No Free Lunch Theorem

CS454 AI-Based Software Engineering

Shin Yoo

No Free Lunch Theorem

> *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.*

Wolpert & Macready, *No free lunch theorems for optimization*, IEEE Transactions on Evolutionary Computation, 1997[1].

Let $\mathcal{X}$ be the search space, and $\mathcal{Y}$ be the *finite* space of fitness values (e.g. the space of 32bit or 64it floating point numbers).

- **Fitness function** f is of type $f : \mathcal{X} \rightarrow \mathcal{Y}$.
- **Space of all problems**: $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$, the finite size of which is $|\mathcal{Y}|^{|\mathcal{X}|}$ (i.e., each solution has the choice of $|\mathcal{Y}|$ fitness values).

A search can be represented as a time ordered sample of m visited points in the search space. We denote such samples as $d_m \equiv \{(d_m^x(1), d_m^y(1)), \ldots, (d_m^x(m), d_m^y(m))\}$. Think of this as a search trajectory. Here, $d_m^x(i)$ indicates the $\mathcal{X}$ value of the ith successive element in the sample of size m, and $d_m^y(i)$ is the corresponding fitness value. The space of all samples of size m is $\mathcal{D}_m = (\mathcal{X} \times \mathcal{Y})^m$.

NFL

We make a probabilistic argument. Let $P(d_m^y|f, m, a)$ be the conditional probability of obtaining a particular fitness value d_m^y when running algorithm a against fitness fucntion f using m samples (i.e. fitness evaluations). Then

Theorem 1 (No Free Lunch Theorem for Optimisations)

For any pair of algorithm a_1 and a_2,

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

where m is the number of fitness evaluation used by a_1 and a_2, f is the fitness(objective function).

In other words, aggregated over *all* fitness functions, algorithm a_1 and a_2 have the same probability to obtain d_m^y.

Proof.

Intuitively, the proof simply shows that $\sum_f P(\vec{c}|f, m, a)$ has no dependence on a. The proof is based on induction on m, of which we only present a sketch here.

- When $m = 1$: the sample is $d_1 = \{(d_1^x, f(d_1^x))\}$. The only possible value for d_1^y is $f(d_1^x)$. As such, for an arbitrary value d^y, the fitness function either returns d^y (i.e., $P(d^y|f, m, a) = 1$), or not (i.e., $P(d^y|f, m, a) = 0$). Consequently,

$$\sum_f P(d_1^y|f, m = 1, a) = \sum_f \delta(d_1^y, f_d(x_1))$$

where δ is the Kronecker delta function (i.e., only returns 1 when two arguments are equal to each other). Again, summing over all possible cost functions f, $\delta(d_1^y, f(x_1))$ is 1 only for those functions which have fitness of d_1^y at point d_1^x. There are $|\mathcal{Y}|^{|\mathcal{X}|-1}$ such functions (i.e., out of $|\mathcal{X}|$ solutions, one has a fixed fitness value, and $|\mathcal{X} - 1|$ solutions have the choice of $|\mathcal{Y}|$ fitness values), therefore:

$$\sum_f P(d_1^y|f, m = 1, a) = |\mathcal{Y}|^{|\mathcal{X}|-1}$$

which is not dependent on a.

Proof.

- For $m + 1$:

$$\sum_f P(d^y_{m+1}|f, m+1, a) = \frac{1}{|\mathcal{Y}|} \sum_f P(d^y_m|f, m, a)$$

Intuitively, each f at m samples have $|\mathcal{Y}|$ choices of fitness values for $m + 1$ sample size, and we are only interested in one of them, d^y_{m+1}, hence the division. Please see Wolpert and Mcready [1] for full detail.

$\square$

No Free Lunch Theorem

Does it mean that we can just use whatever favourite optimisation algorithm for whatever problem?

- No. The proof was ade against all problems, i.e., the entire set of $|\mathcal{Y}|^{|\mathcal{X}|}$ fitness functions. For a specific fitness function, there can be meaningful differences between algorithms.
- Furthermore, additional knowledge into f (i.e., the fitness landscape), will give us competitive edge. We have already seen such a case: elementary landscape.

David H. Wolpert and William G. Macready.
No free lunch theorems for optimization.
IEEE Transactions on Evolutionary Computation, 1(1):67–82, April 1997.