A NO FREE LUNCH RESULT FOR OPTIMIZATION AND ITS IMPLICATIONS

Marisa B. Smith

Advisor: Dr. Jeffrey Jackson
Department of Mathematics & Computer Science
Duquesne University

Thesis Presentation
May 5, 2009

Outline

- Motivation
- Introduction/Background
- NFL Theorems for Optimization
- Result 1: A New NFL Theorem
- Result 2: A Superior Choosing Procedure
- Conclusion/Future Work

Motivation

- No Free Lunch Theorems for Learning
  - On the rationality of belief in free lunches in learning
    - J. C. Jackson and C. Tamon
    - Unpublished manuscript-in-preparation
  - Apply similar ideas to the NFL theorems for optimization
- Address misinterpretation of NFL results
  - No Free Lunch Theorems for Optimization
    - D. H. Wolpert and W. G. Macready
    - 1997

Introduction

- Combinatorial Optimization
  - Functions (problems) in which a finite search space *X* maps to a finite space of cost values *Y*
- Typical Goal of Optimization
  - Find maximum (or minimum) of a function
  - Search for large (or small) cost values
- Optimization Algorithm
  - Some method of choosing *x*'s in *X* in order to meet this goal

Interests in Optimization

- Performance comparison of different optimization algorithms
  - On average, how well do different algorithms do
  - Which algorithms are "better" than others
- In this paper, interested whether there exist algorithms that, on average, are better than random

Background on NFL Theorems

- Mathematically, when averaged over all possible optimization problems, the performance of any pair of optimization algorithms is statistically equivalent [WolMac97]
- What Wolpert and Macready infer from this
  - Instances of good performance are *necessarily* offset by instances of poor performance
    - "no free lunch"
  - On average, hill-climbing is no better than hill-descending
  - On average, hill-climbing is no better than random guessing
  - On average, no algorithm is better than random guessing

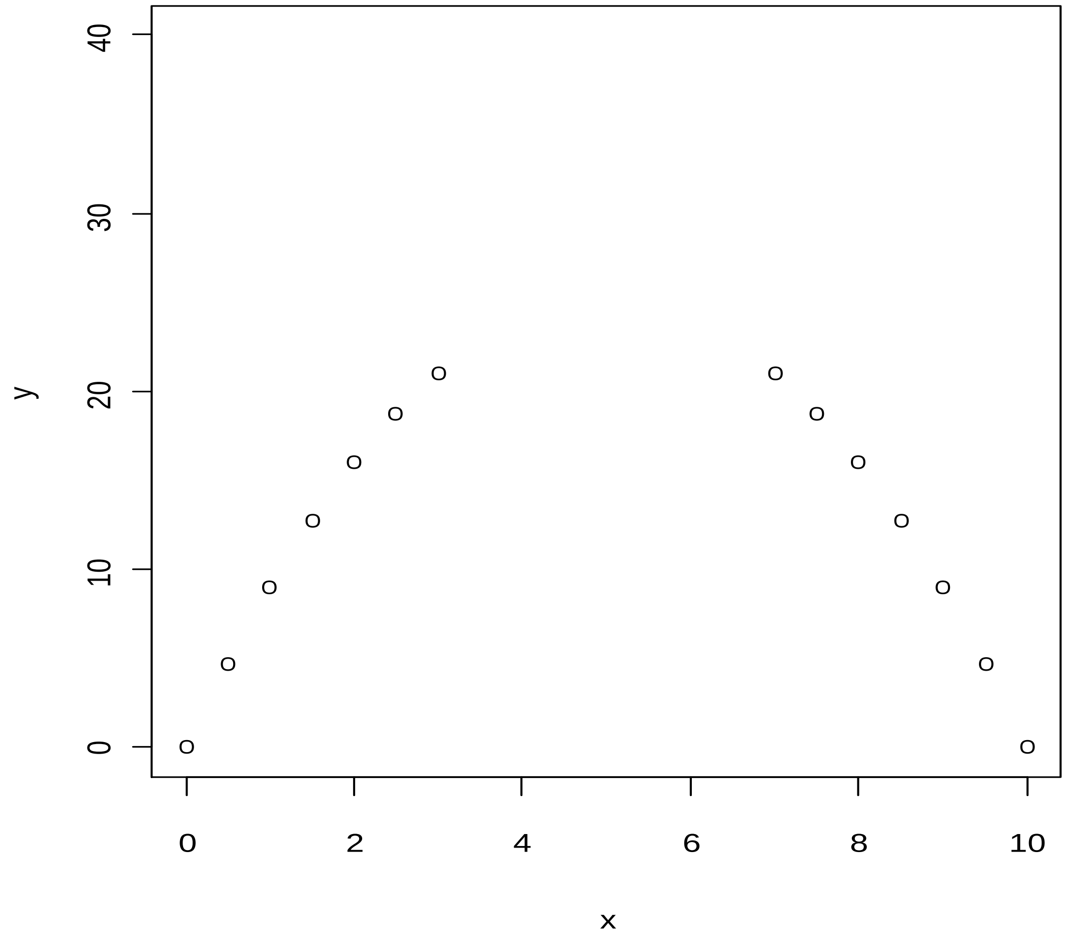
Objective of Present Study

- Result 1
  - Extend NFL theorem
    - Seems to imply that no choosing procedure better than random
- Result 2
  - Give reason to question this inference
  - Use probability theory and concepts in cryptography
  - Implications of NFL theorem are not as negative as expected

