between the actual y and the estimated y from the neural network.
- Step 4: repeat for each input variable.
- Step 5: determine which variable was the best to include based on the lowest error.

Experiment:

Finding Important Variables

- Once we determine the best input variable that input variable was always included as one of the input factors.
    - Repeated Step 1 – Step 5 to find the next most important variable.
    - Added one variable at a time.
- In total, we trained 120 different NNET to find the best set of input variables.
- For each training, we had to determine the NNET structure that provided the lowest error term.
- On average, we trained 5 NNET structures for each data set.
    - Resulted in 500 different NNET training scenarios.
- Our machine learning experiment, was quite time consuming.

Neural Network Data:

Input Data

- Order Size
- Volatility
- Side = 1, -1
- Lambda
- “a4” Model Parameter
- IStar
- Temp MI
- Perm MI
- Timing Risk (VWAP)

Input Data

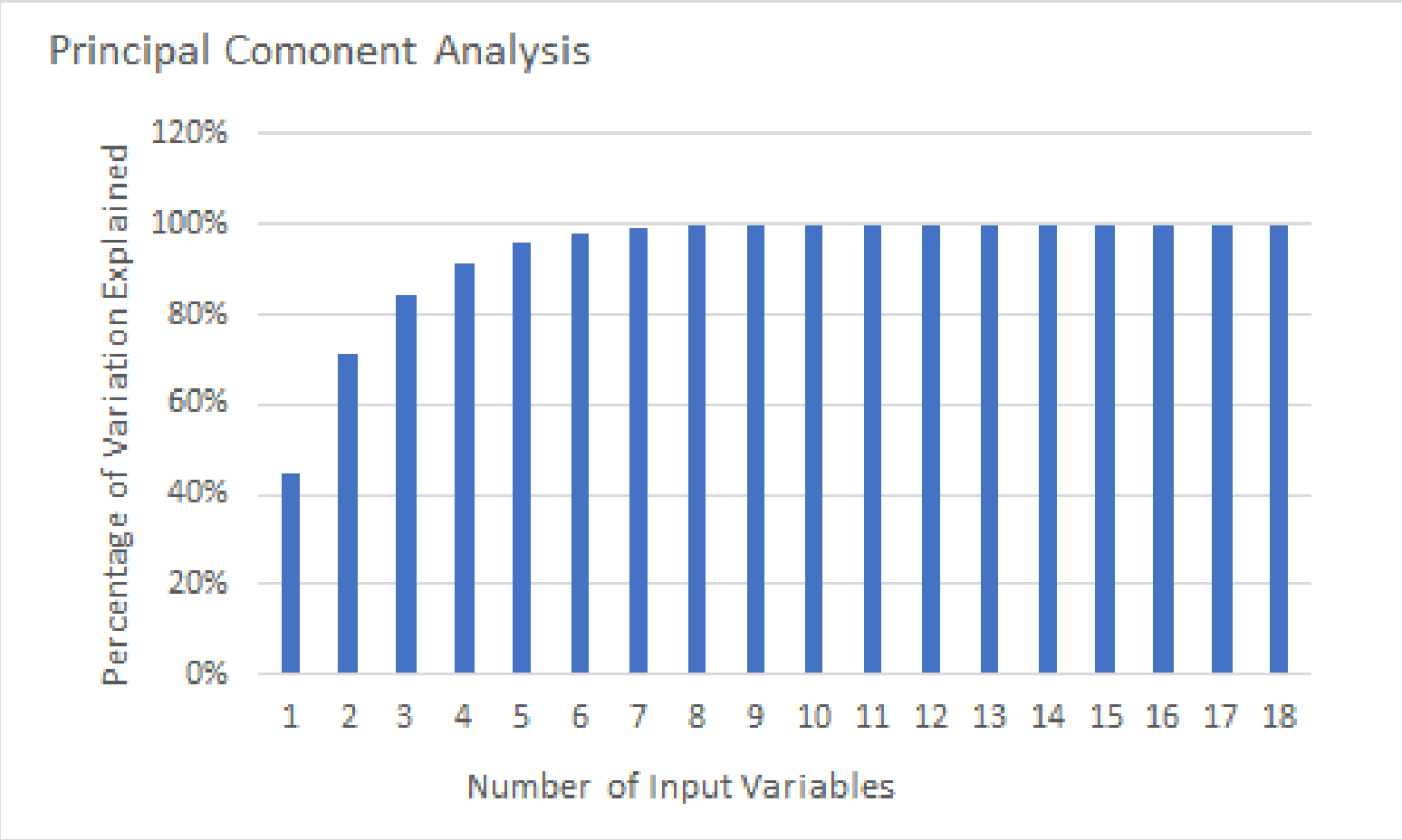
- Risk Reduction
- RC
- MCR
- Excess Size
- Excess Volatility
- Excess Weight
- Excess Price
- SS POV
- Number Stocks = 1/N

