MUTUAL EXPLAINABILITY, OR, A COMEDY IN COMPUTERLAND

Simon DeDeo*
Department of Social and Decision Sciences, Carnegie Mellon University, Pittsburgh, PA 15213
Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501

April 19, 2020

ABSTRACT

Many people believe that the universe is a giant computer. This view implies that what is uncomputable has no causal power. We propose that, to the contrary, at least one uncomputable object, Kolmogorov Complexity, does play a causal role in the physical world, and that we have good scientific reasons to therefore believe it exists. We use a simple set of arguments, based on the probabilistic extension of algorithmic information theory, to show that such a causal role is not only consistent with the best evidence from cosmology, but also predicts an otherwise mysterious feature of our environment: mutual explainability. Mutual explainability is the fact that things that tend to correlate with each other also tend to explain each other.

There have always been things we will never know, but the 20th Century discovered a new variety. These are claims about the world that contain self-reference, things that, at some point, start to talk about what they are talking about.

As a species, we did not anticipate this. We thought self-reference was a way to get clarity, not confusion, and the ability to talk about what we are talking about seemed to be a good thing. We think we make our arguments better, for example, in part by talking about how they work, talking about that talking, and so on. In Europe, this appears with the Socratic dialogues, and similar ideas define all the major surviving civilizations. The same traditions talk a great deal about the unknowable, but also—usually—prescribe ways of beating it back, though meditation and self-starvation, or drinking and falling in love.

The new unknowables resisted those solutions. At first, happily, they were curiosities: outré mathematical facts (whether or not, say, a certain equation crossed the zero axis at an integer), statements about the infinite (whether or not this or that computer program would ever finish running), or worries about an ultimate Black Swan (the possibility that two plus two might be provably equal to five).

Two things changed that forever. First, we began to think of ourselves, and our societies, as computers. Second, we began to think of the physical world itself as a computer simulation. In general, when people take this perspective—call it "computerland"—they stop believing in every kind of unknowability except the one that comes from self-reference.

The goal of this essay is to show what happens next, and how it plays out for a core unknowable of computerland, Kolmogorov Complexity. Kolmogorov Complexity is a double whammy because not only is it unknowable, it keeps turning up when you try to go about learning ordinary things.

An attraction of computerland is that it doubles down on the Enlightenment idea that knowledge is power, and knowledge is unbounded. The uncomputability of Kolmogorov Complexity throws residents of computerland back onto the boundedness they arrived with. In an analogy we develop below, it is the computer sciences' Lorentz Invariance, making any claim to knowledge dependent upon a background, and every background equally preferred.

Great progress in physics came from taking relativity seriously. We ought to expect something similar here: success in the project of general artificial intelligence may require we take seriously the relativity implied by self-reference.

If we do so, we not only see what we can't do—produce universal, unbiased reasoners—but also what we can. Further, an unexpected confluence of evidence from, on the one hand, our experience of the world as cognitive agents and, on the other, scientific knowledge in cosmology, suggests the unknowable may even play a causal role in making the universe what it is.

*sdedeo@andrew.cmu.edu; http://santafe.edu/~simon

1 On having good taste

Kolmogorov Complexity (KC) is the computerland version of asking how *complex* something is. Let's first understand why the answer matters so much.

We do the non-computerland version of KC all the time. Consider how we go about explaining things—say, why a friend was late for dinner. Good explanations have many properties, but one of the simplest is what cognitive scientists call power: a good explanation should account well for the facts at hand. Say I weigh two explanations for why a guest was late for dinner: (1) the buses stopped running, and he had to walk, or (2) his bicycle had a flat tire, and he stopped to fix it. If I look out the window and see buses going by, explanation (1) might look less secure than (2), because it has trouble accounting for the fact that the buses appear to still be running.

All other things being equal, a more powerful explanation is a better one. But things are rarely equal, and there are many other virtues of a good explanation. A competing one is simplicity.