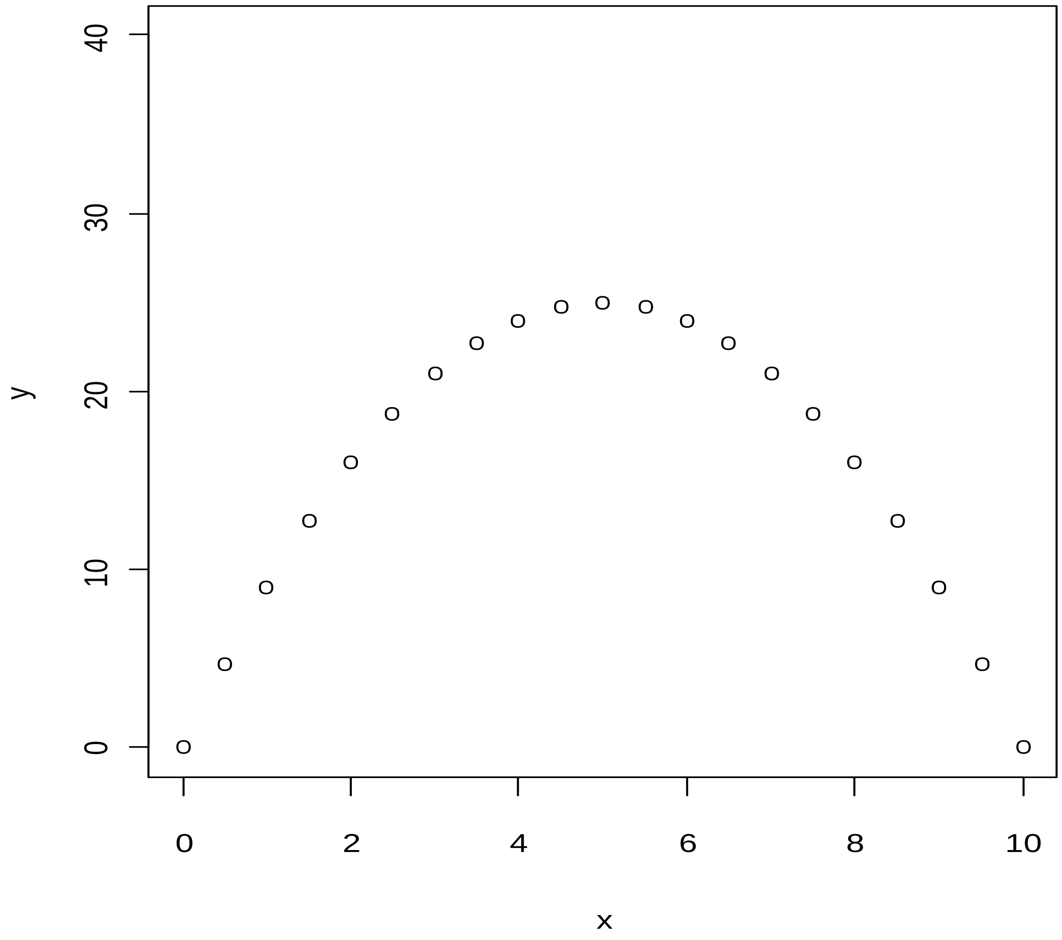
Some Intuition on Why the NFL Theorems Hold

- Averaging over *<u>all possible</u>* problems (functions)
  - Mathematically, when averaged over all possible optimization problems, the performance of any pair of optimization algorithms is statistically equivalent [WolMac97]
- On unknown function, past performance of an algorithm tells us nothing about future performance
  - "Good" algorithm can suddenly perform badly
  - "Bad" algorithm can suddenly perform well

