WIKIPEDIA

No free lunch theorem

In mathematical folklore, the "**no free lunch**" (**NFL**) **theorem** (sometimes pluralized) of David Wolpert and William Macready appears in the 1997 "No Free Lunch Theorems for Optimization".[1] Wolpert had previously derived no free lunch theorems for machine learning (statistical inference).[2]

In 2005, Wolpert and Macready themselves indicated that the first theorem in their paper "state[s] that any two optimization algorithms are equivalent when their performance is averaged across all possible problems".[3]

The "no free lunch" (NFL) theorem is an easily stated and easily understood consequence of theorems Wolpert and Macready actually prove. It is weaker than the proven theorems, and thus does not encapsulate them. Various investigators have extended the work of Wolpert and Macready substantively. See No free lunch in search and optimization for treatment of the research area.

While some scholars argue that NFL conveys important insight, others argue that NFL is of little relevance to machine learning research.[4][5]

Contents

Example

Posit a toy universe that exists for exactly two days and on each day contains exactly one object, a square or a triangle. The universe has exactly four possible histories:

1. (square, triangle): the universe contains a square on day 1, and a triangle on day 2
2. (square, square)
3. (triangle, triangle)
4. (triangle, square)

Any prediction strategy that succeeds for history #2, by predicting a square on day 2 if there is a square on day 1, will fail on history #1, and vice versa. If all histories are equally likely, then any prediction strategy will score the same, with the same accuracy rate of 0.5.[6]

Original NFL theorems

Wolpert and Macready give two NFL theorems that are closely related to the folkloric theorem. In their paper, they state:

> We have dubbed the associated results NFL theorems because they demonstrate that if an algorithm performs well on a certain class of problems then it necessarily pays for that with degraded performance on the set of all remaining problems.[1]

The first theorem hypothesizes objective functions that do not change while optimization is in progress, and the second hypothesizes objective functions that may change.[1]

> *Theorem 1*: For any algorithms a_1 and a_2, at iteration step m
> $$\sum_f P(d_m^y \mid f, m, a_1) = \sum_f P(d_m^y \mid f, m, a_2),$$

where d_m^y denotes the ordered set of size m of the cost values y associated to input values $x \in X$, $f : X \to Y$ is the function being optimized and $P(d_m^y \mid f, m, a)$ is the conditional probability of obtaining a given sequence of cost values from algorithm a run m times on function f.

The theorem can be equivalently formulated as follows:

> *Theorem 1*: Given a finite set V and a finite set S of real numbers, assume that $f : V \to S$ is chosen at random according to uniform distribution on the set S^V of all possible functions from V to S. For the problem of optimizing f over the set V, then no algorithm performs better than blind search.

Here, *blind search* means that at each step of the algorithm, the element $v \in V$ is chosen at random with uniform probability distribution from the elements of V that have not been chosen previously.

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 optimization does not depend upon the algorithm. In the analytic framework of Wolpert and Macready, performance is a function of the sequence of observed values (and not e.g. of wall-clock time), so it follows easily that all algorithms have identically distributed performance when objective functions are drawn uniformly at random, and also that all algorithms have identical mean performance. But identical mean performance of all algorithms does not imply Theorem 1, and thus the folkloric theorem is not equivalent to the original theorem.

Theorem 2 establishes a similar, but "more subtle", NFL result for time-varying objective functions.[1]

Motivation

The NFL theorems were explicitly *not* motivated by the question of what can be inferred (in the case of NFL for machine learning) or found (in the case of NFL for search) when the "environment is uniform random". Rather uniform randomness was used as a tool, to compare the number of environments for which algorithm A outperforms algorithm B to the number of environments for which B outperforms A. NFL tells us that (appropriately weighted) there are just as many environments in both of those sets.

This is true for many definitions of what precisely an "environment" is. In particular, there are just as many prior distributions (appropriately weighted) in which learning algorithm A beats B (on average) as vice versa. This statement about *sets of priors* is what is most important about NFL, not the fact that any two algorithms perform equally for the single, specific prior distribution that assigns equal probability to all environments.

While the NFL is important to understand the fundamental limitation for a set of problems, it does not state anything about each particular instance of a problem that can arise in practice. That is, the NFL states what is contained in its mathematical statements and it is nothing more than that. For example, it applies to the situations where the algorithm is fixed a priori and a worst-case problem for the fixed algorithm is chosen a posteriori. Therefore, if we have a "good" problem in practice or if we can choose a "good" learning algorithm for a given particular problem instance, then the NFL does not mention any limitation about this particular problem instance. Though the NFL might seem contradictory to results from other papers suggesting generalization of learning algorithms or search heuristics, it is important to understand the difference between the exact mathematical logic of the NFL and its intuitive interpretation.[7]

Implications for computing and for the scientific method

To illustrate one of the counter-intuitive implications of NFL, suppose we fix two supervised learning algorithms, C and D. We then sample a target function f to produce a set of input-output pairs, *d*. How should we choose whether to train C or D on *d*, in order to make predictions for what output would be associated with a point lying outside of *d?*

It is common in almost of all science and statistics to answer this question – to choose between C and D – by running cross-validation on *d* with those two algorithms. In other words, to decide whether to generalize from *d* with either C or D*,* we see which of them has better out-of-sample performance when tested within *d*.

