No Free Lunch and Algorithmic Randomness

Simon McGregor
Centre for Research in Cognitive Science (COGS)
University of Sussex
BN1 9RH, UK
sm66@sussex.ac.uk

ABSTRACT

Schumacher, Vose & Whitley [1] have shown that Wolpert & MacReady's celebrated No Free Lunch theorem [2] applies only to classes of target functions which are closed under permutation (c.u.p.). In the same paper, Schumacher et al. demonstrated that there exist both highly compressible and highly incompressible classes of objective functions for which NFL applies. However, I will show that there is a free lunch for the class of all n-compressible target functions $f : \mathcal{X} \to \mathcal{Y}$ given reasonable conditions on n, $|\mathcal{X}|$ and $|\mathcal{Y}|$. While previous authors [3, 4] have considered NFL in the context of some form of complexity restriction on function classes, this paper appears to be the first to contain a proof using the general measure of Kolmogorov complexity.

1. NO FREE LUNCH

When evolutionary algorithms were first introduced, it was hoped that they might provide a general-purpose "black box" search/optimisation tool. However, in [2], Wolpert & MacReady proved that the average performance of all search algorithms, considered over the class of all possible target functions, is the same. Consequently evolutionary computing methods are "no better" than random search when considered over all possible fitness functions. It was subsequently proved in [1] that this "No Free Lunch" (NFL) result extends to classes of function other than the uniform class. NFL holds in the average case, regardless of the algorithm performance measure used, if and only if the class of functions under consideration is closed under permutation (c.u.p.). The lack of structure of c.u.p. classes has information-theoretic implications which have been explored in [5, 6] using Shannon-Weaver information theory; this paper investigates that lack of structure from an algorithmic information theory perspective using Kolmogorov complexity. I conclude that if we consider the class of algorithmically non-random (i.e. compressible) target functions, NFL does not hold.

It should be mentioned that the No Free Lunch theorem is of largely theoretical interest to researchers using artificial evolution. This is due to a number of practical considerations such as local correlation structure in fitness spaces [7] and NFL assumptions ignoring algorithmic complexity [8]. The result presented in this paper is similarly theoretical rather than practical; nevertheless, it extends previously published work.

2. ASSUMPTIONS

As in most previous NFL work, it is assumed that the search domain $\mathcal{X}$ and the set of objective values $\mathcal{Y}$ are sets of finite cardinality. Because the terms will be used frequently, I will use $X = |\mathcal{X}|$ and $Y = |\mathcal{Y}|$ to denote the cardinality of $\mathcal{X}$ and $\mathcal{Y}$ respectively. To avoid the trivial case where NFL necessarily holds, I will assume that $Y \geq 2$. Additionally, it will be assumed that $X \gg 1$ and that $X \geq Y$. This is justified by the fact that in most practical search tasks the set of possible values of a target function is represented as a single fixed-width integer or floating-point number whilst the set of possible solutions is far larger. It is also justified in any case where target values are only used to rank possible solutions relative to one another, because the number of possible ranks is no larger than the number of possible solutions.

3. COMPRESSIBILITY

3.1 Compressibility of Strings

The Kolmogorov complexity $K(S)$ of a binary string S is defined as the length of the smallest program[1] which outputs S and terminates when given no input [9]. A binary string S is defined as n-compressible if $K(S) \leq \text{length}(S) - n$. Incompressible strings (i.e. those for which $K(S) \geq \text{length}(S)$) are also called *algorithmically random* strings, and indeed strings chosen uniformly at random by a stochastic process will typically be incompressible or very nearly so [9].

3.2 Encoding Functions As Strings

Before we can consider compressibility of target functions we need to specify binary string encodings for them. We will begin by encoding each member of $\mathcal{X}$ and $\mathcal{Y}$ as some binary integer in the range $1 \ldots X$ or $1 \ldots Y$ respectively. The encoding of a target function $f : \mathcal{X} \to \mathcal{Y}$ will be a lookup table, i.e. a string listing the encoding of every $f(x), x \in \mathcal{X}$ ordered by x. This string is of length $X \log_2 Y$ bits for every

[1] On some arbitrary but fixed universal reference machine.

f. In a minor abuse of notation I will refer to a function as if it and its binary string encoding were equivalent.