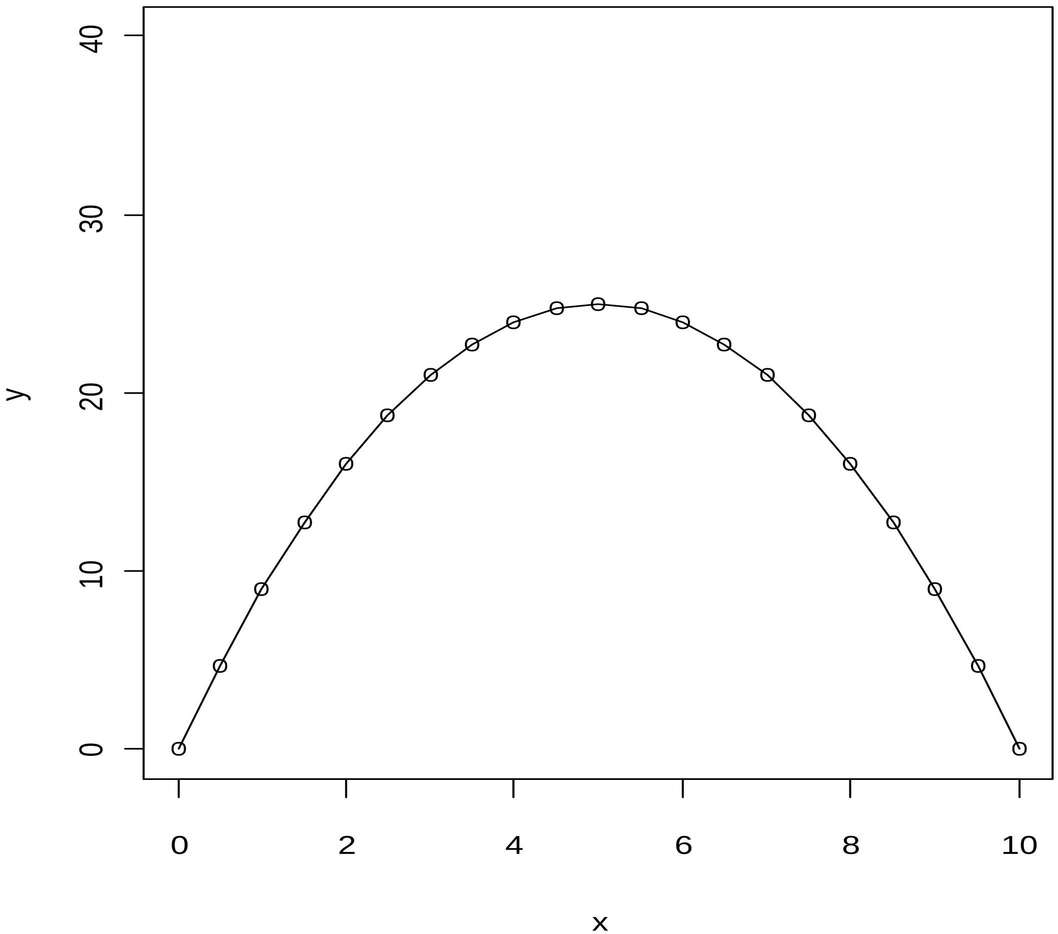
Points in Initial Search



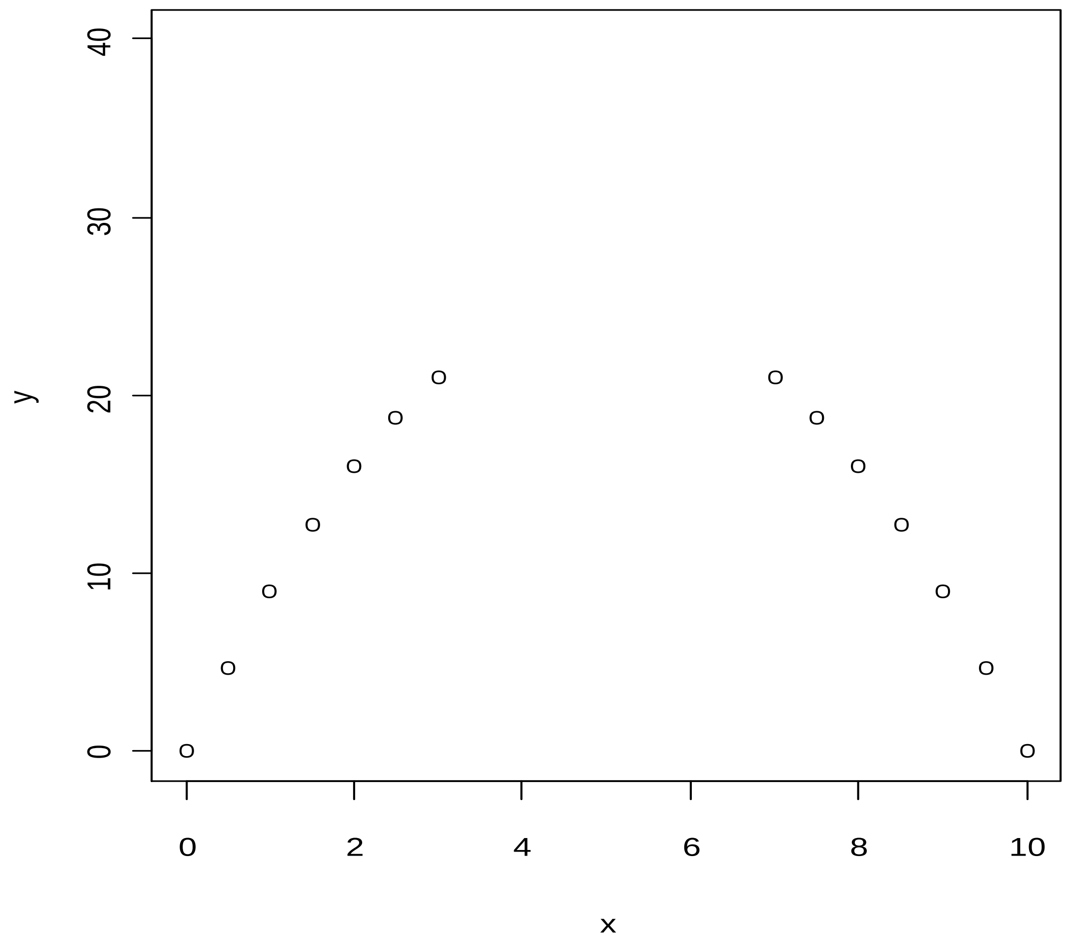
Possible Points in Continuation of Search 1