Note that since C and D are fixed, this use of cross-validation to choose between them is itself an algorithm, i.e., a way of generalizing from an arbitrary dataset. Call this algorithm A. (Arguably, A is a simplified model of the scientific method itself.)

Note as well though that we could also use *anti*-cross-validation to make our choice. In other words, we could choose between C and D based on which has *worse* out-of-sample performance within *d*. Again, since C and D are fixed, this use of anti-cross-validation is itself an algorithm. Call that algorithm B.

NFL tells us (loosely speaking) that B must beat A on just as many target functions (and associated datasets *d*) as A beats B. In this very specific sense, the scientific method will lose to the "anti" scientific method just as readily as it wins.[8]

However, note that NFL only applies if the target function is chosen from a uniform distribution of all possible functions. If this is not the case, and certain target functions are more likely to be chosen than others, then A may perform better than B overall. The contribution of NFL is that it tells us choosing an appropriate algorithm requires making assumptions about the kinds of target functions the algorithm is being used for. With no assumptions, no "meta-algorithm", such as the scientific method, performs better than random choice.

While some scholars argue that NFL conveys important insight, others argue that NFL is of little relevance to machine learning research.[4][5] If Occam's razor is correct, for example if sequences of lower Kolmogorov complexity are more probable than sequences of higher complexity, then (as is

observed in real life) some algorithms, such as cross-validation, perform better on average on practical problems (when compared with random choice or with anti-cross-validation).[9]

See also

- There ain't no such thing as a free lunch

Notes

1. Wolpert, D.H., Macready, W.G. (1997), "No Free Lunch Theorems for Optimization (http://ti.arc.nasa.gov/m/profile/dhw/papers/78.pdf)", *IEEE Transactions on Evolutionary Computation* **1**, 67.
2. Wolpert, David (1996), "The Lack of *A Priori* Distinctions between Learning Algorithms (http://www.zabaras.com/Courses/BayesianComputing/Papers/lack_of_a_priori_distinctions_wolpert.pdf)", *Neural Computation*, pp. 1341–1390. Archived (https://web.archive.org/web/20161220125415/http://www.zabaras.com/Courses/BayesianComputing/Papers/lack_of_a_priori_distinctions_wolpert.pdf) 2016-12-20 at the Wayback Machine
3. Wolpert, D.H., and Macready, W.G. (2005) "Coevolutionary free lunches", *IEEE Transactions on Evolutionary Computation*, 9(6): 721–735
4. Whitley, Darrell, and Jean Paul Watson. "Complexity theory and the no free lunch theorem (https://www.researchgate.net/profile/Darrell_Whitley2/publication/226085645_Complexity_Theory_and_the_No_Free_Lunch_Theorem/links/5632148608ae0530378e94b9.pdf)." In Search Methodologies, pp. 317–339. Springer, Boston, MA, 2005.
5. Giraud-Carrier, Christophe, and Foster Provost. "Toward a justification of meta-learning: Is the no free lunch theorem a show-stopper (https://www.researchgate.net/profile/Christophe_Giraud-Carrier/publication/228671734_Toward_a_justification_of_meta-learning_Is_the_no_free_lunch_theorem_a_show-stopper/links/0fcfd510c5d5b83ec8000000/Toward-a-justification-of-meta-learning-Is-the-no-free-lunch-theorem-a-show-stopper.pdf)." In Proceedings of the ICML-2005 Workshop on Meta-learning, pp. 12–19. 2005.
6. Forster, Malcolm R. (1999). *Minds and Machines*. **9** (4): 543–564. doi:10.1023/A:1008304819398 (https://doi.org/10.1023%2FA%3A1008304819398). Missing or empty `|title=` (help)
7. Kawaguchi, K., Kaelbling, L.P, and Bengio, Y.(2017) "Generalization in deep learning", https://arxiv.org/abs/1710.05468
8. Wolpert, D.H. (2013) "What the no free lunch theorems really mean", Ubiquity, Volume 2013, December 2013, doi:10.1145/2555235.2555237 (https://doi.org/10.1145%2F2555235.2555237)
9. Lattimore, Tor, and Marcus Hutter. "No free lunch versus Occam's razor in supervised learning (https://arxiv.org/abs/1111.3846)." In Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence, pp. 223–235. Springer, Berlin, Heidelberg, 2013.

External links

- No Free Lunch Theorems (http://www.no-free-lunch.org/)
- Graphics illustrating the theorem (https://commons.wikimedia.org/wiki/File:No_free_lunch_theorem.svg)