Return to our two explanations above. After I see buses going by, (1) drops in acceptability relative to (2). It is not, however, ruled out, and I can revive it by saying that buses *have* stopped running, and we're just seeing them head back empty to the depot. Watching a little longer, I might see some of them heading away from the depot (trouble again), but I can rescue that by saying that the depot is full, and some of them were diverted to the overflow lot.

At some point, my friend might object: "come now—isn't the simpler explanation that he rode his bike this time?" This is different from the objection that the buses are unlikely to have stopped running so early, or that the depot is unlikely to be full. It argues, rather, that the explanation is bad because it is complicated.

This is a familiar move, and an example of Occam's Razor, the heuristic that biases us against complicated explanations. We teach Occam's Razor in schools, invoke it in science, and use it in debate. The very idea of making sense of something is tied up in the possibility of finding a simple explanation. Conversely, people whose simplicity module is broken are prone to conspiracy theories. The problem with a conspiracy theory, from an epistemic point of view, is often *not* that it can't account for facts—some even gain attention because they explain what the official story can not. It is rather that, in doing so, they become too complicated to be true.

Complexity is also a quality of the things we try to explain. Having come up with a long list of explanations for a phenomenon, we might come to believe that the correct one—the one that accounts for what we see—is extremely complex. There may be simpler explanations, but they don't work. At this point, we might want to say that the thing itself has high complexity.

Is complexity good or bad? For some people, good: a well-written novel ought to have no simpler adequate paraphrase. Similar things might be said of the beauty of a forest, or the arc of a human life. Others will say simplicity is good: the existence of a simple explanation for why flowers look the way they do makes the garden more beautiful. One might try to split the difference, and say that what we value has both high complexity, and low complexity "projections": we can make sense of this or that aspect, but never the ensemble as a whole.

In any case, making good judgements about the complexity of an explanation is part of being a good thinker. Let's leave aside *why*, for the moment, and see what it means for computerland.

2 Kolmogorov Complexity

If we are computers, living in a computer program, then Kolmogorov Complexity is the arbiter of good taste. It's the ultimate, algorithmic Occam's Razor.

In computerland, everything we know is the consequence of running a program. This means explanations are computer programs too, bits of mindware we run to efficiently describe what we see. If you adopt explanation E in computerland you load up a program, call it P, that can print out all the things the explanation predicts. For any phenomenon, there are many different programs P that might explain it, and some are more efficient than others—in particular, some are shorter than others, in terms of their source code.

KC says that the complexity of an explanation is the length of its shortest program. In computerland, if we want to know the complexity of E, we are going to have to run a separate "KC program" on it that tells us that length. The idea of a KC program is not, on the surface, crazy. It matches, for example, the intuition that figuring out a more efficient way to explain what we think is a good way to get closer to the truth.

As with non-computerland notions of simplicity, KC can be turned around to reflect not on explanations, but on the things they're trying to explain. In computerland, we might run the KC program to talk about how complex a society is, or a living organism, or the planet Earth, or, indeed, the Universe itself,

There's just one problem: the KC program does not exist. It's not that the KC program is a very complicated thing that we haven't written yet. The KC program does not exist in the same way that a square triangle does not exist, or the second even prime does not exist, meaning that to believe it exists is to believe a logical contradiction, and to tell someone to use it is tell them to square the circle. There is no program, and can never be. In computerland, it is a thought that can not be thought.

The simplest way to show this is by a reductio ad absurdum: we assume the program exists, and derive a contradiction. "Berry's Paradox", made famous by Bertrand Russell, does this using a simpler version of KC, call it KC'. KC' is a program that takes a number and tells you the smallest number of words it takes to name it. If you run KC' on "2", it gives the answer "1". This is because I can name the number 2 in one word ("two"). I could also have said "the first even prime" (four words), but "two" is more efficient.

Now consider the following sentence:

> "The smallest number that can not be named in less that fifty words."

If we have KC' to hand this is easy, if tedious, to solve. We go through all the numbers, starting with zero, until KC' gives the answer 50. It might take a long time, but that's OK—eventually (if only because the number of words we have is finite) we'll feed it some number, call it N, and it will reply "50".

