Measuring intelligence and growth rate: variations on Hibbard's intelligence measure

Samuel Alexander*

2020

Abstract

In 2011, Hibbard suggested an intelligence measure for agents who compete in an adversarial sequence prediction game. We argue that Hibbard's idea should actually be considered as two separate ideas: first, that the intelligence of such agents can be measured based on the growth rates of the runtimes of the competitors that they defeat; and second, one specific (somewhat arbitrary) method for measuring said growth rates. Whereas Hibbard's intelligence measure is based on the latter growth-rate-measuring method, we survey standard methods for measuring function growth rates, and exhibit the Hibbard-like intelligence measures which result from those. Of particular interest, we obtain intelligence measures based on Big-O notation and similar notation systems, which measures are novel in that they challenge conventional notions of what an intelligence measure should look like.

1 Introduction

In his insightful paper [13], Bill Hibbard introduces a novel intelligence measure (which we will here refer to as the *original Hibbard measure*) for agents who play a game of adversarial sequence prediction [12] "against a hierarchy of increasingly difficult sets of" evaders (environments that attempt to emit 1s and 0s in such a way as to evade prediction). The levels of Hibbard's hierarchy are labelled by natural numbers, and an agent's original Hibbard measure is the maximum $n \in \mathbb{N}$ such that said agent learns to predict all the evaders in the nth level of the hierarchy, or implicitly[1] an agent's original Hibbard measure is ∞ if said agent learns to predict all the evaders in all levels of Hibbard's hierarchy.

The hierarchy which Hibbard uses to measure intelligence is based on the growth rate of the runtime complexity of evaders. We will argue that Hibbard's idea is really a combination of two orthogonal ideas. First: that in some sense

*The U.S. Securities and Exchange Commission

[1]Hibbard does not explicitly include the ∞ case in his definition, but in his Proposition 3 he refers to agents having "finite intelligence", and it is clear from context that by this he means agents who fail to predict some evader somewhere in the hierarchy.

the intelligence of a predicting agent can be measured based on the growth rates of the runtimes of the evaders whom that predictor learns to predict. Second: Hibbard proposed one particular method for measuring said growth rates. The growth rate measurement which Hibbard proposed yields a corresponding intelligence measure for these agents. We will argue that *any* method for measuring growth rates of functions yields a corresponding Hibbard measure.

The particular method which Hibbard used to measure function growth rates is not very standard, nor (in our opinion) very useful. We will survey standard ways of measuring function growth rates, and these will yield corresponding Hibbard-like intelligence measures.

The structure of the paper is as follows.

- In Section 2, we review the original Hibbard measure.
- In Section 3, we argue that any method of measuring the growth rate of functions yields a Hibbard-like intelligence measure, and that the original Hibbard measure is just a special case resulting from one particular method of measuring function growth rate.
- In Section 4, we consider the most standard methods of measuring function growth rate—Big-O notation, Big-Θ notation, and Big-Ω notation—and define corresponding predictor intelligence taxonomies.
- In Section 5, we consider several numeric solutions to the problem of measuring the growth rate of functions (which use various extended number systems), and define corresponding Hibbard-like measures of predictor intelligence.
- In Section 6, we give pros and cons of different Hibbard-style measures.
- In Section 7, we summarize and make concluding remarks.

2 Hibbard's original measure

Hibbard proposed an intelligence measure for measuring the intelligence of agents who compete to predict evaders in a game of adversarial sequence prediction (we define this formally below). A predictor p (whose intelligence we want to measure) competes against evaders e. In each step of the game, both predictor and evader simultaneoulsy choose a binary digit, 1 or 0. Only after both of them have made their choice do they see which choice the other one made, and then the game proceeds to the next step. The predictor's goal in each step of the game is to choose the same digit that the evader will choose; the evader's goal is to choose a different digit than the predictor. The predictor wins the game (and is said to *learn to predict* e, or simply to *learn* e) if, after finitely many initial steps, eventually the predictor always chooses the same digit as the evader.

Definition 1. *By B, we mean the binary alphabet $\{0,1\}$. By B^*, we mean the set of all finite binary sequences. By $\langle\rangle$ we mean the empty binary sequence.*

Definition 2. *(Predictors and evaders)*

1. *By a* predictor*, we mean a Turing machine p which takes as input a finite (possibly empty) binary sequence $(x_1,\ldots,x_n) \in B^*$ (thought of as a sequence of* evasions*) and outputs 0 or 1 (thought of as a* prediction*), which output we write as $p(x_1,\ldots,x_n)$.*

2. *By an* evader*, we mean a Turing machine e which takes as input a finite (possibly empty) binary sequence $(y_1,\ldots,y_n) \in B^*$ (thought of as a sequence of* predictions*) and outputs 0 or 1 (thought of as an* evasion*), which output we write as $e(y_1,\ldots,y_n)$.*

3. *For any predictor p and evader e, the* result of p playing the game of adversarial prediction against e *(or more simply, the* result of p playing against e*) is the infinite binary sequence $(x_1,y_1,x_2,y_2,\ldots)$ defined as follows:*

   (a) *The first evasion $x_1 = e(\langle\rangle)$ is the output of e when run on the empty prediction-sequence.*

   (b) *The first prediction $y_1 = p(\langle\rangle)$ is the result of applying p to the empty evasion-sequence.*

   (c) *For all $n > 0$, the $(n+1)$th evasion $x_{n+1} = e(y_1,\ldots,y_n)$ is the output of e on the sequence of the first n predictions.*

   (d) *For all $n > 0$, the $(n+1)$th prediction $y_{n+1} = p(x_1,\ldots,x_n)$ is the result of applying p to the first n evasions.*

4. *Suppose $r = (x_1,y_1,x_2,y_2,\ldots)$ is the result of a predictor p playing against an evader e. For every $n \geq 1$, we say* the predictor wins round n in r *if $x_n = y_n$; otherwise,* the evader wins round n in r*. We say that p* learns to predict e *(or simply that p* learns e*) if there is some $N \in \mathbb{N}$ such that for all $n > N$, p is the winner of round n in r.*

Note that if e simply ignores its inputs $(y_1,\ldots,y_n)$ and instead computes $e(y_1,\ldots,y_n)$ based only on n, then e is essentially a sequence. Thus Definition 2 is a generalization of sequence prediction, which many authors have written about (such as Legg [16], who gives many references).

