The Semimeasure Property of Algorithmic Probability – "Feature" or "Bug"?

Douglas Campbell

Philosophy Programme, University of Canterbury, Christchurch 8041, New Zealand
douglas.campbell@canterbury.ac.nz

Abstract. An unknown process is generating a sequence of symbols, drawn from an alphabet, $\mathcal{A}$. Given an initial segment of the sequence, how can one predict the next symbol? Ray Solomonoff's theory of inductive reasoning rests on the idea that a useful estimate of a sequence's true probability of being outputted by the unknown process is provided by its *algorithmic probability* (its probability of being outputted by a species of probabilistic Turing machine). However algorithmic probability is a "semimeasure": i.e., the sum, over all $x \in \mathcal{A}$, of the conditional algorithmic probabilities of the next symbol being x, may be less than 1. Solomonoff thought that algorithmic probability must be normalized, to eradicate this semimeasure property, before it can yield acceptable probability estimates. This paper argues, to the contrary, that the semimeasure property contributes substantially, in its own right, to the power of an algorithmic-probability-based theory of induction, and that normalization is unnecessary.

Keywords: Algorithmic probability, sequence prediction, inductive reasoning, Solomonoff induction, Solomonoff normalization, semimeasure, convergence theorem.

1 Introduction

This paper is about whether a certain property of algorithmic probability (ALP) – namely, its so-called "semimeasure" property – should be regarded as a "bug" (i.e., as a source of theoretical weakness, that must be worked around and corrected for) or as a "feature" (i.e., as serving a useful or necessary function) within the context of an ALP-based theory of inductive reasoning. I will begin by describing ALP and its application to inductive inference. Next I will describe the semimeasure property of ALP, and explain why it is commonly considered to be a bug that must be eradicated and patched over with an *ad hoc* normalization procedure. Finally I will contend that this negative assessment of the semimeasure property's worth is incorrect. I will argue that the semimeasure property is properly seen as being a valuable and important feature of ALP, which makes a major contribution to the power, scope and elegance of an ALP-based theory of inductive reasoning. I will demonstrate that to normalize ALP is to pay a high price, in terms of lost theoretical elegance, in order to attain a result – the

D.L. Dowe (Ed.): Solomonoff Festschrift, LNAI 7070, pp. 79–90, 2013.

elimination of the semimeasure property – that is wholly undesirable in the first place. It is to "cut off the nose of ALP to spite its face", so to speak.

2 Notation

The symbol, $\wedge$, denotes the empty string, "". $|x|$ denotes the length, in symbols, of the string, x. E.g., $|\wedge| = 0$ and $|ABC| = 3$. xy denotes the concatenation of strings x and y. $\wedge x = x = x\wedge$, and $|xy| = |x| + |y|$. The *prefixes* of a string include all initial segments of the string. Every string is a prefix of itself, $\wedge$ is a prefix of every string, and x is a prefix of xy.

3 Algorithmic Probability (ALP)

The concept of ALP involves a type of computing device that I shall here call a *Solomonoff machine*. Such a machine is a finite state automaton equipped with an indefinitely expandable internal working memory, which accepts a sequence of randomly generated binary digits as input, and which emits another sequence of binary digits as output. At each step, the machine either might or might not accept a randomly generated digit of input, and might or might not emit a digit of output. Over the full course of its operation it might accept either a finite, or an infinite, number of input digits, and it might emit either a finite, or an infinite, number of output digits. It has no capacity to retract or modify its output, so each digit of output is "set in stone" the moment it is produced. The indeterministic process that generates its input is "fair", 0s and 1s being equiprobable.

A Solomonoff machine can be concretely realized as a probabilistic "monotonic" Turing machine with three tapes, these being: (i) a two-way, read-only, initially blank work tape; (ii) a one-way, read-only input tape pre-inscribed with an ongoing randomly generated binary sequence; and (iii) a one-way, write-only, unidirectionally accessible, initially blank output tape with the alphabet, $\{0, 1\}$.

Let Sx denote a particular Solomonoff machine (having some particular state-transition table).