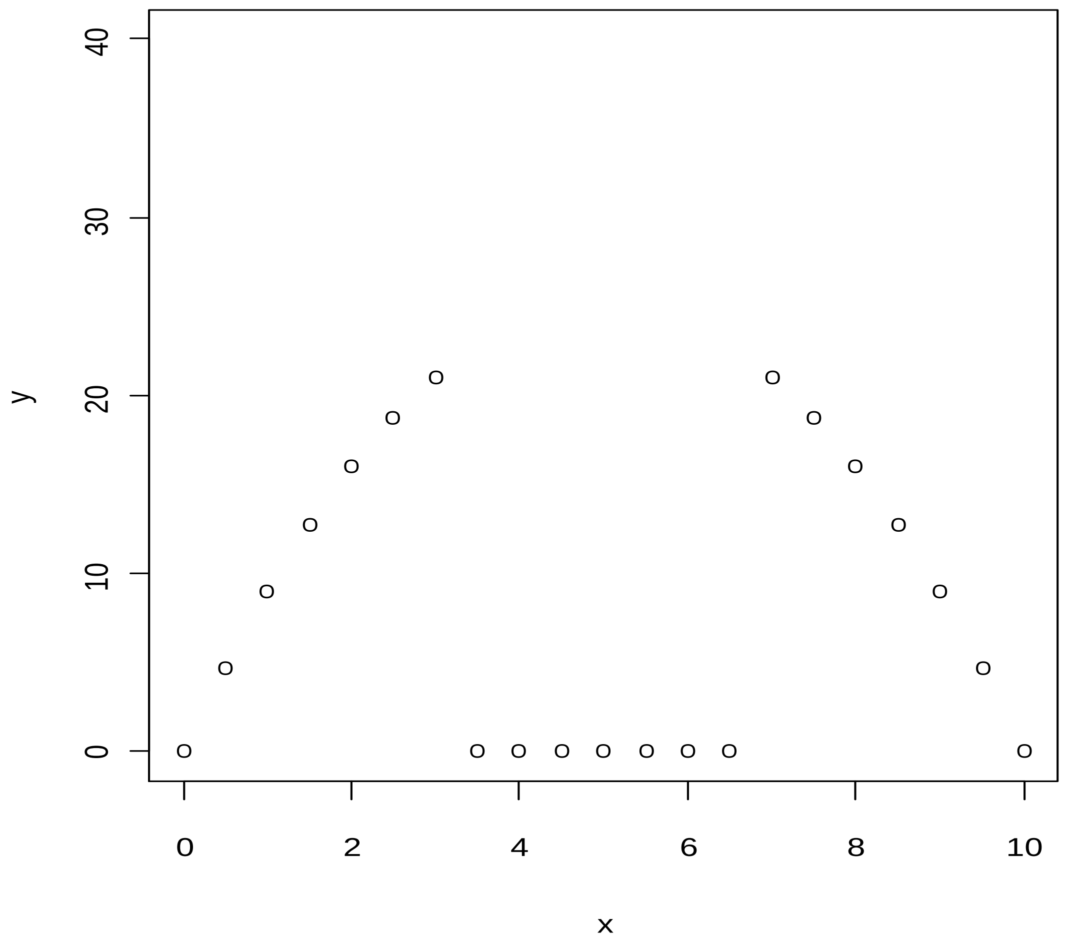
Actual Function 1



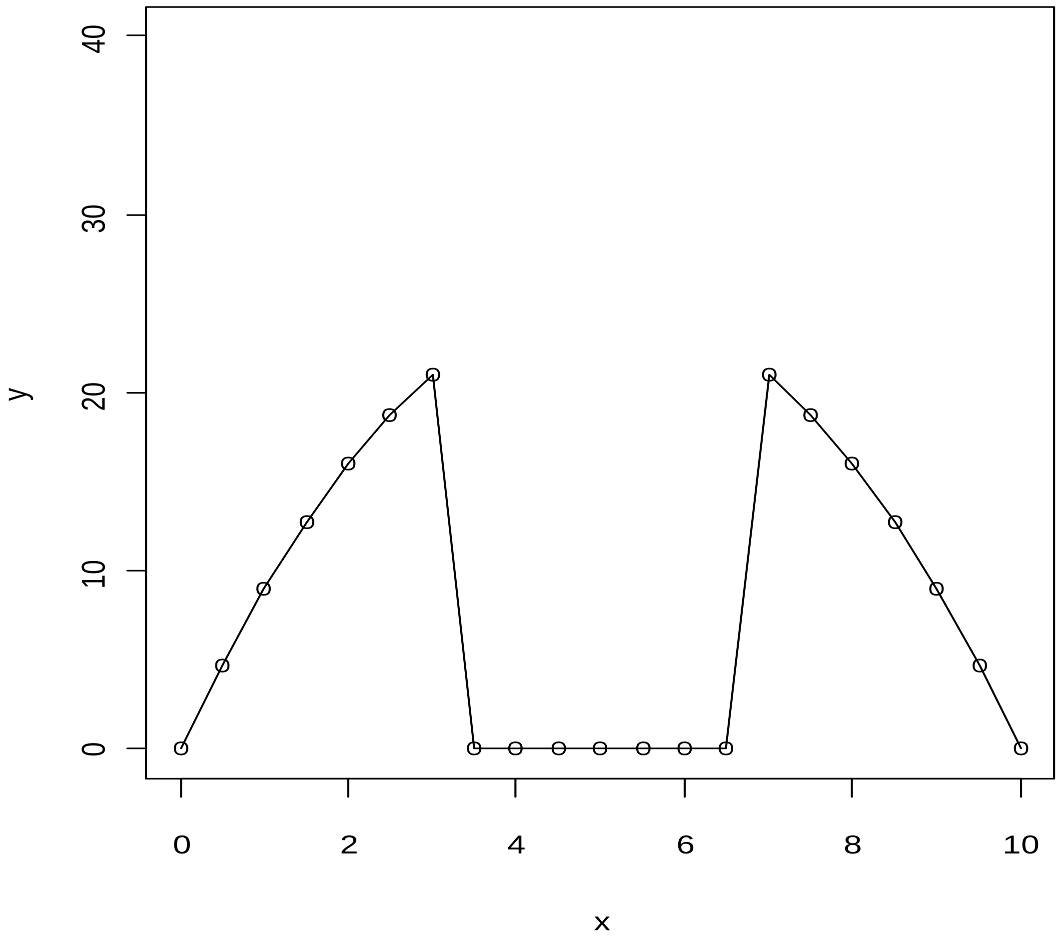
Points in Initial Search



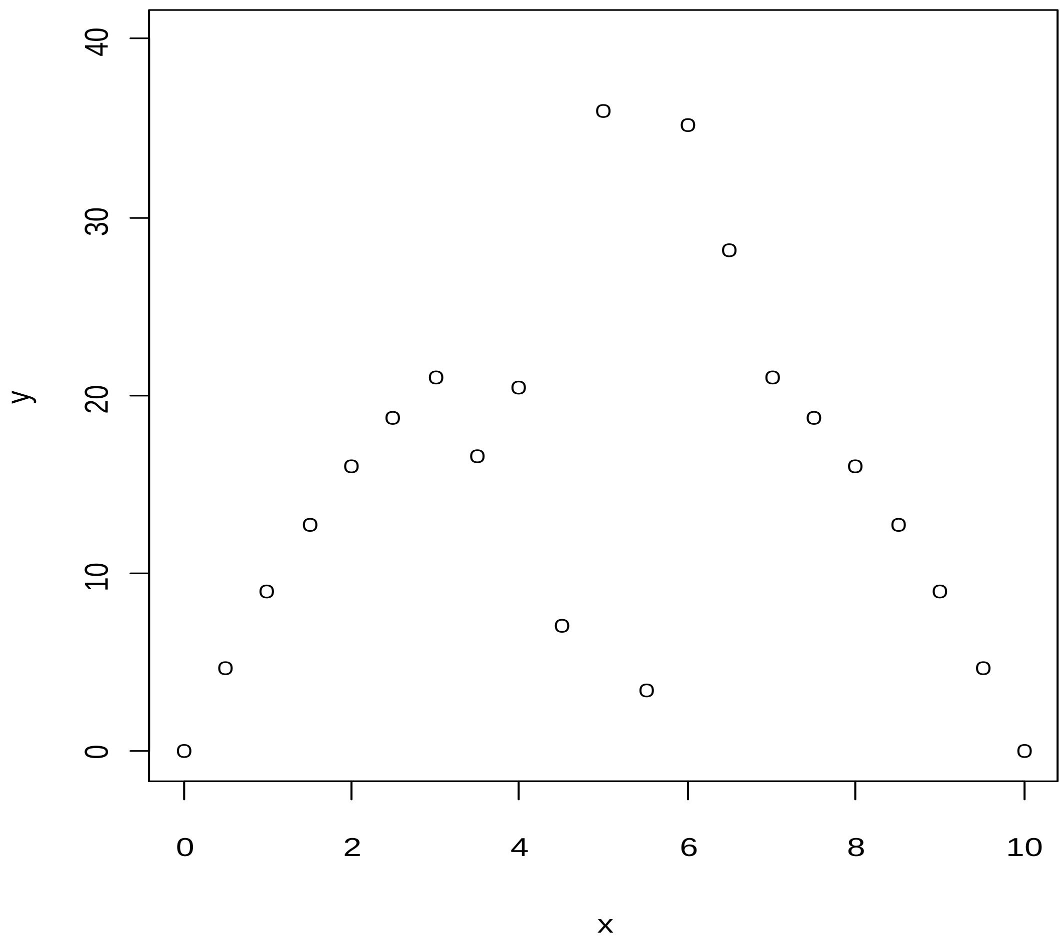
Possible Points in Continuation of Search 2