In the following definition, we differ from Hibbard's original paper because of a minor (and fortunately, easy-to-fix) error there.

Definition 3. *Suppose e is an evader. For each $n \in \mathbb{N}$, let $t_e(n)$ be the maximum number of steps that e takes to run on any length-n sequence of binary digits. In other words, $t_e(0)$ is the number of steps e takes to run on $\langle\rangle$, and for all $n > 0$,*

$$t_e(n) = \max_{b_1,\ldots,b_n \in \{0,1\}} (\textit{number of steps } e \textit{ takes to run on } (b_1,\ldots,b_n)).$$

Example 4. *Let e be an evader. Then $t_e(2)$ is equal to the number of steps e takes to run on input $(0,0)$, or to run on input $(0,1)$, or to run on input $(1,0)$, or to run on input $(1,1)$—whichever of these four possibilities is largest.*

Definition 5. *Suppose $f:\mathbb{N}\to\mathbb{N}$ and $g:\mathbb{N}\to\mathbb{N}$. We say $f\succ g$ if there is some $n_0\in\mathbb{N}$ such that for all $n>n_0$, $f(n)>g(n)$.*

Definition 6. *Suppose $f:\mathbb{N}\to\mathbb{N}$. We define E_f to be the set of all evaders e such that $f\succ t_e$.*

Definition 7. *(The original Hibbard measure) Let $g_1,g_2,\ldots$ be the enumeration of the primitive recursive functions given by Liu [17]. For each $m>0$, define $f_m:\mathbb{N}\to\mathbb{N}$ by*

$$f_m(k)=\max_{0<i\le m}\max_{j\le k} g_i(j).$$

For any predictor p, we define the original Hibbard intelligence *of p to be the maximum $m>0$ such that p learns to predict e for every $e\in E_{f_m}$ (or 0 if there is no such m, or ∞ if p learns to predict e for every $e\in E_{f_m}$ for every $m>0$).*

3 Quantifying growth rates of functions

The following is a very general and open-ended problem.

Problem 8. *Quantify the growth-rate of functions from $\mathbb{N}$ to $\mathbb{N}$.*

The definition of the original Hibbard measure (Definition 7) can be thought of as implicitly depending on a specific solution to Problem 8, which we make explicit in the following definition.

Definition 9. *For each $m>0$, let f_m be as in Definition 7. For each $f:\mathbb{N}\to\mathbb{N}$, we define the* original Hibbard growth rate *$H(f)$ to be $\min\{m>0\,:\,f_m\succ f\}$ if there is any such $m>0$, and otherwise $H(f)=\infty$.*

Lemma 10. *For every natural $m>0$ and every $f:\mathbb{N}\to\mathbb{N}$, $H(f)\le m$ if and only if $f_m\succ f$.*

Proof. Straightforward. □

Definition 11. *For every $m\in\mathbb{N}$, let E^H_m be the set of all evaders e such that $H(t_e)\le m$.*

Lemma 12. *For every natural $m>0$, $E^H_m=E_{f_m}$.*

Proof. Let e be an evader. By Definition 11, $e\in E^H_m$ if and only if $H(t_e)\le m$. By Lemma 10, $H(t_e)\le m$ if and only if $f_m\succ t_e$. But by Definition 6, this is the case if and only if $e\in E_{f_m}$. □

Corollary 13. *For every predictor p, the original Hibbard measure of p is equal to the maximum natural $m>0$ such that p learns e whenever $e\in E^H_m$, or is equal to ∞ if p learns e whenever $e\in E^H_m$ for all $m>0$.*

Proof. Immediate by Lemma 12 and Definition 7. $\square$

Remark 14. *Corollary 13 shows that the definition of the original Hibbard measure can be rephrased in such a way as to show that it depends in a uniform way on a particular solution to Problem 8, namely to the solution proposed by Definition 9. For* any *solution H' to Problem 8, we could define evader-sets $E_m^{H'}$ in a similar way to Definition 11, and, by copying Corollary 13, we could obtain a corresponding intelligence measure given by H' (it might be necessary to replace the "maximum" in Corollary 13 by a "supremum", if H' measures growth rates using a non-discrete number system, or transform the form of it if H' solves Problem 8 using a non-complete number system or by categorizing functions into classes rather than by assigning them numerical measurements, as in the case of Big-O notation). This formalizes what we claimed in the Introduction, that Hibbard's idea can be decomposed into two sub-ideas, firstly, that a predictor's intelligence can be measured in terms of the growth rates of the runtimes of the evaders it learns, and secondly, a particular method (Definition 9) of measuring those growth rates (i.e., a particular solution to Problem 8).*

4 Big-O, Big-Ω, and Big-Θ intelligence measurement

The standard solution to Problem 8 in computer science is to quantify growth rates of arbitrary functions by comparing them to more familiar functions using Big-O notation, Big-Ω notation, or Big-Θ notation. Knuth defines [15] these as follows (we modify the definition slightly because we are only concerned here with functions from $\mathbb{N}$ to $\mathbb{N}$).

