

No free lunch in search and optimization

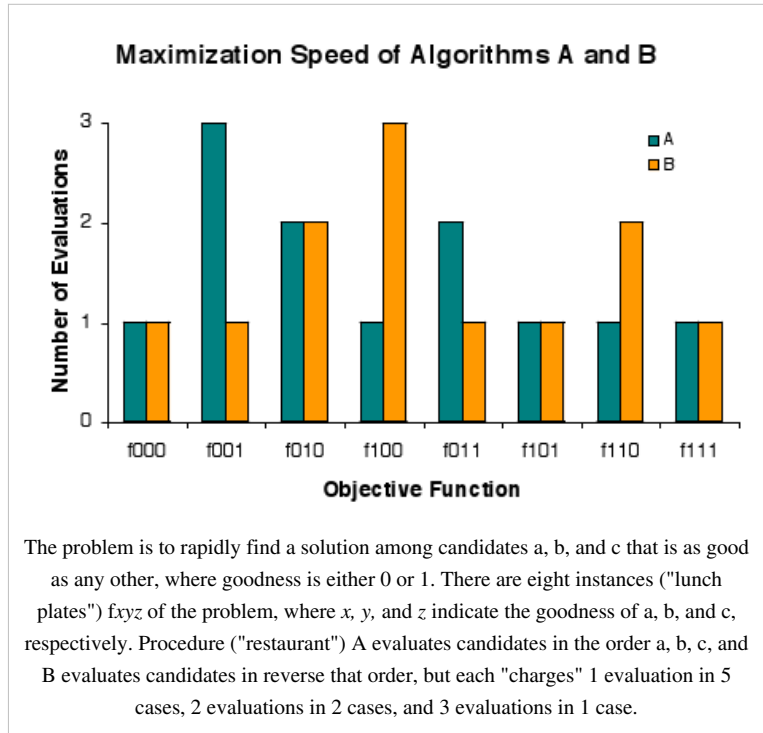
In computing, there are circumstances in which the outputs of all procedures solving a particular type of problem are statistically identical. A colorful way of describing such a circumstance, introduced by David Wolpert and William G. Macready in connection with the problems of search^[1] and optimization,^[2] is to say that **there is no free lunch**. Wolpert had previously derived no free lunch theorems for machine learning (statistical inference).^[3] Before Wolpert's article was published, Cullen Schaffer had summarized a preprint version of this work of Wolpert's, but used different terminology.^[4]

In the "no free lunch" metaphor, each "restaurant" (problem-solving procedure) has a "menu" associating each "lunch plate" (problem) with a "price" (the performance of the procedure in solving the problem). The menus of restaurants are identical except in one regard — the prices are shuffled from one restaurant to the next. For an omnivore who is as likely to order each plate as any other, the average cost of lunch does not depend on the choice of restaurant. But a vegan who goes to lunch regularly with a carnivore who seeks economy pays a high average cost for lunch. To methodically reduce the average cost, one must use advance knowledge of a) what one will order and b) what the order will cost at various restaurants. That is, improvement of performance in problem-solving hinges on using prior information to match procedures to problems.^{[2][4]}

In formal terms, there is no free lunch when the probability distribution on problem instances is such that all problem solvers have identically distributed results. In the case of search, a problem instance is an objective function, and a result is a sequence of values obtained in evaluation of candidate solutions in the domain of the function. For typical interpretations of results, search is an optimization process. There is no free lunch in search if and only if the distribution on objective functions is invariant under permutation of the space of candidate solutions.^{[5][6][7]} This condition does not hold precisely in practice,^[6] but an "(almost) no free lunch" theorem suggests that it holds approximately.^[8]

Overview

Some computational problems are solved by searching for good solutions in a space of candidate solutions. A description of how to repeatedly select candidate solutions for evaluation is called a search algorithm. On a particular problem, different search algorithms may obtain different results, but over all problems, they are indistinguishable. It follows that if an algorithm achieves superior results on some problems, it must pay with inferiority on other problems. In this sense there is no free lunch in search.^[1] Alternatively, following Schaffer,^[4] search performance is conserved. Usually search is interpreted as optimization, and this leads to the observation that there is no free lunch in optimization.^[2]



"The 'no free lunch' theorem of Wolpert and Macready," as stated in plain language by Wolpert and Macready themselves, is that "any two algorithms are equivalent when their performance is averaged across all possible problems."^[9] The "no free lunch" results indicate that matching algorithms to problems gives higher average performance than does applying a fixed algorithm to all. Igel and Toussaint^[6] and English^[7] have established a general condition under which there is no free lunch. While it is physically possible, it does not hold precisely.^[6] Droste, Jansen, and Wegener have proved a theorem they interpret as indicating that there is "(almost) no free lunch" in practice.^[8]