Now we have a problem, because we just named N with less than fifty words. We've done it using the sentence "the smallest number that can not be named in fewer that fifty words", which contains thirteen. KC' has produced an incorrect answer or, to complete the reductio, we began by assuming there's a KC' program that gives correct answers, and derived the fact that it doesn't. The crucial step was that KC' could be used to talk about itself: the sentence above has an implicit reference to running KC'.

Here's an attempt at actually writing the KC program (rather than assuming it already exists):

1. Write down all the programs of length k.
2. Run each of these programs in turn.
3. If one of them spits out X, stop! The KC of X is k.
4. If you finished all of them, start again, but now try programs of length $k + 1$.

The reductio says it can't work, but where's the problem? The answer is that some of programs you consider may not actually finish. Once you hit one, the KC program hangs. You might think you can check ahead of time for these bombs, but this turns out itself to involve a contradiction, known as the Halting Problem. The problem of the Halting Problem is, effectively, a contradiction in the idea that you can see, in all cases, the consequences of an idea without actually thinking it.

This gives us an idea, though. Let's try the following, resource-limited KC program (RLKC):

1. Write down all the programs of length k.
2. Run them in turn, each for a maximum of ten minutes.
3. If one of them spits out X, stop! The KC of X is k.
4. If you finished all of them, start again, but now try programs of length $k + 1$.

If the KC program says simplicity is brevity, then RLKC says simplicity is brevity, as long as it doesn't get too deep. A good explanation is concise, but doesn't take too much time to unpack.

Given any X, RLKC will eventually return a number. In the worst case, it will skip over all the too-clever-by-half programs and say that the complexity of X is the length of the program "`Print X`". RLKC escapes the reductio because it will reject any program that includes a copy of itself. As soon as k gets large enough to include a bomb program, running RLKC will take at least ten minutes; then, if RLKC contemplates an explanation that includes RLKC, it will terminate consideration.

RLKC ideas play an interesting role in statistical inference, but at the cost of barring self-reference. A being equipped with only RLKC can use simplicity to decide between explanations, but can't consider explanations that themselves invoke simplicity. Among other things, this means it will be unable to judge this paper.

RLKC has more problems than an inability to philosophize. This is because explanations build on explanations. When I say "he missed the bus and walked home", what I mean is something like "he missed the bus (and take what that means in the simplest possible way consistent with events), and walked home (and take what that means in the simplest possible way consistent with events)." One way to reveal those hidden references is to imagine the response if someone

objected to your explanation in a way that suggested he hadn't taken "walking home" in what you thought was the simplest possible way. RLKC, however, will reject that explanation ("too deep for me!") and prefer instead "he missed the bus and walked home on his hands."

The KC program does not exist, but there are many fascinating things about what it would say if it did. Some of these fascinating things make it seems like you can (for example) come up with a program that approximates KC. And this might be enough.

For example, there is a deep relationship between uncertainty (not knowing what my friend is going to do) and complexity (how hard it is to talk about what he did in any particular case). In technical language, the entropy of a timeseries approximates the average value of its KC: the uncertainty I have about what my friend is going to do tomorrow is very close to the complexity of the (best) explanation of whatever he did in any particular case.

Since uncertainty (it turns out) is easy to measure,[2] it seems like we've found a loophole in the logical impossibility of the KC program. There's a snag: the error of the approximation to the average KC of what your friend does is *itself* uncomputable. It's equal, in fact, to the KC of the probability distribution over the different things he will do. That error is not only potentially arbitrarily large, it is also unknowable.

All of the apparent escape paths for the fact that the KC program doesn't exist have this property.[3] Not only can't you compute KC, but you can't kind of guess KC, and then fix it up in a way that lets you do anything you couldn't do before. Any scheme you cook up is going to be fragile and, of course, any scheme you cook up to handle that fragility will have the same problem. What Frank Ramsey said of Wittgenstein's *Tractatus* also applies to Kolmogorov Complexity: what you can't speak about you can't speak about—and you can't whistle it, either.