Definition 15. *Suppose $f : \mathbb{N} \to \mathbb{N}$. We define the following function-sets.*

- *$O(f(n))$ is the set of all $g : \mathbb{N} \to \mathbb{N}$ such that there is some real $C > 0$ and some $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $g(n) \leq Cf(n)$.*
- *$\Omega(f(n))$ is the set of all $g : \mathbb{N} \to \mathbb{N}$ such that there is some real $C > 0$ and some $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $g(n) \geq Cf(n)$.*
- *$\Theta(f(n))$ is the set of all $g : \mathbb{N} \to \mathbb{N}$ such that there are some real $C > 0$ and $C' > 0$ and some $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, $Cf(n) \leq g(n) \leq C'f(n)$.*

We would argue that Definition 15 is a strictly better solution to Problem 8 than Definition 9. Definition 9 would clearly not be much use in practice. It is much more useful to say that an algorithm has, say, asymptotic runtime $O(n^3)$ than to say that said algorithm's runtime has original Hibbard complexity (say) 835. If someone were told that an algorithm's runtime had original Hibbard complexity (say) 835, that number would be totally meaningless to them except as a key which they could use to look up which function f_{835} happens to be (in Definition 7). The number 835 would merely serve to obfuscate, it would merely play the role of a worthless middleman.

By Remark 14, Definition 15 yields the following elegant measure of predictor intelligence.

Definition 16. *Suppose p is a predictor, and suppose $f : \mathbb{N} \to \mathbb{N}$.*

- *We say p* has Hibbard intelligence $O(f(n))$ *if p learns every evader e such that t_e is $O(f(n))$.*
- *We say p* has Hibbard intelligence $\Omega(f(n))$ *if p learns every evader e such that t_e is $\Omega(f(n))$.*
- *We say p* has Hibbard intelligence $\Theta(f(n))$ *if p learns every evader e such that t_e is $\Theta(f(n))$.*

At first glance, Definition 16 might seem inferior to Definition 7 because Definition 7 assigns *numerical* intelligence levels. However, as we pointed out above, those numbers are almost meaningless except as indices for a dictionary-lookup making them equivalent to (a more limited version of) Definition 16. We could imagine someone declaring:

> From now on, $f(n) = n$ is complexity 5, $f(n) = n^2$ is complexity 8, $f(n) = 2^n$ is complexity 203, $f(n) = n!$ is complexity 8022, ..., and from now on, instead of saying a function is $\Theta(n)$, we will say that function has growth rate 5, and instead of saying a function is $\Theta(n^2)$, we will say that function has growth rate 8, and instead of saying a function is $\Theta(2^n)$, we will say that function has growth rate 203, and instead of saying a function is $\Theta(n!)$, we shall say that function has growth rate 8022, and ...

Superficially, this would "improve" upon Big-Θ by giving a "more quantitative" solution to Problem 8, however, it is clear that this would just be sleight of hand and there would be no actual improvement.

5 Hibbard measures using extended number systems

We have argued above that it is silly to try to replace asymptotic complexity classes with natural numbers. In [3], we go further, and cite asymptotic runtime complexity as an example of a *non-Archimedean* measure which, we argue, implies that it is silly to replace asymptotic complexity classes even with *real* numbers. There are, however, non-Archimedean number systems flexible enough to measure some or all growth rates, and these alternative solutions to Problem 8 would in turn yield Hibbard-style intelligence measures taking values from said non-Archimedean number systems.

- Hyperreal numbers, studied in the field of non-standard analysis [18] [9], are equivalence classes of infinite sequences of reals, so for every sequence $(r_0, r_1, r_2, \ldots)$ of reals, there is a corresponding hyperreal represented by that sequence. Thus, there is a natural and elegant way to solve Problem 8 using hyperreal numbers. Namely: the growth rate of $f(n)$ is the hyperreal number represented by $(f(0), f(1), f(2), \ldots)$. This solution to Problem 8 will yield an approximate Hibbard-style intelligence measure similar to Definition 16 (the reason this only yields an approximate Hibbard-style intelligence measure is because the hyperreals are not complete).

- The surreal numbers [6] [14] are an even larger extension of the real numbers, into which the hyperreals can be embedded [7]. Thus, for any such embedding, there is another natural and elegant way to solve Problem 8, namely: the growth rate of $f(n)$ is the surreal number corresponding to the hyperreal number represented by $(f(0), f(1), f(2), \ldots)$ under the embedding. The advantage of the surreal numbers is that they are complete, which will allow the corresponding Hibbard-style measure to be exact, not approximate.
- Another natural way to measure growth rate is using majorization hierarchies (such as the slow-growing hierarchy or the fast-growing hierarchy [21]) from mathematical logic. A majorization hierarchy assigns ordinal number values to growth rates of functions (but not to all functions—for any particular majorization hierarchy, there are functions which grow too fast for that hierarchy). These yield corresponding Hibbard-style intelligence measures which are ordinal-number-valued.
- In order to reduce the arbitrary limitations of traditional majorization hierarchies, we could delegate the problem of defining a majorization hierarchy to an Artificial General Intelligence (or AGI). This leads to a family of Hibbard-style measures—one measure $|\bullet|_X$ for each AGI X, so that for any particular AGI X, for any predictor p, there is a corresponding intelligence measure $\|p\|_X$ which might be thought of as "p's intelligence as judged by X".