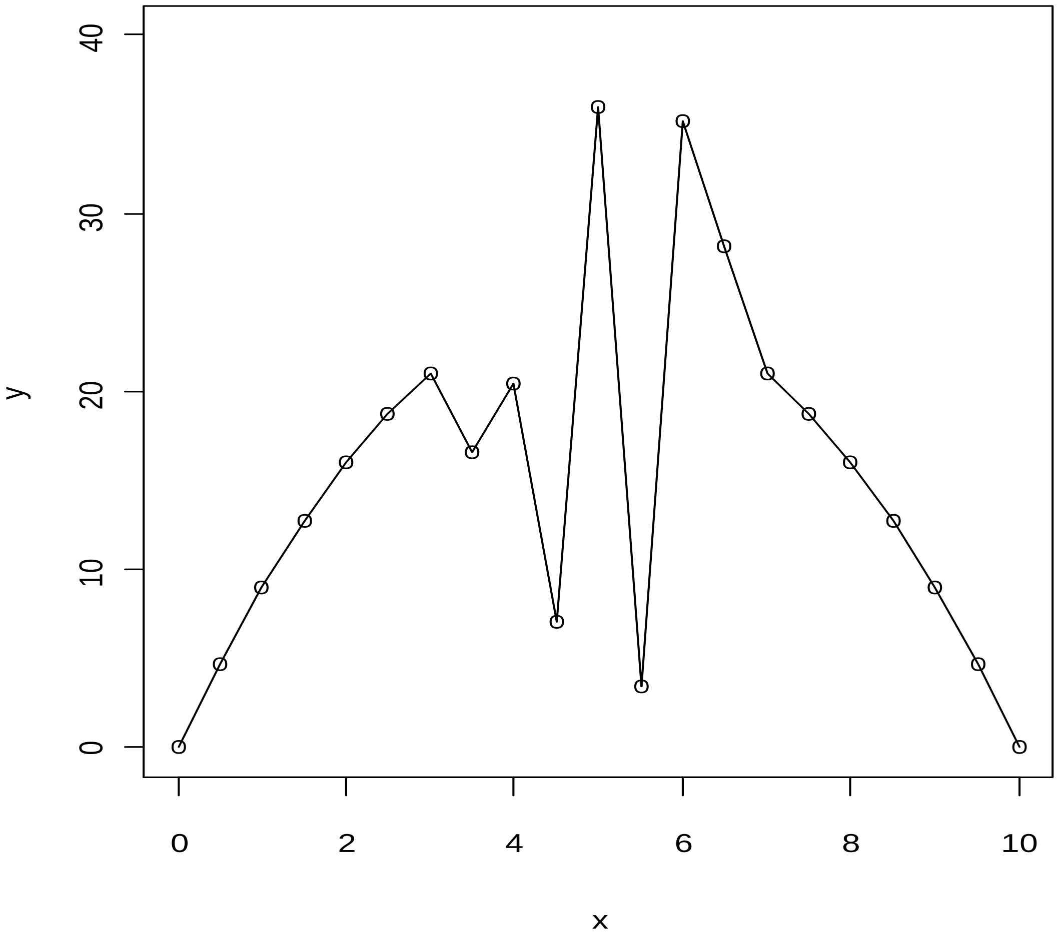
Actual Function 2



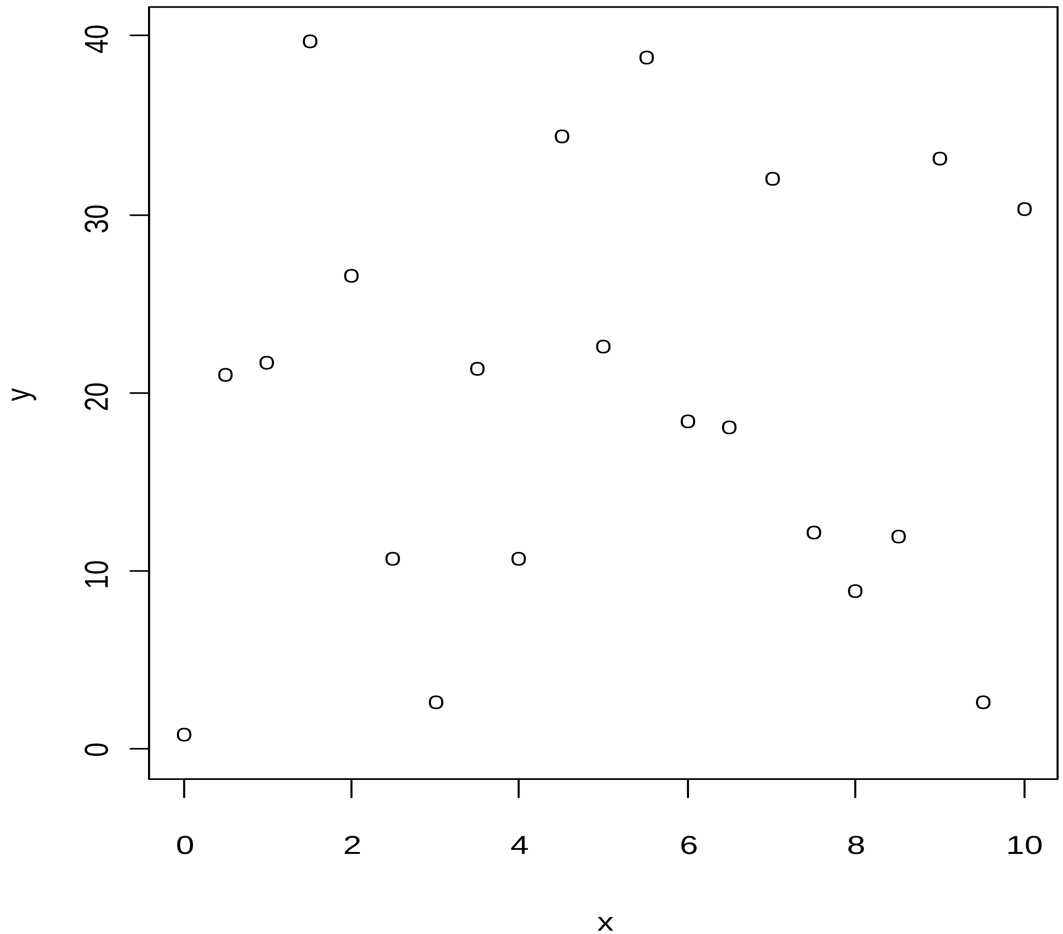
Possible Points in Continuation of Search 3