To make matters more concrete, consider an optimization practitioner confronted with a problem. Given some knowledge of how the problem arose, the practitioner may be able to exploit the knowledge in selection of an algorithm that will perform well in solving the problem. If the practitioner does not understand how to exploit the knowledge, or simply has no knowledge, then he or she faces the question of whether some algorithm generally outperforms others on real-world problems. The authors of the "(almost) no free lunch" theorem say that the answer is essentially no, but admit some reservations as to whether the theorem addresses practice.^[8]

No free lunch (NFL)

A "problem" is, more formally, an objective function that associates candidate solutions with goodness values. A search algorithm takes an objective function as input and evaluates candidate solutions one-by-one. The output of the algorithm is the sequence of observed goodness values.^{[10][11]}

Wolpert and Macready stipulate that an algorithm never reevaluates a candidate solution, and that algorithm performance is measured on outputs.^[2] For simplicity, we disallow randomness in algorithms. Under these conditions, when a search algorithm is run on every possible input, it generates each possible output exactly once.^[7] Because performance is measured on the outputs, the algorithms are indistinguishable in how often they achieve particular levels of performance.

Some measures of performance indicate how well search algorithms do at optimization of the objective function. Indeed, there seems to be no interesting application of search algorithms in the class under consideration but to optimization problems. A common performance measure is the least index of the least value in the output sequence. This is the number of evaluations required to minimize the objective function. For some algorithms, the time required to find the minimum is proportional to the number of evaluations.^[7]

The original no free lunch (NFL) theorems assume that all objective functions are equally likely to be input to search algorithms.^[2] It has since been established that there is NFL if and only if, loosely speaking, "shuffling" objective functions has no impact on their probabilities.^{[6][7]} Although this condition for NFL is physically possible, it has been argued that it certainly does not hold precisely.^[6]

The obvious interpretation of "not NFL" is "free lunch," but this is misleading. NFL is a matter of degree, not an all-or-nothing proposition. If the condition for NFL holds approximately, then all algorithms yield approximately the same results over all objective functions.^[7] Note also that "not NFL" implies only that algorithms are inequivalent overall by *some* measure of performance. For a performance measure of interest, algorithms may remain equivalent, or nearly so.^[7]

In theory, all algorithms perform well in optimization almost always. That is, an algorithm obtains good solutions with relatively few evaluations for almost all objective functions.^[11] The reason is that almost all objective functions exhibit a high degree of Kolmogorov randomness. This equates to extreme irregularity and unpredictability. All levels of goodness are equally represented among candidate solutions, and good solutions are scattered all about the space of candidates. A search algorithm will rarely evaluate more than a small fraction of the candidates before locating a very good solution.^[11]

In practice, almost all objective functions and algorithms are of such high Kolmogorov complexity that they cannot arise.^{[5][7][11]} There is more information in the typical objective function or algorithm than Seth Lloyd estimates the

observable universe is capable of registering.^[12] For instance, if each candidate solution is encoded as a sequence of 300 0's and 1's, and the goodness values are 0 and 1, then most objective functions have Kolmogorov complexity of at least 2^{300} bits,^[13] and this is greater than Lloyd's bound of $10^{90} \approx 2^{299}$ bits. It follows that not all of "no free lunch" theory applies to physical reality. In a practical sense, algorithms "small enough" for application in physical reality are superior in performance to those that are not.

Formal synopsis of NFL

\mathbf{Y}^X is the set of all objective functions $f: X \rightarrow Y$, where X is a finite solution space and Y is a finite poset. The set of all permutations of X is J . A random variable F is distributed on \mathbf{Y}^X . For all j in J , $F \circ j$ is a random variable distributed on \mathbf{Y}^X , with $P(F \circ j = f) = P(F = f \circ j^{-1})$ for all f in \mathbf{Y}^X .

Let $a(f)$ denote the output of search algorithm a on input f . If $a(F)$ and $b(F)$ are identically distributed for all search algorithms a and b , then F has an *NFL distribution*. This condition holds if and only if F and $F \circ j$ are identically distributed for all j in J .^{[6][7]} In other words, there is no free lunch for search algorithms if and only if the distribution of objective functions is invariant under permutation of the solution space.