Output Data:

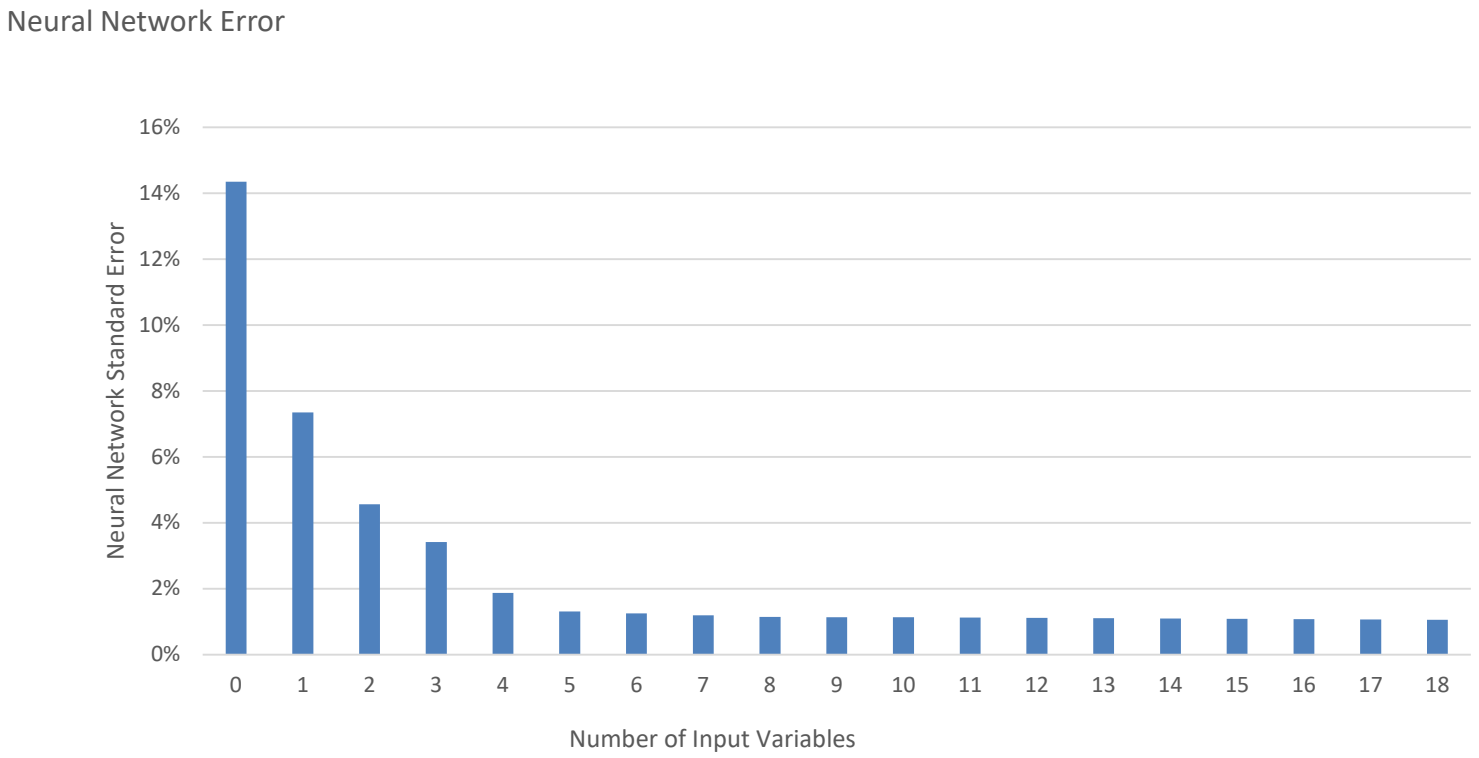
Y = Optimal POV Rate

(Determined from Optimization Routine)

Principal Component Analysis



Best Fit NNET Model



Results

Best Predictive Variables

Input Data:

1. Lambda
2. IStar
3. Timing Risk
4. Risk Change
5. Risk Contribution
6. Excess Size
7. Excess Volatility
8. Excess Weight

NNET Structure:

The best NNET performance consisted of eight (8) input factors and three hidden layers.

- First Hidden Layer = 4 nodes and Inverse Tangent function.
- Second Hidden Layer = 8 nodes and Inverse Tangent function.
- Third Hidden Layer = 4 nodes and Sigmoid function.

Best Fit NNET Model

Comparison of NNET Initial Trade Rate Solution to Optimal Trade Rate Solution

Experiment (Part 2): Calculating Solution Time

Calculating Solution Time Improvement

Step I:

For each of the 2,100 baskets we solved the Multi-Period Trade Schedule Optimization Problem using a random initial starting value.