4. THE FREE LUNCH

4.1 Permutation-Classes

Define two functions $f : \mathcal{X} \to \mathcal{Y}$ and $g : \mathcal{X} \to \mathcal{Y}$ as equivalent under permutation of $\mathcal{X}$ if there exists some permutation π of $\mathcal{X}$ such that $g = f \circ \pi$. Call an equivalence class defined by this equivalence relation a permutation-class. The total number of permutation-classes of functions from $\mathcal{X}$ to $\mathcal{Y}$ is Y multichoose X, or $\binom{X+Y-1}{X}$ (see, e.g. [7]). Note that a permutation-class is uniquely characterised by what Igel & Toussaint [7] call a "histogram", i.e. a function $h : \mathcal{Y} \to \mathbb{Z}^*$ indicating how many distinct values of $x \in \mathcal{X}$ are preimages of each $y \in \mathcal{Y}$.

There exists some fixed-length program which, given the size of $\mathcal{X}$ and $\mathcal{Y}$, and a number n in the range $1 \cdots \binom{X+Y-1}{X}$, outputs a function from a different permutation-class for each n. The simplest such program to describe is one which generates a list of all possible "histograms", picks the nth, and outputs the least (e.g. by dictionary ordering) function lookup table which has that histogram. By incorporating the inputs into the program using some fixed-length algorithmic glue, that means that every permutation-class C has at least one member f_C whose Kolmogorov complexity $K(f_C)$ is no more than

$$\log_2 X + \log_2 Y + \log_2 \binom{X+Y-1}{X} + k \tag{1}$$

We will ignore the k term in this and subsequent expressions because it becomes negligible for large enough X. The $\binom{X+Y-1}{X}$ term is certainly no greater than $\binom{X+Y}{X} = \frac{(X+Y)!}{X!Y!}$. Using Stirling's approximation[2] $\ln(n!) \approx n \ln n - n$ we have

$$\log_2 \binom{X+Y}{X} \approx \tfrac{1}{\ln 2}\big((X+Y)\ln(X+Y) - (X+Y) - (X \ln X - X) - (Y \ln Y - Y)\big)$$

$$= (X+Y)\log_2(X+Y) - X\log_2 X - Y \log_2 Y \tag{2}$$

Combining eq. 1 and eq. 2 gives us

$$K(f_C) \leq (X+Y)\log_2(X+Y) - X\log_2 X - Y\log_2 Y + \log_2 X + \log_2 Y \tag{3}$$

4.2 Arbitrary Functions

There are Y^X total possible functions from $\mathcal{X}$ to $\mathcal{Y}$, which by a simple counting argument (see e.g. [9] section 2.2) means that most such functions have a Kolmogorov complexity of at least

$$X \log_2 Y \tag{4}$$

[2]For a rigorous proof we may use Robbins' strict inequality for Stirling's approximation

$$\sqrt{2\pi n}(n^n)e^{-n+\frac{1}{12n+1}} < n! < \sqrt{2\pi n}(n^n)e^{-n+\frac{1}{12n}}$$

Details of this somewhat longer proof are available from the author on request.

4.3 Compressible Functions

Consider the class $C_{K\leq a}$ of all functions whose Kolmogorov complexity is a or less. If a is larger than or equal to eq. 3, but less than eq. 4, this class cannot be closed under permutation. This is because when a is larger than or equal to eq. 3, $C_{K\leq a}$ contains at least one member of every permutation-class, but when a is less than eq. 4, it cannot contain every function.