The string, y, *encodes* the string, z, on Sx, iff any input to Sx prefixed by y will result in Sx's output being prefixed by z. (So, for example, if Sa's output must start with 11 provided its input starts with 010, then 010 encodes 11 on Sa.)[1]

Let Fx be the function computed by Sx. $Fx(y) = z$ iff z is the longest string encoded by y on Sx. (So, for example, if 010 encodes 11 on Sa, but if it encodes neither 110 or 111 on Sa, then $Fa(010) = 11$.)

A given Solomonoff machine will usually produce any one of a variety of different outputs with different probabilities, its output depending on which particular random input it is fed with. Let Px denote the probability distribution, over binary output strings, associated with Sx. $Px(z) = q$ just in case the probability

[1] This is similar to the notion of Educated Turing Machine in [1, sec. 4][2, sec. 2.3].

of Sx's output being prefixed by the binary string, z, is q. $Px(w|z)$ denotes the *conditional probability* of Sx's next symbols of output constituting the binary string w, given that Sx's output to date is z. $Px(w|z) = Px(zw)/Px(z)$.

The string, p, is a *program* that causes Sx to *simulate* Sy, iff, for any string z, $Fx(pz) = Fy(z)$. In other words, the effect of Sx receiving an input prefixed by a program that causes Sx to simulate Sy is to cause Sx to “change personalities” (so to speak), by thereafter exhibiting input-output behavior indistinguishable from that of Sy. ($\wedge$ is a program that causes every Solomonoff machine to simulate itself.)

A *universal Solomonoff machine* is a Solomonoff machine that can be programmed to simulate any Solomonoff machine. That is, if Su is universal, then for any Solomonoff machine, Sx, there is a program that causes Su to simulate Sx.

Let the *reference machine*, Sm, be some particular universal Solomonoff machine that has been selected, by us, to serve as our benchmark for measuring the ALP of strings. The ALP of any string, x, is simply $Pm(x)$. That is, a string's ALP is the probability of our reference machine's output being prefixed by the string. ALP is obviously *machine-dependent*, in the sense that the ALP of a string will tend to vary depending on which particular universal Solomonoff machine we choose to be our reference machine.

4 The Semimeasure Property of ALP

A probability distribution, ρ, over binary sequences is a *measure* if and only if $\rho(x) = \rho(x0) + \rho(x1)$ for any string, x. In other words, it is a measure if the conditional probabilities it assigns to the next symbol after x being a 0 and to the next symbol after x being a 1 must always, for any x, sum to unity. On the other hand, ρ is a *semimeasure* if and only if there is some string, x, such that $\rho(x) > \rho(x0) + \rho(x1)$. For example, suppose that $\rho(010) = 0.6$, while $\rho(0100) = 0.3$ and $\rho(0101) = 0.1$. This being so, not all the probability assigned by ρ to the sequence “010” is split between and inherited by the two, longer strings “0100” and “0101”. Some of the shorter string's probability (0.2 of the 0.6) instead “goes missing”, so to speak. This makes ρ a semimeasure.

Recall that $Pm(x)$ is the probability of Sm's output being prefixed by x. Having outputted x, Sm must next do one of three different things: (i) it might output another 0; (ii) it might output another 1; or (iii) it might stop outputting 0s and 1s once and for all as a result of either having halted or having gone into an infinite, unproductive loop. Sm always, for any x, has a non-zero probability of doing the last of these things (there being a non-zero probability that Sm's random input will start with a program that causes it to simulate a second Solomonoff machine that will always, regardless of its input, output x and then halt). It follows that $Pm(x) > Pm(x0) + Pm(x1)$, which makes Pm a semimeasure, not a measure.

5 ALP's Application to Induction, and the Semimeasure Problem

ALP was discovered by Ray Solomonoff, who used it as the central ingredient of a theory of inductive reasoning [3, 4]. The theory concerns a method for accomplishing a certain type of sequence prediction task. By way of illustrating the task, let's imagine that a black box has fallen to Earth from a place unknown. Attached to the black box's exterior is a symbol-stamping mechanism, through which is threaded an initially blank tape. Casual inspection of the mechanism reveals that it is capable of stamping only two types of symbols onto the tape – 0s and 1s – and that each symbol will be stamped on the tape to the immediate right of its predecessor. Both the ordering of these symbols, and the timing of each symbol's delivery, are under the control of a process hidden within the black box. We have little or no idea what this process might be, but the gradually accumulating sequence of symbols it produces is exposed to our view. The black box receives no input. Let the *black box task* be the task of making a probabilistic prediction about the black box's next symbol of output, based on its observed output-to-date.