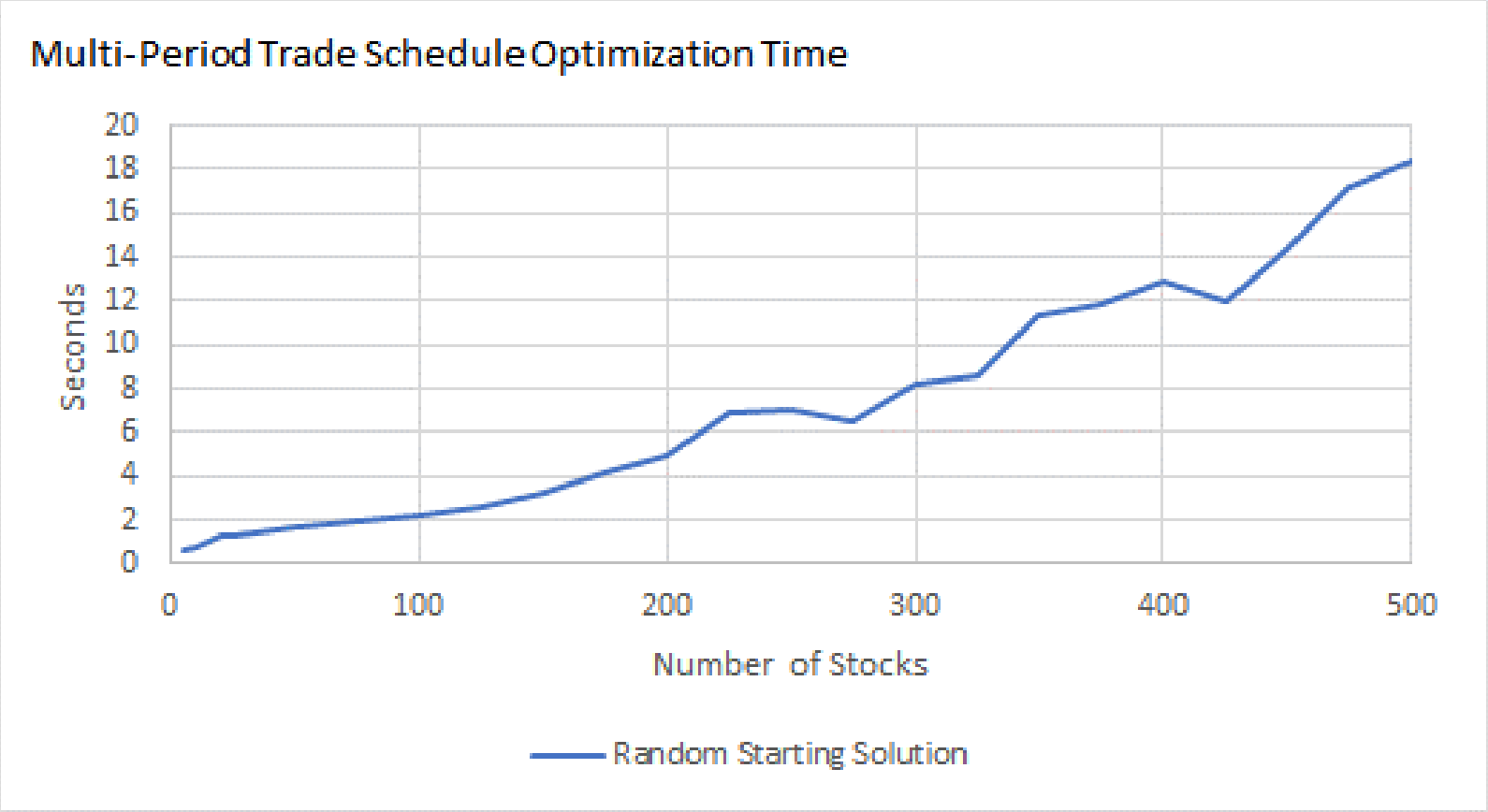
- We recorded the calculation Time

Step II:

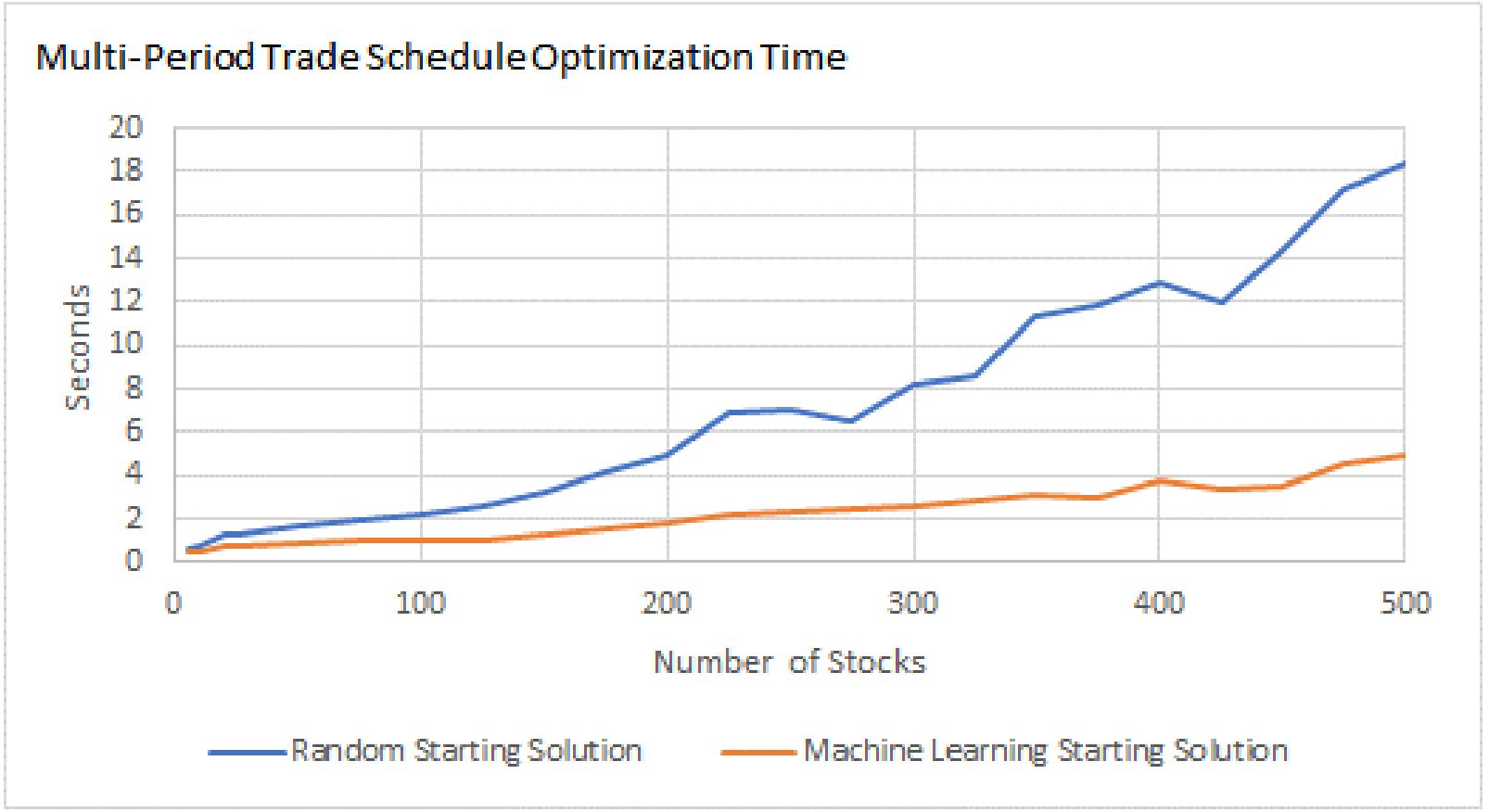
For each of the 2,100 baskets, we solved the Multi-Period Trade Schedule Optimization Problem using the NNET solution as the initial starting value.