3 Taste is not universal

The KC program doesn't exist—and neither does anything that approximates it. However, things that sound a lot like it are everywhere.

This is because any compression algorithm contains an implicit answer to the question "how complex is X": it will find some Xs easier to compress than others. One way to compare the complexity of two explanations is to type them into Microsoft Word documents, compress them into ZIP files, and compare the resulting sizes.

What makes solutions like these different from KC is that a compression algorithm contains an implicit account of the kinds of things it is likely to encounter. Informally, it represents the objects using the patterns it expects ("X occurrences of pattern 1, Y occurrences of pattern 2," and so on), along with a count of exceptions. Things that match the algorithm's expectations will be easier for it to compress; what don't will look complex. Text compression, for example, might use the fact that some letter pairs ("th") are more common than others. Fed a text where that pattern is weak, but an unexpected one is strong (say, pairs separated by seven intervening letters), it will get the complexity wrong.[4]

Every compression algorithm tells you simplicity relative to a particular story about how things usually go. What constrains computers echoes in us: when I say "take what I mean by 'walked home' in the simplest possible way", what I often mean is "take what I mean by 'walked home' in the normal sense", *i.e.*, the way people in this town normally walk home.

The impossibility of the KC program throws us back on expectations and norms. It's helpful to think of it as an aspect of a Theory of Relativity for computer science, where "relativity" here actually does finally, annoyingly, mean "it's all relative" in the ultimate nightmare Grievance Studies sense. Relativity in physics means there's no preferred reference frame; the computer science version, the No Free Lunch Theorem of David Wolpert and Bill Macready, that there is no preferred learning algorithm. A parallel argument tells us that any compression algorithm, and thus any notion of simplicity, is no better or worse than how it matches the pattern of things it was raised on. This relativity of knowing means that there's no knowledge that does not take, as its ground, one's personal experiences, or those of the tribe, or the species.[5] Self-modifying tribes and species are no exception to this law, which prescribes humility: even to say we approach the KC program is to fall into contradiction.[6]

[2] See `http://santafe.edu/~simon/it.pdf` for a simple introduction to uncertainty, entropy, and estimation.

[3] If you think you've found one, feed it to itself.

[4] More sophisticated algorithms not only find patterns, but metapatterns—logics for how patterns compose. They're on dangerous ground, because a broad-enough logic will also be, secretly, self-referential. This means that you're actually trying to compute KC, which will have to fail, or, more likely, a somewhat opaque RLKC.

[5] Mature accounts of Occam's Razor understand this relativity; see, *e.g.*, David MacKay's *Information Theory, Inference and Learning Algorithms* (2003).

[6] See "The Paradox Manifesto", *Cambridge Literary Review*, Vol. 11, Michaelmas (2018).

4 Kolmogorov Complexity as physical law: an anthropic comedy

The KC program does not exist. That doesn't mean that Kolmogorov Complexity, the mathematical object, can't.

Let's short-circuit the philosophy of mathematics and say a mathematical object exists if it plays a causal role in the natural world. Then there are plenty of mathematical objects that exist. Odd numbers exist: they play a causal role in making it hard to split the check.[7] The electric field is a mathematical object (a vector field over three-dimensional Eucldiean space), but it also moves particles around. In that sense, "vector fields over three-dimensional Euclidean spaces" exist. This is true whether or not Maxwell's Equations are computable.

In computerland, however, we live in a computer simulation. This means that only computable objects exist for us, because what exists for us is (by hypothesis) the outcome of a computer simulation. The beings who programmed this simulation can make Maxwell's Equations causally real because there's a Maxwell's Equation program that runs them. They can't make Kolmogorov Complexity real, because the KC program does not exist.[8]

Imagine there was a cupboard, and you put something into the cupboard, and when you open the door again, it has a post-it note with the object's KC. All of a sudden, KC can play a causal role in the world. Computer scientists call this an oracle, and have a literature devoted to things like putting one cupboard inside another, and so on. Some people want to forbid the possibility of oracles, as they would be under what Scott Aaronson calls the Physical Church-Turing Thesis. This is a half-way house to computerland: it doesn't say we live in a simulation, only that we might as well be. Certainly, the cupboard's existence is inconsistent with computerland.