5.1 Hyperreal Hibbard intelligence

The so-called *ultrapower construction* of the hyperreals depends on an object called a *free ultrafilter*, which is a set of subsets of $\mathbb{N}$ satisfying certain requirements. It is not important for the purposes of this paper to define what a free ultrafilter is here. The important things to know are:

- A free ultrafilter $\mathscr{U}$ provides a notion of what it means for a subset $S \subseteq \mathbb{N}$ to be "large". Namely: S is considered to be "large" if and only if $S \in \mathscr{U}$.
- Mathematical logicians have proven that free ultrafilters exist, but that it is impossible to constructively exhibit one. This makes definitions based on free ultrafilters non-constructive, and computationally impractical.

Because of the computational impracticality of free ultrafilters, the following notion is also computationally impractical. However, it could potentially be useful for proving theoretical properties about the intelligence of predictors. In the following definition, rather than assigning a particular hyperreal number intelligence to every predictor, rather, we categorize predictors into classes. This is necessary because the hyperreals are not complete, so Corollary 13 cannot directly be mimicked either with "maximum" or with "supremum".

Definition 17. *(Hyperreal Hibbard intelligence) The* hyperreal Hibbard intelligence *of a predictor p is defined to be $\geq$ a hyperreal number r if and only if p learns every evader e such that if r' is the hyperreal number represented by $(t_e(0), t_e(1), t_e(2), \ldots)$ then $r' < r$.*

5.2 Surreal Hibbard intelligence

The surreal number system is even more flexible than the hyperreal number system. There are many ways to embed the hyperreals into the surreals, and there is no standard or canonical way which stands out. In this subsection, we fix some embedding of the hyperreals into the surreals.

A key property of the surreals is that they are complete. Thus, given any set S of surreals with a surreal upper bound, there is a *least* surreal upper bound of S, called the *supremum* of S. This allows for a Hibbard-style intelligence measure which is more exact than Definition 17.

Definition 18. *(Surreal Hibbard intelligence) For every predictor p, the* surreal Hibbard intelligence *of p is equal to the supremum of all surreal numbers r such that p learns e whenever the surreal number corresponding to $(t_e(0), t_e(1), t_e(2), \ldots)$ is $< r$.*

5.3 Hibbard intelligence and the traditional majorization hierarchies

Majorization hierarchies [21] provide ordinal-number-valued measures for the growth rate of certain functions. A majorization hierarchy depends on many infinite-dimensional parameters. For illustrative purposes, we will describe the slow-growing hierarchy up to the ordinal ϵ_0, using standard choices for the parameters.

Definition 19. *(Classification of ordinal numbers) Ordinal numbers are divided into three types:*

1. *Zero: The ordinal* 0*.*
2. *Successor ordinals: Ordinals of the form $\alpha + 1$ for some ordinal α.*
3. *Limit ordinals: Ordinals which are not successor ordinals nor* 0*.*

For example, the smallest infinite ordinal, ω, is a limit ordinal. It is not zero (because zero is finite), nor can it be a successor ordinal, because if it were a successor ordinal, say, $\alpha + 1$, then α would be finite (since ω is the *smallest* infinite ordinal), but then $\alpha + 1$ would be finite as well.

The ordinal ϵ_0 is the smallest ordinal bigger than the ordinals $\omega, \omega^\omega, \omega^{\omega^\omega}, \ldots$. It satisfies the equation $\epsilon_0 = \omega^{\epsilon_0}$ and can be intuitively thought of as

$$\epsilon_0 = \omega^{\omega^{\omega^{\cdot^{\cdot^{\cdot}}}}}.$$

Ordinals below ϵ_0 include such ordinals as ω, $\omega^{\omega+1} + \omega^\omega + \omega^5 + 3$,

$$\omega^{\omega^{\omega^{\omega^\omega}}} + \omega^{\omega^{\omega^\omega} + \omega^{\omega\cdot 2+1} + \omega^4 + 3} + \omega^{\omega^5 + \omega^3} + \omega^8 + 1,$$

and so on. Any ordinal below ϵ_0 can be uniquely written in the form

$$\omega^{\alpha_1} + \omega^{\alpha_2} + \cdots + \omega^{\alpha_k}$$

where $\alpha_1 \geq \cdots \geq \alpha_k$ are smaller ordinals below ϵ_0—this form for an ordinal below ϵ_0 is called its *Cantor normal form.* For example, the Cantor normal form for $\omega^{\omega\cdot 2} \cdot 2 + \omega \cdot 3 + 2$ is

$$\omega^{\omega\cdot 2} \cdot 2 + \omega \cdot 3 + 2 = \omega^{\omega\cdot 2} + \omega^{\omega\cdot 2} + \omega^1 + \omega^1 + \omega^1 + \omega^0 + \omega^0.$$

Definition 20. *(Standard fundamental sequences for limit ordinals $\leq \epsilon_0$) Suppose λ is a limit ordinal $\leq \epsilon_0$. We define a* fundamental sequence for λ*, written $(\lambda[0], \lambda[1], \lambda[2], \ldots)$, inductively as follows.*

- *If $\lambda = \epsilon_0$, then $\lambda[0] = 0$, $\lambda[1] = \omega^0$, $\lambda[2] = \omega^{\omega^0}$, and so on.*
- *If λ has Cantor normal form $\omega^{\alpha_1} + \cdots + \omega^{\alpha_k}$ where $k > 1$, then each*

$$\lambda[i] = (\omega^{\alpha_1} + \cdots + \omega^{\alpha_{k-1}}) + (\omega^{\alpha_k})[i].$$