Two types of method for accomplishing the black box task may be distinguished. A *three-way method* is a method which accepts any given binary string of the black box's output-to-date, and then assigns conditional probabilities to each of three distinct possibilities, these being: (i) the next symbol will be a 0; (ii) the next symbol will be a 1; and (iii) the black box will never output another 0 or 1 again, and so the next symbol on the black box's output tape (together with all subsequent symbols) will default to $\_$ (where $\_$ represents a blank). A *two-way method*, on the other hand, assigns conditional probabilities to only two possibilities: (i) the next symbol will be a 0; and (ii) the next symbol will be a 1. A two-way method should obviously be used only if the possibility of the black box's output terminating can be dismissed out of hand. Such might be the case because one knows from the outset that the process operating in the box will keep producing binary digits forever (e.g., perhaps one has been told as much by a trustworthy source who has looked into the box).

Let $\mu(x)$ denote the true, objective probability of the black box's output being prefixed by the binary string, x, and let $\mu(y|x)$ denote the conditional probability of the black box's next symbols of output comprising the string, y, given that its output-to-date is x. $\mu(y|x) = \mu(xy)/\mu(x)$. If the process in the black box is somehow guaranteed by facts about its constitution to keep producing 0s and 1s forever, then μ will be a measure. Otherwise, if there is a non-zero objective probability of the black box's output terminating at some point, then μ will be a semimeasure. If the process in the black box is deterministic then, for any string x, either $\mu(x) = 0$ or $\mu(x) = 1$. If it is indeterministic then there will be some strings x such that $0 < \mu(x) < 1$.

The essential idea behind Solomonoff's theory of induction is that we should predict the output of the black box (or equivalent symbol source) by assuming it has the same output producing dispositions as our reference machine. In its simplest form, the idea is that we should use $Pm(x)$ as an estimate of $\mu(x)$ (or, equivalently, $Pm(y|x)$ as an estimate of $\mu(y|x)$). So, for instance, if the reference machine would, if its output-to-date were “0011”, have a probability of 0.3 of next outputting a 0, and if the black box’s output-to-date is “0011”, then, so the idea goes, we should assign a probability of 0.3 to the black box’s next symbol of output being a 0.[2]

Solomonoff focused specifically on using ALP to develop a two-way method for predicting the extension of a binary string. He seems not to have considered using it to construct a three-way method. Hence, at least as far as Solomonoff was concerned, μ must be a measure, and there are only two things that the black box might legitimately do next – output a 0, or output a 1. (Indeed, Li and Vitanyi report that Solomonoff, “viewed the notion of measure as sacrosanct” [12, p. 280].) But, as we have seen, Pm is a semimeasure, and there are, at any point in time, *three* things the reference machine might do next – output a 0, output a 1, or stop producing binary output. Hence a problem arises (the “semimeasure problem”, as I will call it). Since a two-way method must divide conditional probability only between the possibilities of the next symbol being a 0 or of it being a 1, the conditional probabilities it assigns to these two possibilities should sum to unity. However, because the reference machine divides probability between *three* future possibilities, not just two, the conditional probabilities it apportions to 0 and to 1 may (and in fact, always will) sum to a value less than unity.

Solomonoff addressed this problem by describing a normalization operation that converts the semimeasure, Pm, into a corresponding measure, Pm' [4, 13]. This operation works by, in effect, taking the probability of the reference machine receiving a random input that will cause it to terminate its output after outputting the binary string, x, and then redistributing this probability back over all random inputs to the reference machine that will cause it to output at least one more 0 or 1 after x. This is done recursively, for progressively longer strings, x. That is:

$$
\begin{aligned}
Pm'(\wedge) &= 1 \\
Pm'(x0) &= Pm'(x)\frac{Pm(x0)}{Pm(x0)+Pm(x1)} \\
Pm'(x1) &= Pm'(x)\frac{Pm(x1)}{Pm(x0)+Pm(x1)}
\end{aligned}
$$