Let me propose that this cupboard actually exists. In this cupboard are all the universes. Every few trillion years, someone opens the cupboard, and chooses one of them with a probability inversely proportional to its KC. A Universe with smaller KC is more likely to be picked. Less picturesquely, you can imagine that a fundamental law of Universe creation has a prior that is a function of KC.

Most of the time, the Universes are extraordinarily simple—so simple, in fact, that nothing happens. What really matters, though, is what they look like conditional upon our being to observe them. What do *these* Universes look like?

First, like the others, they follow extraordinarily simple laws. KC doesn't care about resource limitations, so the Universes can be very very big. It also doesn't care about computation time, so even though the laws are very very simple, they might take a long time to understand. Call the description of the universe f; the way this description changes over time should follow a very simple law. One need not belabor the point that this first prediction is consistent with the bizarre fact that the laws of physics, though hard to master, are very simple to state.

Most physical theories distinguish laws from initial conditions. A second prediction, then, is that these universes would have the simplest possible initial conditions consistent with life.

This is a rather subtle point. A classical Universe that began in a particular initial configuration—this atom here, that one there—will tend to have very high KC. But if the initial conditions are specified, as they are in quantum theory, as a wavefunction over Hilbert Space, the complexity can decrease dramatically. What matters is not $\mathrm{KC}(x)$, the complexity of the particular configuration x, but $\mathrm{KC}(f)$, the complexity of the probability distribution $f(x)$ (or, rather, its underlying wavefunction). The simplest possible distribution[9] is the Gaussian, and if f is a Gaussian, it can be specified with just two numbers, the mean and the variance. We just have to make sure that the mean and variance are consistent with life.

The prediction that f is Gaussian is consistent with the best astronomical evidence we have, from WMAP, about the statistical properties of the Universe's initial conditions. Although they gave it their best shot, cosmologists were unable to find any deviation from this simplest possible specification.

When the initial conditions are simple to explain, *i.e.*, $\mathrm{KC}(f)$ is small, it becomes easy to talk about the expected complexity of the classical universe that we experience. This is because the average value of $\mathrm{KC}(x)$ becomes very close to the entropy, or uncertainty, of the initial conditions, $H(f)$. The fundamental law of algorithmic information theory (encountered, in passing, at the end of section two) tells us that

$$0 \leq \left(\sum_{x \in X} f(x) \mathrm{KC}(x) \right) - H(f) \leq \mathrm{KC}(f), \tag{1}$$

[7] Some hard-liners on cause require interventions (Judea Pearl's "do" operator), but there's no "do" to make odd numbers even. If this bothers you, say "explains why" instead.

[8] This is not quite true. Maxwell's Equations—at least the ones I learned—are defined over the reals, and the reals are not computable. If you're a hard-liner, as I learn from Joscha Bach, only computable numbers exist. Because the reals contain lots of uncomputable numbers, computerland is full of holes.

[9] "...with non-compact support". You can always show something has low KC if you find the program. Gauss found this one.

or, in words, when the distribution f is very simple, the average experienced KC is very close to the entropy of f. There's an overhead from the evolution laws, but by stipulation that's small.

At this point, something unexpected happens. We wanted a small KC, which led us to propose that f was a Gaussian. But while $\mathrm{KC}(f)$ is very small, $H(f)$ is very large. In fact, as E.T. Jaynes taught us, f is the distribution with *maximum* entropy, as large as it could possibly be given the constraints of mean and variance. This means we expect $\mathrm{KC}(x)$ to be very large. The universe follows very simple laws, and has very simple initial conditions, but in our experience, as classical beings, it's irreducibly complex. The equation on the t-shirt is simple, but the t-shirt is a mess.