The "only if" part was first published by C. Schumacher in his PhD dissertation "Black Box Search - Framework and Methods" (The University of Tennessee, Knoxville (2000)). Set-theoretic NFL theorems have recently been generalized to arbitrary cardinality X and Y ("Reinterpreting No Free Lunch", accepted, Evolutionary Computation Journal).

Original NFL theorems

Wolpert and Macready give two principal NFL theorems, the first regarding objective functions that do not change while search is in progress, and the second regarding objective functions that may change.^[2]

Theorem 1: For any pair of algorithms a_1 and a_2

$$\sum_f P(h_m^y | f, m, a_1) = \sum_f P(h_m^y | f, m, a_2).$$

In essence, this says that when all functions f are equally likely, the probability of observing an arbitrary sequence of m values in the course of search does not depend upon the search algorithm. Theorem 2 establishes a "more subtle" NFL result for time-varying objective functions.

Interpretations of NFL results

A conventional, but not entirely accurate, interpretation of the NFL results is that "a general-purpose universal optimization strategy is theoretically impossible, and the only way one strategy can outperform another is if it is specialized to the specific problem under consideration".^[14] Several comments are in order:

A general-purpose almost-universal optimizer exists theoretically. Each search algorithm performs well on almost all objective functions.^[11]

An algorithm may outperform another on a problem when neither is specialized to the problem. It may be that both algorithms are among the worst for the problem. Wolpert and Macready have developed a measure of the degree of "match" between an algorithm and a problem.^[2] To say that one algorithm matches a problem better than another is not to say that either is specialized to the problem.

In practice, some algorithms reevaluate candidate solutions. The superiority of an algorithm that never reevaluates candidates over another that does on a particular problem may have nothing to do with specialization to the problem.

For almost all objective functions, specialization is essentially accidental. Incompressible, or Kolmogorov random, objective functions have no regularity for an algorithm to exploit. Given an incompressible objective

function, there is no basis for choosing one algorithm over another. If a chosen algorithm performs better than most, the result is happenstance.^[11]

In practice, only highly compressible (far from random) objective functions fit in the storage of computers, and it is not the case that each algorithm performs well on almost all compressible functions. There is generally a performance advantage in incorporating prior knowledge of the problem into the algorithm. While the NFL results constitute, in a strict sense, full employment theorems for optimization professionals, it is important not to take the term literally. For one thing, humans often have little prior knowledge to work with. For another, incorporating prior knowledge does not give much of a performance gain on some problems. Finally, human time is very expensive relative to computer time. There are many cases in which a company would choose to optimize a function slowly with an unmodified computer program rather than rapidly with a human-modified program.

The NFL results do not indicate that it is futile to take "pot shots" at problems with unspecialized algorithms. No one has determined the fraction of practical problems for which an algorithm yields good results rapidly. And there is a practical free lunch, not at all in conflict with theory. Running an implementation of an algorithm on a computer costs very little relative to the cost of human time and the benefit of a good solution. If an algorithm succeeds in finding a satisfactory solution in an acceptable amount of time, a small investment has yielded a big payoff. If the algorithm fails, then little is lost.

Coevolutionary free lunches

Wolpert and Macready have proved that there are free lunches in coevolutionary optimization.^[9] Their analysis "covers 'self-play' problems. In these problems, the set of players work together to produce a champion, who then engages one or more antagonists in a subsequent multiplayer game."^[9] That is, the objective is to obtain a good player, but without an objective function. The goodness of each player (candidate solution) is assessed by observing how well it plays against others. An algorithm attempts to use players and their quality of play to obtain better players. The player deemed best of all by the algorithm is the champion. Wolpert and Macready have demonstrated that some coevolutionary algorithms are generally superior to other algorithms in quality of champions obtained. Generating a champion through self-play is of interest in evolutionary computation and game theory. The results are inapplicable to coevolution of biological species, which does not yield champions.^[9]