- *If λ has Cantor normal form $\omega^{\alpha+1}$, then each $\lambda[i] = \omega^\alpha \cdot i$.*
- *If λ has Cantor normal form ω^{λ_0} where λ_0 is a limit ordinal, then each $\lambda[i] = \omega^{\lambda_0[i]}$.*

Example 21. *(Fundamental sequence examples)*

- *The fundamental sequence for $\lambda = \omega = \omega^1 = \omega^{0+1}$ is $\omega^0 \cdot 0, \omega^0 \cdot 1, \omega^0 \cdot 2, \ldots$, i.e., $0, 1, 2, \ldots$.*
- *The fundamental sequence for $\lambda = \omega^5$ is $0, \omega^4, \omega^4 \cdot 2, \omega^4 \cdot 3, \ldots$.*
- *The fundamental sequence for $\lambda = \omega^\omega$ is $\omega^0, \omega^1, \omega^2, \ldots$.*
- *The fundamental sequence for $\lambda = \omega^\omega + \omega$ is $\omega^\omega + 0, \omega^\omega + 1, \omega^\omega + 2, \ldots$.*

Definition 22. *(The standard slow-growing hierarchy up to ϵ_0) We define functions $g_\beta : \mathbb{N} \to \mathbb{N}$ (for all ordinals $\beta \leq \epsilon_0$) by transfinite induction as follows.*

- *$g_0(n) = 0$.*

- $g_{\alpha+1}(n) = g_\alpha(n) + 1$.
- $g_\lambda(n) = g_{\lambda[n]}(n)$ *if* λ *is a limit ordinal.*

Here are some early levels in the slow-growing hierarchy, spelled out in detail.

Example 23. *(Early examples of functions in the slow-growing hierarchy)*

1. $g_1(n) = g_{0+1}(n) = g_0(n) + 1 = 0 + 1 = 1$.
2. $g_2(n) = g_{1+1}(n) = g_1(n) + 1 = 1 + 1 = 2$.
3. *More generally, for all* $m \in \mathbb{N}$, $g_m(n) = m$.
4. $g_\omega(n) = g_{\omega[n]}(n) = g_n(n) = n$.
5. $g_{\omega+1}(n) = g_\omega(n) + 1 = n + 1$.
6. $g_{\omega+2}(n) = g_{\omega+1}(n) + 1 = (n + 1) + 1 = n + 2$.
7. *More generally, for all* $m \in \mathbb{N}$, $g_{\omega+m}(n) = n + m$.
8. $g_{\omega\cdot 2}(n) = g_{(\omega\cdot 2)[n]}(n) = g_{\omega+n}(n) = n + n = n \cdot 2$.

Following Example 23, the reader should be able to fill in the details in the following example.

Example 24. *(More examples from the slow-growing hierarchy)*

1. $g_{\omega^2}(n) = n^2$.
2. $g_{\omega^3}(n) = n^3$.
3. $g_{\omega^\omega}(n) = n^n$.
4. $g_{\omega^{\omega\cdot 3+1}+\omega+5}(n) = n^{3n+1} + n + 5$.
5. $g_{\omega^{\omega^\omega}}(n) = n^{n^n}$.

What about g_{ϵ_0}? Thinking of ϵ_0 as

$$\omega^{\omega^{\omega^{\cdot^{\cdot^{\cdot}}}}},$$

one might expect $g_{\epsilon_0}(n)$ to be

$$n^{n^{n^{\cdot^{\cdot^{\cdot}}}}},$$

but such an infinite tower of exponents makes no sense if $n > 1$. Instead, the answer defies familiar mathematical notation.

Example 25. *(Level* ϵ_0 *in the slow-growing hierarchy) The values of* g_{ϵ_0} *are as follows:*

- $g_{\epsilon_0}(0) = 0$.

- $g_{\epsilon_0}(1) = 1^1$.
- $g_{\epsilon_0}(2) = 2^{2^2}$.
- $g_{\epsilon_0}(3) = 3^{3^{3^3}}$.
- *And so on.*

Example 25 illustrates how the slow-growing hierarchy can systematically lead to fast-growing computable functions. By extending the slow-growing hierarchy to larger ordinals (and choosing appropriate fundamental sequences for those larger ordinals), one can obtain stupendously fast-growing functions in this way. These in turn serve as a partial solution to Problem 8: we can say that the growth rate of an arbitrary function $f : \mathbb{N} \to \mathbb{N}$ is equal to α, where α is the smallest ordinal such that $g_\alpha \succ f$ (or ∞ if no such α exists). This in turn provides a corresponding Hibbard-style measure.

Definition 26. *If p is a predictor, the* Hibbard intelligence of X given by the standard slow-growing hierarchy up to ϵ_0 *is defined to be the supremum of all ordinals $\alpha < \epsilon_0$ such that p learns every evader e such that $g_\alpha \succ t_e$ (or to ∞ if said property holds for all ordinals $\alpha < \epsilon_0$).*

5.4 Generalized majorization hierarchies

There are many other majorization hierarchies besides the slow-growing hierarchy. In Definition 22, there is nothing special about defining $g_{\alpha+1}(n) = g_\alpha(n)+1$. Many other definitions for $g_{\alpha+1}$ would work, provided $g_{\alpha+1}$ ends up being faster-growing than g_α. For example, in the so-called *fast-growing hierarchy*, $g_{\alpha+1}(n)$ is defined to be $g^n_\alpha(n)$, where g^n_α denotes the result of iterating g_α n times, that is, $g^1_\alpha(n) = g_\alpha(n)$, $g^2_\alpha(n) = g_\alpha(g_\alpha(n))$, $g^3_\alpha(n) = g_\alpha(g_\alpha(g_\alpha(n)))$, and so on. In the early levels, this produces much faster-growing functions, but astonishingly, at sufficiently high ordinals, the slow-growing hierarchy actually catches up with the fast-growing hierarchy [8] (a beautiful illustration of how counter-intuitive large ordinal numbers can be). One of the earliest majorization hierarchies is the Hardy hierarchy [11], where $g_{\alpha+1}(n) = g_\alpha(n + 1)$.