Once we realize the problem is on two planes—the simplicity of f, the complexity of x—we learn a great deal. For example, it tells us that the mutual information between two parts of the universe is very close what you might call the mutual explainability.[10]

Formally, say we partition the Universe (its wavefunction) into two parts, A, and B. Consider a realization of A, a, and a realization of B, b. We can talk about the complexity of a, given knowledge of b, "how simple an explanation of a is, if it can take as input facts about b", written as $\mathrm{KC}(a|b)$. In general, $\mathrm{KC}(a|b)$ will be less than or equal to $\mathrm{KC}(a)$; in the worst case, if b is irrelevant to a, we can ignore it and go about explaining a on its own. The difference between these two terms is the mutual explainability. Going back to the distribution f, the average mutual explainability is

$$\sum_{a\in A; b\in B} f(a,b)\left(\mathrm{KC}(a) - \mathrm{KC}(a|b)\right). \tag{2}$$

If (and only if) f has low KC, as it is when pulled from the cupboard, then this is equal, up to some nuisance constants, to something called mutual information.[11] In contrast to explainability, mutual information is easy: it just tells you how much knowing that a reduces your surprise about whether b (or vice versa)—a generalized notion of how much A correlates with B. This *only* works if f has low KC. Otherwise, mutual explainability is replaced by (complicated) talk about coincidences baked in by initial conditions.

In short, things that are correlated with each other also tend to explain each other. Explanation here is to be taken in the deep, Kolmogorov sense: in these universes, if samples of a correlate with samples of b, then even though a, on its own, looks complicated, you can (on average) find a simpler explanation for it once you know b. This also goes the other way: if you see that a is easier to explain given b, you ought to expect the processes that generated them to be statistically correlated.

These facts are such a pervasive feature of life that it takes a while to realize they never had to be true in first place.[12] With a little thought, however, you can imagine universes where they aren't. They are not hostile to life but they are to explanations, and involve (as you would expect from the arguments here) strange and fine-tuned initial conditions. They are Robert Anton Wilson worlds of persistent, meaningless coincidences—where a toss of the *I Ching* ("Hexagram 53: Chien") predicts the front page of the *Daily Californian*, but helps it make no greater sense. "Prediction without explanation" is a challenge for machine learning, too, but our account here suggests it may only be a temporary one.

The idea that mutual explainability provides evidence against the physical Church-Turing Thesis is radical. The standard (in computerland, "normie") story of how we explain gestures to a background of expectations that come from biological evolution. Yet while evolution can account for some of our ability to explain what is correlated, it cannot account for the explainability itself. It is also limited in its ability to account for why we are able to make progress on problem types we never evolved to explain, such as the structure of the atom.

It also can't explain the nature of that progress, which is characterized by, among other things, the unreasonable effectiveness of mathematical reasoning. That includes not only that any particular piece of mathematics works, but also how mathematical simplicity has served as an anticipatory pointer to good explanations.[13] Mathematics is not the only language in which things become simple: the explainability of our world is also what underlies the sense-making possibilities of the novel and, I am told, Kant's first *Critique*.

In the end, of course, we may still be living in a finely-tuned computer simulation. That's one explanation for what's going on. The alternative has the advantage of simplicity.

[10]The following considers explainability in terms of the complexity of the explanation—and not, say, consistency with previously-known laws. If that bothers you, substitute "describe efficiently", or "model" instead.

[11]The equality holds when we know not only b, but also $K(b)$, *i.e.*, both b and how explainable it is. This contributes a small additional term, $\log n$, the length of b. See "Shannon Information and Kolmogorov Complexity," Peter Grünwal and Paul Vitányi (2008).

[12]To be clear, correlation implies explanation, not (sorry!) causation. We may need to explain by reference to a hidden variable.

[13]Mark Steiner's *The Applicability of Mathematics as a Philosophical Problem* (1988) covers this in a rigorous fashion.