References and notes

- [1] Wolpert, D.H., Macready, W.G. (1995), No Free Lunch Theorems for Search, Technical Report SFI-TR-95-02-010 (Santa Fe Institute).
- [2] Wolpert, D.H., Macready, W.G. (1997), "No Free Lunch Theorems for Optimization," *IEEE Transactions on Evolutionary Computation* **1**, 67. <http://ti.arc.nasa.gov/m/profile/dhw/papers/78.pdf>
- [3] Wolpert, David (1996), "The Lack of A Priori Distinctions between Learning Algorithms," *Neural Computation*, pp. 1341-1390.
- [4] Schaffer, Cullen (1994), "A conservation law for generalization performance," *International Conference on Machine Learning*, H. Willian and W. Cohen, Editors. San Francisco: Morgan Kaufmann, pp.295-265.
- [5] Streeter, M. (2003) "Two Broad Classes of Functions for Which a No Free Lunch Result Does Not Hold," *Genetic and Evolutionary Computation—GECCO 2003*, pp. 1418–1430.
- [6] Igel, C., and Toussaint, M. (2004) "A No-Free-Lunch Theorem for Non-Uniform Distributions of Target Functions," *Journal of Mathematical Modelling and Algorithms* **3**, pp. 313–322.
- [7] English, T. (2004) No More Lunch: Analysis of Sequential Search, *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pp. 227-234. <http://BoundedTheoretics.com/CEC04.pdf>
- [8] S. Droste, T. Jansen, and I. Wegener. 2002. "Optimization with randomized search heuristics: the (A)NFL theorem, realistic scenarios, and difficult functions," *Theoretical Computer Science*, vol. 287, no. 1, pp. 131-144.
- [9] Wolpert, D.H., and Macready, W.G. (2005) "Coevolutionary free lunches," *IEEE Transactions on Evolutionary Computation*, 9(6): 721-735
- [10] A search algorithm also outputs the sequence of candidate solutions evaluated, but that output is unused in this article.
- [11] English, T. M. 2000. "Optimization Is Easy and Learning Is Hard in the Typical Function," *Proceedings of the 2000 Congress on Evolutionary Computation: CEC00*, pp. 924-931. <http://www.BoundedTheoretics.com/cec2000.pdf>
- [12] Lloyd, S. (2002) "Computational capacity of the universe," *Physical Review Letters* **88**, pp. 237901-237904. <http://arxiv.org/abs/quant-ph/0110141>
- [13] Li, M., and Vitányi, P. (1997) *An Introduction to Kolmogorov Complexity and Its Applications* (2nd ed.), New York: Springer.

- [14] Ho, Y.C., Pepyne, D.L. (2002), "Simple Explanation of the No-Free-Lunch Theorem and Its Implications," *Journal of Optimization Theory and Applications* **115**, 549.

External links

- <http://www.no-free-lunch.org>
 - Yin-Yang: No-Free-Lunch Theorems for Search (<http://www.cs.uwo.edu/~wspears/yin-yang.html>)
 - Radcliffe and Surry, 1995, "Fundamental Limitations on Search Algorithms: Evolutionary Computing in Perspective" (the first published paper on NFL, available in various formats) (<http://citeseer.ist.psu.edu/radcliffe95fundamental.html>)
 - NFL publications by Thomas English (<http://BoundedTheoretics.com/Papers.html>)
 - NFL publications by Christian Igel and Marc Toussaint (<http://www.neuroinformatik.ruhr-uni-bochum.de/PEOPLE/igel/publications.html>)
 - NFL and "free lunch" publications by Darrell Whitley (<http://www.cs.colostate.edu/~genitor/Pubs.html>)
 - Publications by David Wolpert, William Macready, and Mario Koeppen on optimization and search (<http://ti.arc.nasa.gov/profile/dhw/optimization/>)
-

Article Sources and Contributors

No free lunch in search and optimization *Source:* <http://en.wikipedia.org/w/index.php?oldid=469102838> *Contributors:* 1ForTheMoney, ABCD, Altenmann, Artur adib, Baegis, Beetstra, BeteNoir, Charles Matthews, Chochopk, Delldot, Dgrant, Disavian, Duncharris, Epbr123, Epktsang, FeloniousMonk, Geometry guy, Gifflite, GregorB, Hairy Dude, Headbomb, Hongooi, It Is Me Here, Japanese Searobin, Joerg Kurt Wegner, Josedavid, Kiefer.Wolfowitz, Kieff, KissL, Lockley, Magister Mathematicae, MagneticFlux, Malycтенar, Marudubshinki, Memming, Michael Hardy, Natevw, Nikai, Novamo, Odd nature, Oleg Alexandrov, Park3r, Pdturney, Prolog, Pudor, Rls, Skatche, Skomorokh, Stoni, The Anome, ThomHImself, Trivial, Vecter, Whoutz, XDaniелx, 75 anonymous edits

Image Sources, Licenses and Contributors

Image:**No free lunch theorems figure.png** *Source:* http://en.wikipedia.org/w/index.php?title=File:No_free_lunch_theorems_figure.png *License:* Public Domain *Contributors:* ThomHImself (talk) 19:25, 19 February 2008 (UTC)

License

Creative Commons Attribution-Share Alike 3.0 Unported
[//creativecommons.org/licenses/by-sa/3.0/](http://creativecommons.org/licenses/by-sa/3.0/)