So how large is eq. 4 compared to eq. 3? Call the difference between these two expressions $d(X,Y)$.

$$d(X,Y) = X\log_2 Y + X\log_2 X + Y\log_2 Y - (X+Y)\log_2(X+Y) - \log_2 X - \log_2 Y \tag{5}$$

$$= X\log_2 X - (X+Y)\log_2 \tfrac{X+Y}{Y} - \log_2 X - \log_2 Y$$

Setting $Y = \lambda X$,

$$d(X,Y) = X\log_2 X - X(1+\lambda)\log_2 \tfrac{(1+\lambda)X}{\lambda X} - \log_2 X - \log_2(\lambda X)$$

$$= X\log_2 X - X(1+\lambda)\log_2(1+\tfrac{1}{\lambda}) - 2\log_2 X - \log_2 \lambda$$

$$= (X-2)\log_2 X - X(1+\lambda)\log_2(1+\tfrac{1}{\lambda}) - \log_2 \lambda \tag{6}$$

Since $\lambda \leq 1$, the only negative term in eq. 6 is the $X(1+\lambda)\log_2(1+\frac{1}{\lambda})$ term. The function $g(\lambda) = (1+\lambda)\log_2(1+\frac{1}{\lambda})$ is monotonically decreasing in λ to an asymptotic value of $(\ln 2)^{-1}$, so its value is larger for smaller λ . However, the assumption $Y \geq 2$ gives a lower bound on λ, which means that the smallest possible value for eq. 6 under a given X occurs when $Y = 2$.

We have from eq. 5

$$d(X,Y) = X\log_2 Y + (Y-1)\log_2 Y - X(\log_2 \tfrac{X+Y}{X}) - Y\log_2(X+Y) - \log_2 X$$

and when $X \gg Y$ (e.g. when $Y = 2$ for large enough X), using

$$X \gg Y \implies (1 + \tfrac{Y}{X})^X \approx e^Y$$

we get

$$d(X,Y) \approx X\log_2 Y + (Y-1)\log_2 Y - Y\log_2 e - Y\log_2(X+Y) - \log_2 X$$

which is dominated by the $X\log_2 Y$ term; this term is the same as the entire length of the encoded function.

In other words, the class of n-compressible functions from $\mathcal{X}$ to $\mathcal{Y}$, where $n > 0$ and n is moderately less than the function's length, is not closed under permutation. By the c.u.p. result of [1], NFL does not apply to this class.

5. CONCLUSIONS

It was shown in [1] that there exist highly compressible classes of target function for which no search algorithm, including those inspired by evolution, can on average outperform random search. However, this paper shows that for the class of *all* n-compressible functions (given reasonable limits on n) under *any* encoding of search points and fitnesses, some search algorithms will indeed outperform others.

6. REFERENCES

[1] Schumacher, C., Vose, M.D., Whitley, L.D.: The no free lunch and problem description length. In Spector, L., Goodman, E.D., Wu, A., Langdon, W.B., Voigt, H.M., Gen, M., Sen, S., Dorigo, M., Pezeshk, S., Garzon, M.H., Burke, E., eds.: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001), San Francisco, California, USA, Morgan Kaufmann (2001) 565–570

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

[3] Droste, S., Jansen, T., Wegener, I.: Perhaps not a free lunch but at least a free appetizer. In Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., Smith, R.E., eds.: Proceedings of theFirstGenetic and Evolutionary Computation Conference(GECCO '99), San Francisco CA, Morgan Kaufmann Publishers, Inc. (1999) 833–839

[4] Whitley, D.: A free lunch proof for gray versus binary encodings. In Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., Smith, R.E., eds.: Proceedings of the Genetic and Evolutionary Computation Conference. Volume 1., Orlando, Florida, USA, Morgan Kaufmann (1999) 726–733

[5] English, T.M.: Some information theoretic results on evolutionary optimization. In Angeline, P.J., ed.: Proceedings of the 1999 Congress on Evolutionary Computation (CEC99), Washington D.C., Piscataway, NJ, IEEE Press (1999) 788–795

[6] Wiesmann, D.: (Optimization and information processing: NFL results)

[7] Igel, C., Toussaint, M.: On classes of functions for which no free lunch results hold. Information Processing Letters **86** (2003) 317–321 See also Los Alamos Preprint cs.NE/0108011.

[8] Woodward, J.R., Neil, J.R.: (No free lunch, program induction and combinatorial problems)

[9] Li, M., Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications. Second edn. Springer-Verlag (1997)