- We recorded the calculation Time.

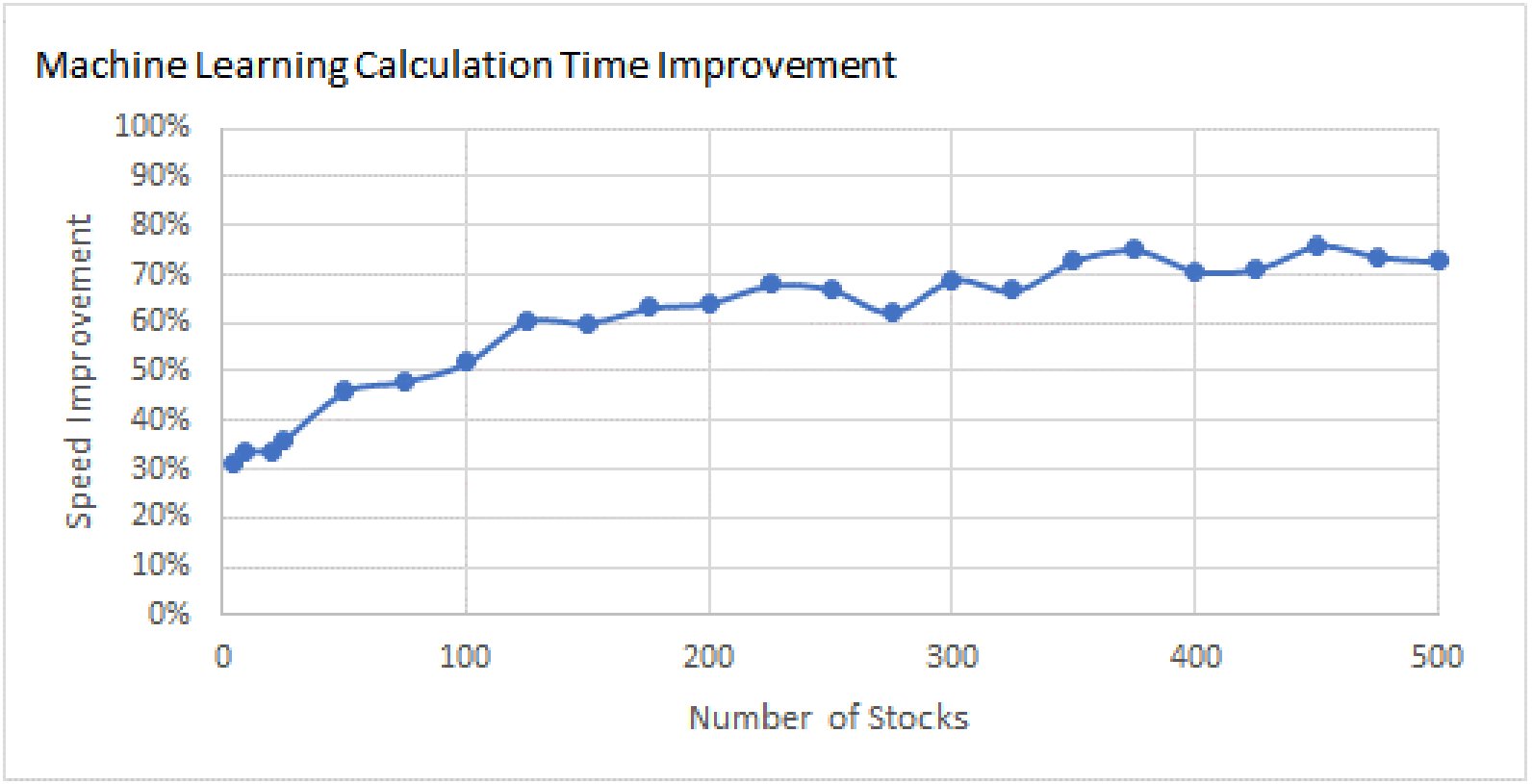
Trade Schedule Optimization – Calculation Time