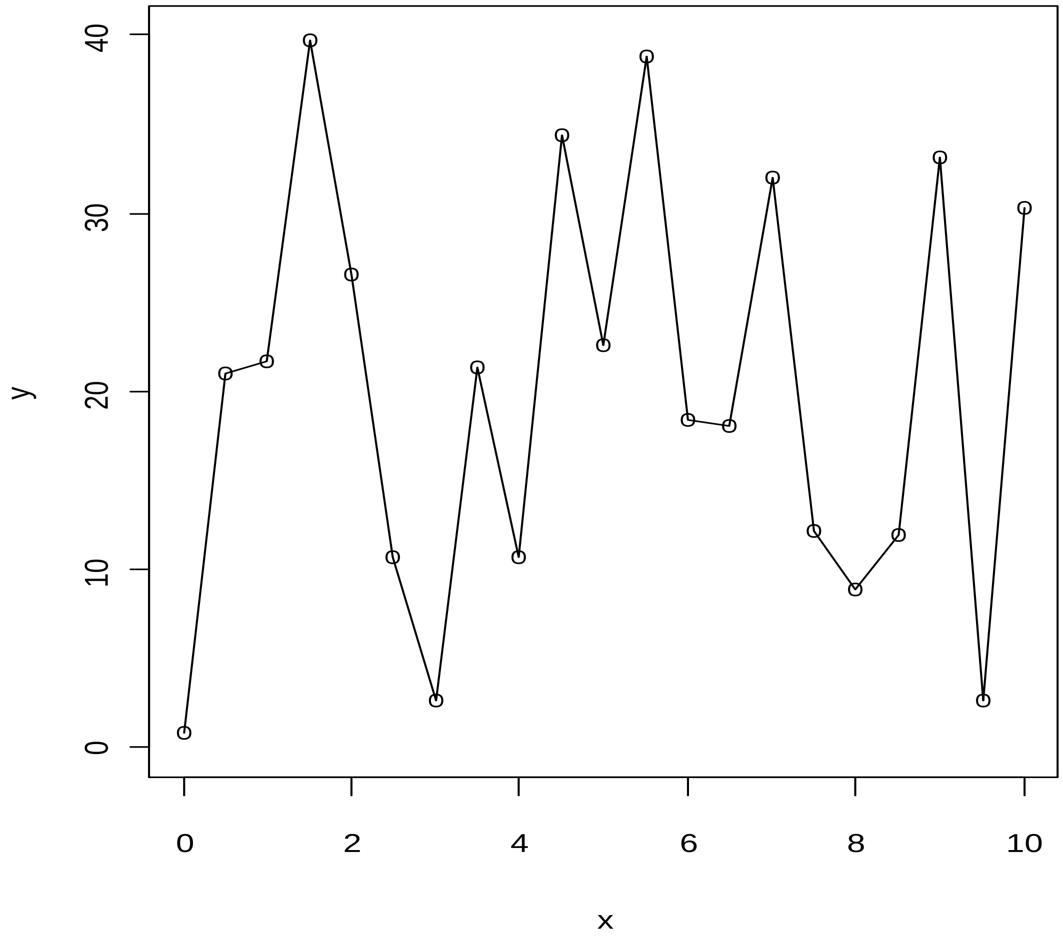
Actual Function 3



Random Points



Random Function



Some Intuition on Why the NFL Theorems Hold

- Algorithm initially finds “good” points
  - Depending on actual function
    - Can continue to find good points
    - Can start to go to bad points
    - Can go anywhere
- Algorithm initially finds “bad” points, same possibilities

Some Intuition on Why the NFL Theorems Hold

- Key point: Averaging over *all possible* functions
  - After initial search, next steps an algorithm takes could lead anywhere if all possible functions considered
  - This is true of all algorithms
    - All algorithms: set of searched (x,y) values, select next x
    - For some function, selected x-value takes on each possible y-value
    - Averaging over all of these possibilities
  - When averaging over all functions, algorithm performance is the same

A Particular NFL Theorem of Interest

- Choosing Procedure NFL Theorem [WolMac97]
- Choosing Procedure
  - Meta-algorithm that compares performance of two algorithms after *m* steps
  - Chooses one of the algorithms to use for continuation of search
- Theorem: Averaged over all possible algorithm pairs, performance of any two choosing procedures is equivalent
  - There is no free lunch for choosing procedures

Preliminaries

- *Sample* from an algorithm run (denoted *d*)
  - The (x,y) pairs the algorithm visits in its search
- Optimization algorithm
  - Mapping from previously visited (ordered) set of points to a single new (previously unvisited) point in X
  - $(x_1,y_1),\ldots,(x_m,y_m) \rightarrow x_{m+1} \mid x_{m+1}$ not in $\{x_1,\ldots,x_m\}$

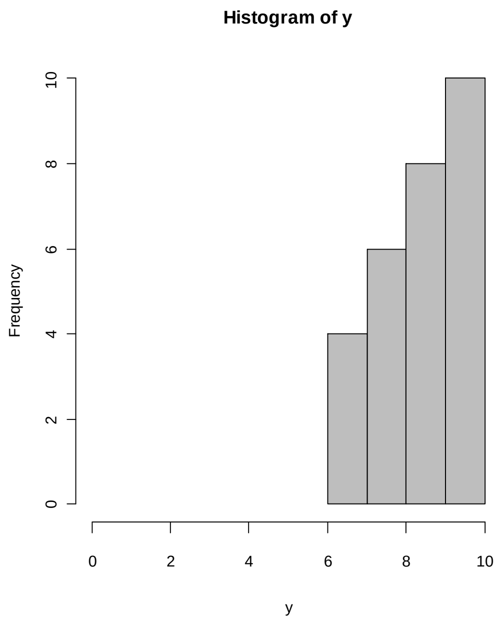
Preliminaries

- Performance of an algorithm
  - Based on *y*-values (cost values) produced from a certain number of searched points
    - *y*-values from *m* iterations of the algorithm
      - d_m^y
  - Performance measure: $\Phi(d_m^y)$
  - Note: Revisits are not counted

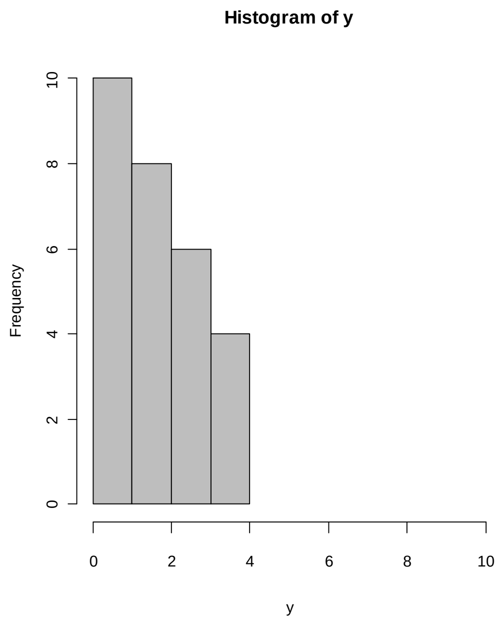
Preliminaries

- Possible performance measures
  - Largest (or smallest) cost value (y-value) in the sample
  - Some function of the histogram of cost values
    - Histogram of cost values: $\vec{c} = (c_{\mathcal{Y}_1}, c_{\mathcal{Y}_2}, \ldots, c_{\mathcal{Y}_{|\mathcal{Y}|}})$
    - $c_{\mathcal{Y}_i}$ = number of times the cost value $\mathcal{Y}_i$ occurs in sample
  - Apply some function that maps the histogram to a "goodness" measure or ranking
    - One possibility $\Phi(\vec{c}) : \vec{c} \mapsto \mathbb{R}$
    - Larger values indicate a better ranking

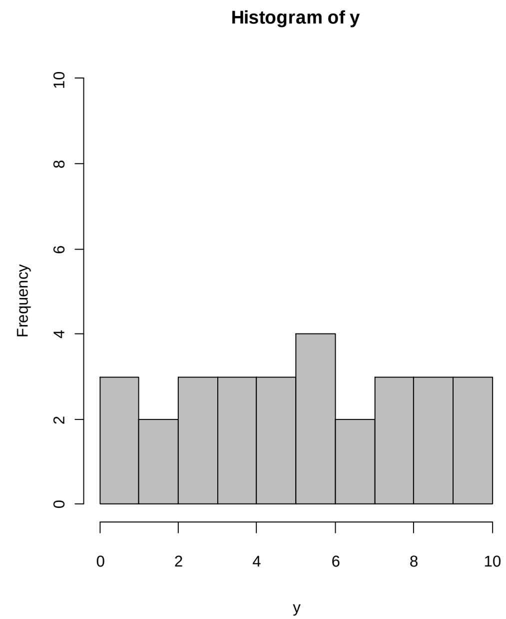
Histogram Examples



10



1



5

Extending the Choosing Procedure NFL Theorem

- Result 1
  - Prove NFL Theorem that is an extension of the Choosing Procedure NFL Theorem

Extending the Choosing Procedure NFL Theorem

Original	Extension
Single run of algorithms	Multiple algorithm runs

Original

- Single run of algorithms
- Performance
  - Continuation of single algorithm run