Likewise, there is also much flexibility in choosing fundamental sequences. For small ordinals such as the ordinals below ϵ_0, there are very clear canonical fundamental sequences, but the larger the ordinals become, the harder it becomes to single out any choice of fundamental sequences as canonical. And even at low levels, choosing non-canonical fundamental sequences can drastically alter the resulting majorization hierarchy [20].

In short, there is no consensus at all about which majorization hierarchies are most canonical. If anything, there is consensus that there is no consensus. However, Hibbard-style intelligence measures are primarily of interest to researchers interested in agents with Artificial General Intelligence (that is, researchers interested in AGIs). No-one knows what exactly an AGI is, but presumably an

AGI can be thought of[2] as a patient, obedient, careful, mechanical employee who can be given commands in English, and who will follow those commands (at least when the commands are *possible* to follow—one could command an AGI to act as a Halting Problem solver, but it is unclear how the AGI would respond to such a command, since no mechanical agent can solve the Halting Problem).

Rather than attempt the futile task of choosing a canonical majorization hierarchy ourselves, we can instead delegate that task to an AGI. By doing so for an arbitrary AGI, we will obtain an AGI-indexed family of semi-canonical majorization hierarchies.

Definition 27. *Suppose X is an AGI. By $h^X_1, h^X_2, \ldots$, we mean the total computable increasing functions from $\mathbb{N}$ to $\mathbb{N}$ which X would output if X were commanded:*

> *"Until further notice, output (codes of) total computable increasing functions $h_1, h_2, \ldots$ from $\mathbb{N}$ to $\mathbb{N}$ satisfying the following constraints:*
>
> 1. *(Linear ordering) For any two $i, j \in \mathbb{N}$ with $i \neq j$, either $h_i \succ h_j$ or $h_j \succ h_i$.*
> 2. *(Largeness) For every Turing machine M such that you know M computes a total computable function $h : \mathbb{N} \to \mathbb{N}$, there is some i such that $h_i \succ h$.*
> 3. *(Well-foundedness) There is no sequence $i_1, i_2, \ldots$ such that $h_{i_1} \succ h_{i_2} \succ \cdots$.*
> 4. *(Pseudo-density) The h_i's are to be as close as possible to being* dense*—that is to say, they are to be as close as possible to having the property that for all $i, k \in \mathbb{N}$ with $h_k \succ h_i$, there is some $j \in \mathbb{N}$ such that $h_k \succ h_j \succ h_i$—without violating the above well-foundedness constraint; you are to use your judgment and discretion to interpret this pseudo-density requirement."*

The above pseudo-density constraint is intentionally vague, which we can get away with because the definition does not depend on what pseudo-density actually means, but only on how the AGI interprets the *words* (i.e., how the AGI responds to a specific well-defined stimulus, regardless of how vague the semantics of that stimulus may be). Thus, we are taking advantage of the AGI's ability to "adapt with insufficient knowledge and resources" [19]. We assume the AGI understands and obeys the command, so $h^X_1, h^X_2, \ldots$ really are total computable increasing functions from $\mathbb{N}$ to $\mathbb{N}$ satisfying the linear ordering, largeness[3], and well-foundedness requirements. Note that the resulting h_i's cannot actually be dense, as that would violate well-foundedness: given any i, j with $h_j \succ h_i$, there would be some k_1 such that $h_j \succ h_{k_1} \succ h_i$, and then there

[2] Our thinking here is reminiscent of some remarks of Yampolskiy [22].

[3] Note that the largeness requirement is defined in terms of the AGI's mathematical knowledge. See [4] for a definition of what it means for an AGI to know a mathematical fact.

would be some k_2 such that $h_{k_1} \succ h_{k_2} \succ h_i$, and then there would be some k_3 such that $h_{k_2} \succ h_{k_3} \succ h_i$, and so on, but then $h_{k_1} \succ h_{k_2} \succ \cdots$ would violate well-foundedness. We will offer some motivation for pseudo-density in Remark 32 below.

Definition 28. *Let X be an AGI. We define a sequence $\alpha_1^X, \alpha_2^X, \ldots$ of ordinals by recursion as follows. Note that at first glance, the following recursive definition looks too circular to work. We will show in Lemma 29 that it avoids infinite regress and thus it does work. For each i, let α_i^X be the the smallest ordinal which is larger than every ordinal α_j^X such that $h_i^X \succ h_j^X$.*

Lemma 29. *Definition 28 works (it does not lead to infinite regress).*

Proof. Assume, for the sake of contradiction, that Definition 28 leads to infinite regress. This means there is some i_1 such that in order to define $\alpha_{i_1}^X$ we must first define $\alpha_{i_2}^X$ for some i_2, and in order to define $\alpha_{i_2}^X$ we must first define $\alpha_{i_3}^X$ for some i_3, and in order to define $\alpha_{i_3}^X$ we must first define $\alpha_{i_4}^X$ for some i_4, and so on forever. Thus, there is an infinite sequence $i_1, i_2, \ldots$ such that in order to define each $\alpha_{i_j}^X$, we must first define $\alpha_{i_{j+1}}^X$. Now, each $\alpha_{i_j}^X$ is defined as the smallest ordinal larger than every ordinal α_j^X such that $h_{i_j}^X \succ h_j^X$. So, since defining $\alpha_{i_j}^X$ requires us to first define $\alpha_{i_{j+1}}^X$, this implies i_{j+1} is such a j, i.e., that $h_{i_j}^X \succ h_{i_{j+1}}^X$. Thus $h_{i_1}^X \succ h_{i_2}^X \succ h_{i_3}^X \succ \cdots$, but this contradicts the *well-foundedness* part of the command from Definition 27. □