Trade Schedule Optimization – Calculation Time



Best Fit NNET Model



Conclusions

Conclusions

- Combining Machine Learning (NNET) with Multi-Period Trade Schedule Optimization to determine the starting solution provided an improvement in calculation speed of:
  - 30% faster for portfolios with n=10 stocks
  - 50% faster for portfolios with n=100 stocks
  - 70% faster for large portfolios (e.g., 300-500 stocks).

- These improvements in calculation time will allow investors to react to profitably market conditions and take advantage of favorable market conditions when they arise in the marketplace.

Questions ???

References

- Almgren, R., and N. Chriss, (2000) "Optimal execution of portfolio transactions", Journal of Risk, Vol. 3 pg. 5-39.
- Freeman, J. (1994), “Simulating Neural Networks with Mathematica,” Addison-Wesley Publishing, 1994.
- Kissell, R., M. Glantz, R. Malamut (2004), "A Practical Framework for Estimating Transaction Costs and Developing Optimal Trading Strategies to Achieving Best Execution," Finance Research Letters 1, Elsevier, Winter 2004, Vol. 1, No. 1, pg. 35 – 46.
- Kissell, R. (2006). “Algorithmic Trading Strategies,” Fordham University, May 2006, Ph.D. Dissertation, www.il.proquest.com, ISBN 978-0-542-67789-2.
- Kissell, R. (2011), “Creating Dynamic Pre-Trade Models: Beyond the Black Box,” Journal of Trading, Fall 2011, Vol. 6, No. 4, pg. 8-15.
- Kissell, R. (2013), “The Science of Algorithmic Trading and Portfolio Management” Elsevier/Academic Press.
- Kissell, R., Jungsun “Sunny” Bae (2018), “Machine Learning for Algorithmic Trading and Trade Schedule Optimization,” Journal of Trading, Fall 2018, Vol. 13, No. 4 Pg. 138-147.
- Malamut, R. (2002), “Multi-Period Optimization Techniques for Trade Scheduling,” QWAFAFEW, April 2002.
- Schmidhuber, J. (2015), “Deep learning in neural networks: An overview,” Neural Networks, Volume 61, January 2015, pg. 85-117. Elsevier.