Extension

- Multiple algorithm runs
  - Training set
  - Choose starting values uniformly at random
- Performance
  - New algorithm run, starting from a new initial *x*-value
  - Test run

Extending the Choosing Procedure NFL Theorem

- New Choosing Procedure Theorem
  - Run a and a' N times on some function f (training runs)
    - Common starting value for each run is chosen uniformly at random
    - Call these values $x_1,\ldots,x_N$
  - CP examines the samples d_1, d_2, …, d_N and d'_1, d'_2, …, d'_N (each of size m) which result from these runs

Samples

From a

- d_1: $\{(x^{(1)},y^{(1)}),\ldots,(x^{(m)},y^{(m)})\}$
- d_2: $\{(x^{(1)},y^{(1)}),\ldots,(x^{(m)},y^{(m)})\}$
- .
- .
- .
- d_N: $\{(x^{(1)},y^{(1)}),\ldots,(x^{(m)},y^{(m)})\}$

From a'

- d'_1: $\{(x^{(1)},y^{(1)}),\ldots,(x^{(m)},y^{(m)})\}$
- d'_2 : $\{(x^{(1)},y^{(1)}),\ldots,(x^{(m)},y^{(m)})\}$
- .
- .
- .
- d'_N : $\{(x^{(1)},y^{(1)}),\ldots,(x^{(m)},y^{(m)})\}$

Extending the Choosing Procedure NFL Theorem

- New Choosing Procedure Theorem
  - CP decides which algorithm, *a* or *a'*, to use on the (N+1)th algorithm run (test run) on *f*
    - Starting value chosen uniformly at random
    - Must be new starting value
      - x_{N+1} not in $\{x_1,\ldots,x_N\}$

Result 1: New NFL Theorem

- $$\sum_{a,a'} P(\vec{c}_{>m\cdot N} | f, x_{N+1}, d_1, d_2, \ldots, d_N, d'_1, d'_2, \ldots, d'_N, m, a, a', A)$$

$$= \sum_{a,a'} P(\vec{c}_{>m\cdot N} | f, x_{N+1}, d_1, d_2, \ldots, d_N, d'_1, d'_2, \ldots, d'_N, m, a, a', B)$$

- Fixed samples, arbitrary new starting point, arbitrary fixed function, A and B are any two CP's
- Sum over all algorithm pairs consistent with samples
  - Probability of obtaining a particular histogram is independent of CP
- Performance (function of histogram) is independent of CP
- On average, performance of any two CP's is equivalent

Sketch of Proof

- Concerned with $P(\vec{c}_{>m \cdot N} | \ldots)$
  - Probability of a particular histogram of cost values on the *(N+1)*th run (test run)
- Starting value on test run, x_{N+1} , not in $\{x_1,\ldots,x_N\}$
  - What algorithms do on test run is independent of the training runs
  - Both algorithms are free to visit any possible sequence of *m* values beginning with x_{N+1}

Sketch of Proof

- Both summations sum over the same set of possibilities for $\vec{c}_{>m \cdot N}$
  - Can be viewed as a change of variables
  - Sum of probabilities is independent of the particular choosing procedure
    - Sum of probabilities for choosing procedure A equals sum of probabilities for choosing procedure B
    - $$\sum_{a,a'} P(\vec{c}_{>m \cdot N} | f, x_{N+1}, d_1, d_2, \ldots, d_N, d'_1, d'_2, \ldots, d'_N, m, a, a', A)$$

$$= \sum_{a,a'} P(\vec{c}_{>m \cdot N} | f, x_{N+1}, d_1, d_2, \ldots, d_N, d'_1, d'_2, \ldots, d'_N, m, a, a', B)$$

Corollary

$$E_{a,a',x_{N+1}}\left[\Phi(\vec{c}_{>m\cdot N})|f, x_{N+1}, d_1, d_2, \ldots, d_N, d'_1, d'_2, \ldots, d'_N, m, a, a', A\right]$$
$$= E_{a,a',x_{N+1}}\left[\Phi(\vec{c}_{>m\cdot N})|f, x_{N+1}, d_1, d_2, \ldots, d_N, d'_1, d'_2, \ldots, d'_N, m, a, a', B\right]$$

- For any fixed training data, the expected performance—over choice of starting point and algorithms—of any two choosing procedures is equivalent

What Is Inferred from Theorem

- Wolpert and Macready
  - Barring assumptions about the optimization algorithms and/or f
    - No theoretical justification for using any particular choosing procedure
    - On average, no choosing procedure is any better than a random choosing procedure
- We will show that this is not necessarily the case

A Superior Choosing Procedure

- Result 2
  - Show that despite this theorem, there exists (at least) one choosing procedure that, on average, is better than random

A Superior Choosing Procedure

- This choosing procedure makes its choice as follows
  - If one algorithm outperforms the other on <u>all</u> algorithm runs in the training set
    - Choose this algorithm
  - Otherwise
    - Randomly choose between the algorithms
    - Each algorithm is chosen with probability ½
- Call this the *unanimous choosing procedure* (UCP)
  - Only makes choice when unanimous support for one of the algorithms

Why the Procedure Is Superior

- If one algorithm consistently beats the other for all *N* runs in the training sets
  - Using standard probability theory
    - Probability that UCP "fooled" into thinking this algorithm is better becomes exponentially small as N grows
- To get fooled
  - One algorithm <u>wins</u> on all runs in the training set
  - More often than not this algorithm will <u>lose</u> on a test run

Why the Procedure Is Superior

- If choose a large enough (yet reasonable) value for the number of training runs *N*
  - Probability that the UCP is fooled in such a way is extremely small, perhaps around 2^{-128}
  - Rational to believe or safe to assume that UCP won't be fooled
- If not fooled into making bad decisions
  - Good performance not necessarily offset by bad performance
  - Average performance is better than random

Cryptographic Practice and Rationality

- Basis of using 2^{-128} as an appropriately small probability
- National Security Agency (NSA) uses encryption algorithm AES-128
  - Encrypt classified documents
  - Uses 128-bit keys
    - Relies on probability of 2^{-128} that random guess will be able to decrypt document

Cryptographic Practice and Rationality

- How small is 2^{-128}?
  - Even if
    - Same key used to encrypt every classified document
    - A billion documents encrypted per second for a billion years
    - Systematically guess and check distinct keys
  - Probability of any guesses succeeding is less than 1 in 10 trillion [JacTam]
- Rational to believe or safe to assume
  - Real-world events with extremely small probability of occurring will not occur, even though mathematically we cannot rule out their possibility [JacTam]

A Sufficient Training Set

- How many training runs is sufficient?
  - Enough so that the prediction error of the UCP is less than ½
- Prediction error
  - Probability that the chosen algorithm will perform worse on a test run
- Why prediction error less than ½?
  - When a random choosing procedure selects an algorithm
    - With probability ½ this choice is correct
      - Chosen algorithm will perform better on a test run
    - With probability ½ this choice is incorrect
    - The *prediction error* is ½