Lemma 30. *Let X be an AGI. For all i, j, $h_i^X \succ h_j^X$ if and only if $\alpha_i^X > \alpha_j^X$.*

Proof. Fix i and j. By the *linear ordering* part of the command in Definition 27, either $h_i^X \succ h_j^X$ or $h_j^X \succ h_i^X$. Assume $h_i^X \succ h_j^X$, the other case is similar. By definition, α_i^X is defined to be the smallest ordinal which is larger than $\alpha_{j'}^X$ for every j' such that $h_i^X \succ h_{j'}^X$. One such j' is j itself, so by definition α_i^X is larger than α_j^X, as desired. □

Definition 31. *(Generalized majorization hierarchies) Let X be an AGI. Let $\alpha = \sup_i \alpha_i^X$ be the smallest ordinal bigger than all of the α_i^X's. The* generalized majorization hierarchy given by X *is the family $(g_\beta)_{\beta<\alpha}$ of functions labeled by ordinals below α, where each g_β is defined such that $g_\beta = h_i^X$ where i is such that $\beta = \alpha_i^X$.*

Remark 32. *With Definition 31 in view, the pseudo-density constraint in Definition 27 can be better motivated: the point of pseudo-density is to ensure that the ordinal $\sup_i \alpha_i^X$ is as large as possible*[4]*. An alternative to pseudo-density would be to directly command (in Definition 27) that the h_i's should be chosen in such a way as to make $\alpha = \sup_i \alpha_i^X$ large (leaving it up to the AGI's judgment and discretion how to interpret that), but this would be difficult because it would*

[4] Chaitin [5] and Good [10] have remarked about the importance of the size of the ordinals which an entity can notate—"and the larger the ordinal, the fitter the organism", to quote Chaitin. See also [1] and [2].

mean that Definition 31 (and its dependencies) would need to be embedded into the command in Definition 27, which would require some significant acrobatics.

A generalized majorization hierarchy (Definition 31) serves as a solution to Problem 8: given any AGI X, we can say that a function $f : \mathbb{N} \to \mathbb{N}$ has growth rate β where β is the smallest ordinal $< \sup_i \alpha_i^X$ such that $g_\beta \succ f$—or f has growth rate ∞ if there is no such β. To this solution to Problem 8, there is a corresponding Hibbard-style intelligence measure.

Definition 33. *(Hibbard intelligence given by an AGI) Suppose X is an AGI. By the* transfinite Hibbard measure given by X, *we mean the measure $\|\bullet\|_X$ which assigns to every predictor p an intelligence measure $\|p\|_X$ defined as follows. $\|p\|_X$ is defined to be the smallest $\beta < \sup_i \alpha_i^X$ such that for all $\gamma < \beta$, p learns every evader e such that $g_\gamma \succ e$ (or $\|p\|_X = \infty$ if p learns every evader e such that $g_\gamma \succ e$ for any $\gamma < \sup_i \alpha_i^X$).*

6 Pros and cons of different Hibbard-style measures

Here are pros and cons of the Hibbard measures which arise from different solutions to the problem (Problem 8) of measuring the growth rate of functions.

- The original Hibbard measure (Definition 7), which arises by measuring growth rate by comparing a function with one particular enumeration [17] of the primitive recursive functions:
  - Pro: Relatively concrete.
  - Pro: Measures intelligence using a familiar number system (the natural numbers).
  - Con: The numbers which the measure outputs are not very meaningful, in the sense that predictor p having a measure of just $+1$ higher than predictor q tells us absolutely nothing about how *much* more computationally complex the evaders which p learns are, versus the evaders which q learns.
  - Con: Only distinguishes sufficiently non-intelligence predictors; all predictors sufficiently intelligent receive measure ∞.
- Big-O/Big-Θ/Big-Ω (Definition 16), in which, rather than directly measuring the intelligence of a predictor, instead, we would talk of a predictor's intelligence being $O(f(n))$, or $\Theta(f(n))$, or $\Omega(f(n))$, for various functions $f : \mathbb{N} \to \mathbb{N}$:
  - Pro: Gets directly at the underlying concept, without obfuscation.
  - Pro: Computer scientists already use Big-O/Big-Θ/Big-Ω routinely and are comfortable with them.

  - Con: This option is not really a measure, but more of a taxonomy—and a non-numerical taxonomy at that.

- Hyperreal intelligence (Definition 17):
  - Pro: A taxonomy like Big-O/Big-Θ/Big-Ω, but with the added benefit that the taxons are numerical (hyperreal numerical, to be more precise).
  - Con: Depends on a non-constructive choice of a free ultrafilter (rendering it impractical for any actual computation).

- Surreal intelligence (Definition 18):
  - Pro: An actual numerical measure (not just a taxonomy), with the same perfect precision as the Big-O/Big-Θ/Big-Ω approach.
  - Con: Abstract and impractical: depends not only on a non-constructive choice of a free ultrafilter, but also on an embedding of the hyperreals into the surreals.
  - Con: The numbers which the measure outputs are surreal numbers, which may be unfamiliar to some users.

- Intelligence based on a standard majorization hierarchy such as the standard slow-growing hierarchy up to ϵ_0 (Definition 26):
  - Pro: A numerical measure, albeit without as much precision as the Big-O/Big-Θ/Big-Ω taxonomy.
  - Pro: Relatively concrete.
  - Pro: The numbers which the measure outputs are meaningful, in the sense that the degree to which a predictor p is more intelligent than a predictor q is reflected in the degree to which p's intelligence-measure is larger than q's.
  - Con: The numbers which the measure outputs are ordinal numbers, which may be unfamiliar to some users.
  - Con: Only distinguishes sufficiently non-intelligence predictors; all predictors sufficiently intelligent receive measure ∞.

- Intelligence based on a generalized majorization hierarchy (Definition 31):
  - Pro: A numerical measure, albeit without as much precision as the Big-O/Big-Θ/Big-Ω taxonomy.
  - Pro: The numbers which the measure outputs are meaningful, in the sense that the degree to which a predictor p is more intelligent than a predictor q is reflected in the degree to which p's intelligence-measure is larger than q's.

– Pro: Potentially capable of assigning non-∞ intelligence measures to very advanced predictors (the more intelligent the AGI used to generate the generalized majorization hierarchy, the more advanced the predictors can be and still receive non-∞ intelligence).
– Con: The numbers which the measure outputs are ordinal numbers, which may be unfamiliar to some users.
– Con: Actually realizing one of these measures would require access to an AGI.

7 Summary and Conclusion

To summarize:

- Hibbard proposed [13] an intelligence measure for predictors in games of adversarial sequence prediction.
- We argued that Hibbard's idea actually splits into two orthogonal sub-ideas. Firstly: that intelligence can be measured via the growth-rate of the run-times of evaders that a predictor can learn to predict. Secondly: that such growth-rate can be measured in one specific way (involving an enumeration of the primitive recursive functions). We argue that there are other, more standard ways to measure growth-rates, and that each method of measuring growth-rates yields a corresponding Hibbard-style intelligence measure.
- We considered several specific ways of measuring growth-rate of functions, and exhibited corresponding Hibbard-style intelligence measures. The growth-rate-measuring methods which we considered were: Big-O/Big-Θ/Big-Ω notation; hyperreal numbers; surreal numbers; standard majorization hierarchies; and (in their debut appearance here) generalized majorization hierarchies.

References

[1] Samuel Allen Alexander. Measuring the intelligence of an idealized mechanical knowing agent. In *CIFMA*, 2019.

[2] Samuel Allen Alexander. AGI and the Knight-Darwin law: why idealized AGI reproduction requires collaboration. In *ICAGI*, 2020.

[3] Samuel Allen Alexander. The Archimedean trap: Why traditional reinforcement learning will probably not yield AGI. *arXiv preprint arXiv:2002.10221*, 2020.

[4] Samuel Allen Alexander. Short-circuiting the definition of mathematical knowledge for an artificial general intelligence. *In preparation*, 2020.

[5] Gregory Chaitin. Metaphysics, metamathematics and metabiology. In Zenil Hector, editor, *Randomness through computation*. World Scientific, 2011.

[6] John H Conway. *On Numbers and Games*. CRC Press, 2nd edition, 2000.

[7] Philip Ehrlich. The absolute arithmetic continuum and the unification of all numbers great and small. *Bulletin of Symbolic Logic*, 18:1–45, 2012.

[8] Jean-Yves Girard. Π^1_2-logic, part 1: Dilators. *Annals of Mathematical Logic*, 21(2-3):75–219, 1981.

[9] Robert Goldblatt. *Lectures on the hyperreals: an introduction to nonstandard analysis*. Springer Science & Business Media, 2012.

[10] Irving John Good. Gödel's theorem is a red herring. *The British Journal for the Philosophy of Science*, 19(4):357–358, 1969.

[11] Godfrey H Hardy. A theorem concerning the infinite cardinal numbers. *Quarterly journal of Mathematics*, 35:87–94, 1904.

[12] Bill Hibbard. Adversarial sequence prediction. In *ICAGI*, pages 399–403, 2008.

[13] Bill Hibbard. Measuring agent intelligence via hierarchies of environments. In *ICAGI*, pages 303–308, 2011.

[14] Donald Ervin Knuth. *Surreal numbers: a mathematical novelette*. Addison-Wesley, 1974.

[15] Donald Ervin Knuth. Big Omicron and big Omega and big Theta. *ACM Sigact News*, 8(2):18–24, 1976.

[16] Shane Legg. Is there an elegant universal theory of prediction? In *International Conference on Algorithmic Learning Theory*, pages 274–287. Springer, 2006.

[17] Shih-Chao Liu. An enumeration of the primitive recursive functions without repetition. *Tohoku Mathematical Journal*, 12(3):400–402, 1960.

[18] Abraham Robinson. *Non-standard analysis*. Princeton University Press, 1974.

[19] Pei Wang. On defining artificial intelligence. *Journal of Artificial General Intelligence*, 10(2):1–37, 2019.

[20] Andreas Weiermann. Sometimes slow growing is fast growing. *Annals of Pure and Applied Logic*, 90(1-3):91–99, 1997.

[21] Andreas Weiermann. Slow versus fast growing. *Synthese*, 133:13–29, 2002.

[22] Roman V Yampolskiy. Turing test as a defining feature of AI-completeness. In *Artificial intelligence, evolutionary computing and metaheuristics*, pages 3–17. Springer, 2013.