Prediction Error of UCP

- Unanimous choosing procedure
  - One algorithm does <u>not</u> consistently beat the other
    - Randomly selects an algorithm
    - Prediction error is ½
  - One algorithm <u>does</u> consistently beat the other
    - If *N* is large enough
      - With extremely high probability, prediction error is less than ½
        - $1-2^{-128}$
  - Averaged over unseen starting values, prediction error is less than ½
    - Better than random

A Sufficient Training Set

- Using probability theory
  - Can show that it's overwhelming likely that a certain classification error holds
  - Classification error
    - Probability over all possible starting values that the chosen algorithm performs worse
      - Prediction error – probability over unseen starting values
  - Use classification error to calculate prediction error

A Sufficient Training Set

- Can show that it is extremely likely that a particular classification error holds
  - Fix this value to 0.24
  - Even if prediction error is double the classification error
    - Prediction error is $0.48 < ½$
    - If number of training runs is less than $½|X|$ then prediction error is at most double (because uniform choice of x)
- Need to calculate *N* such that with extremely high probability
  - Classification error is no more than 0.24
  - Prediction error is no more than 0.48

A Sufficient Training Set

- If classification error is at least 0.24
  - On one training run
    - Probability over randomized choice of starting points that the UCP does not pick losing algorithm is at most
      - $1 - 0.24 = 0.76$
  - On N training runs
    - Probability over randomized choice of starting points that the UCP fails to detect any losses is at most
      - $(1 - 0.24)^N = (0.76)^N$

A Sufficient Training Set

- On the test run of the algorithms
  - Probability that the UCP is “fooled” by the randomized choice of starting values in the training set is at most
    - $(0.76)^N$
  - Probability $(0.76)^N$ that fooled into choosing the “worse” algorithm
    - Because no losses were detected during training runs

A Sufficient Training Set

- To calculate sufficient training set
  - Set probability of being fooled, $(0.76)^N$, less than some extraordinarily small value $\delta > 0$
  - Solve for N
- We will set the extraordinarily small value δ to $2^{-\sigma}$
  - Let $\sigma = 128$
  - This choice of σ is from standard cryptographic practice

A Sufficient Training Set

- In order to find a sufficient training set size N such that $(0.76)^N < \delta$
  - Use the following formula from [Alguin88]

$$N \geq \left\lceil \frac{1}{\epsilon_c} ln(\frac{1}{\delta}) \right\rceil$$

  - ϵ_c is the classification error
  - Note that $(0.76)^N$ is just $(1-\epsilon_c)^N$, so $\epsilon_c = 0.24$
- For $\epsilon_c = 0.24$ and $\delta = 2^{-128}$, we have

$$\left\lceil \frac{1}{\epsilon_c} ln(\frac{1}{\delta}) \right\rceil = \lceil \frac{1}{0.24} ln(\frac{1}{2^{-128}}) \rceil = 370$$

A Sufficient Training Set

- When the UCP makes a choice (doesn't randomly choose)
  - Values of *N* greater than or equal to 370 are sufficient to
    - Produce an algorithm choice that with probability $(1\text{-}2^{-128})$ has
      - Classification error at most 0.24
      - Prediction error at most 0.48

Why UCP is Superior to Random

- UCP either
  - Randomly chooses
    - Prediction error of ½
  - Makes a choice
    - Overwhelmingly likely/rational to believe/safe to assume that prediction error is less than ½
- On average, prediction error is less than ½
  - Better than random

Comparison

- NFL theorem
  - Seems to imply expected prediction error is exactly ½ for all choosing procedures
- We show
  - If believe claim regarding extremely small probabilities
    - Perform enough training runs
    - Rational to believe or safe to assume the expected prediction error of the UCP is less than ½
    - <u>Implications</u> of the NFL theorem are <u>not as negative as expected</u>

Comparison to the St. Petersburg Paradox

- Similar paradox between mathematical probabilities and rational beliefs
- St. Petersburg Paradox
  - Gambling game
  - Flip fair coin until get “tails”
  - If “tails” comes up on
    - 1st flip → payout of $2
    - 2nd flip → payout of $4
    - k^{th} flip → payout of $$2^k$

Comparison to the St. Petersburg Paradox

- Expected payout of game is arbitrarily large

$$\begin{aligned}
\text{Expected payout} &= \sum_{k=1}^{\infty} (\text{Payout on 1st "tails" on kth flip}) \cdot Pr\left[\text{1st "tails" on kth flip}\right] \\
&= \sum_{k=1}^{\infty} 2^k \cdot 2^{-k} \\
&= \sum_{k=1}^{\infty} 1 \\
&= \infty
\end{aligned}$$

Comparison to St. Petersburg Paradox

- How much should someone be willing to pay to play this game?
  - Most rational people would not even pay $25 [Hacking80]

Comparison to St. Petersburg Paradox

- Paradox
  - Mathematically
    - Should be willing to pay arbitrarily large amount
  - Most rational people not willing to do this
    - Mathematics doesn't always provide a good model of rational real-world behavior
- One reason paradox occurs
  - Extremely low probability events used to calculate expected payout
    - Events such as
      - Flipping a coin 128 times before a "tails" comes up

Conclusion

- Mathematically
  - Show an NFL result
    - On average, the performance of any two choosing procedures is mathematically equivalent
- Using probability theory and cryptography concepts
  - If rational to believe/safe to assume extraordinarily small probability events won't occur
    - There exists (at least) one CP—the UCP—that, on average, is better than random
- Although in strict mathematical sense NFL theorem holds
  - Implications are not as negative as expected

Future Work

- Allow ties
  - Investigate appropriate cut-off for an allowable percentage of ties
- Analysis of not requiring one algorithm to always win
  - Better if one algorithm wins on 75% of training runs? 51%?
- Combine with analysis of NFL theorems for learning

References

[1] E. H. L. Aarts and J. K. Lenstra. *Local Search in Combinatorial Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley and Sons, Ltd., Chichester, England (UK), 1997.

[2] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[3] D. L. Smitley and I. Lee. Comparative analysis of hill climbing mapping algorithms. Technical report, University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-88-94, November 1988.

[4] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[5] J. C. Jackson and C. Tamon. On the rationality of belief in free lunches in learning. Unpublished manuscript-in-preparation.

References

[6] National Institute of Standards and Technology. *FIPS PUB 197: Advanced Encryption Standard (AES)*. National Institute for Standards and Technology, Gaithersburg, MD, November 2001.

[7] Committee on National Security Systems. Fact sheet no. 1 for the national policy on the use of the advanced encryption standard (aes) to protect national security systems and national security information. Technical report, June 2003.

[8] D. Bernoulli. Exposition of a new theory on the measurement of risk. *Econometrica*, 22(1):23–36, 1954.

[9] I. Hacking. Strange expectations. *Philosophy of Science*, 47(4):562–567, 1980.

[10] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.