[2] Solomonoff’s theory of induction is to be contrasted with the closely related *Minimum Message Length* (MML) approach of Wallace and Boulton [5–10]. For a comparison of the two approaches, see [1] and [2, p. 404]. Some proponents of MML argue that Solomonoff’s theory isn’t really a theory of “induction” at all (see, for instance, [2, p. 405–407] and [11, p. 930–931]), with one of the thoughts being that, whereas genuine induction involves reasoning from a body of observations to a general hypothesis, Solomonoff’s procedure yields no such general hypothesis, and instead yields only predictions about future observations (of upcoming 0s and 1s). I contend that Solomonoff’s procedure is genuinely inductive, in at least the sense that it yields predictions about the future behaviour of the black box *that are not deductively implied* by anything that is known about the black box or its output to date. However pursuing this issue would take me far from the topic of this paper.

Solomonoff's considered proposal was that we should predict the black box's output by using the measure, Pm', rather than the un-normalized semimeasure, Pm, as an estimate of μ [4].

6 "Bug" or "Feature"?

Solomonoff himself made mention of a distinction between the "features" and "bugs" of ALP while defending his theory of induction from a pair of criticisms [14]. The first criticism concerns the fact that the values of $Pm(x)$ and $Pm'(x)$ are uncomputable, and hence largely unknowable in practice. The second concerns the fact that these values are also radically dependent on our particular choice of reference machine, and to this extent arbitrary and subjective. Solomonoff responded to these criticisms by maintaining that both the uncomputability and the machine-dependence of ALP are to be properly seen as playing useful, and indeed indispensible, roles in his theory of induction, rather than as being sources of theoretical weakness. Specifically, he held that uncomputability is simply a necessary flipside of completeness: that ALP is uncomputable precisely because it can be used to detect and extrapolate *any* computable regularity or pattern in a sequence of data [15]. In a similar vein, he held that machine dependence is vital in enabling us to factor in whatever prior information we might possess about the symbol source. Our prior knowledge about the symbol source should, Solomonoff maintained, be directly reflected in our particular choice of reference machine [16]. He summed up the situation by saying that both ALP's uncomputability and its machine-dependence count as "necessary features" of his theory, not as "bugs" [14].

Following Solomonoff's lead, let's count among the "features" of ALP any of its properties that should be celebrated by a proponent of an ALP-based theory of induction for the valuable role they play in the theory, and let's count among its "bugs" those of its properties (if any) that are to be regretted for the problems and weaknesses they introduce. When Solomonoff held that uncomputability and machine-dependence are features, not bugs, of ALP, he was charging critics of his theory of induction with overlooking ways in which these properties can be turned to the theory's advantage, by being made to serve useful or necessary functions within it. It is clear that Solomonoff himself regarded the semimeasure property as a genuine "bug" in the idea that we should use ALP to predict the output of the symbol source, for – as just explained – he used a normalization procedure to eradicate it, and did not attempt to show that it can be exploited to play a useful role in the theory. I will now try to show that it is instead properly regarded as being a very valuable "feature".

7 Another Way of Tackling the Semimeasure Problem

We've seen that the semimeasure problem arises because, whereas the black box must do one of only *two* things next – output a 0, or output a 1 – the reference machine can instead do either one of *three* things next – output a 0, output a 1, or terminate its output. However this disparity between the ranges of behaviours the two devices can exhibit arises only when it is stipulated from the outset that the black box's output can't terminate. When a three-way method is used to predict the black box's output, no such stipulation is in force. Hence, provided we use ALP to construct a three-way method, rather than a two-way method, then each and every possible behaviour of the reference machine corresponds directly to a possible behaviour of the black box, and *vice versa*.

The following proposal for resolving the semimeasure problem therefore suggests itself: whereas Solomonoff used ALP to construct a two-way method for predicting the extension of a binary sequence, we will instead use it to construct a three-way method. In other words, we will include the possibility of the black box's output terminating among the set of alternative outcomes to which a probability must be assigned. The probability we assign to the black box's output terminating after it has outputted the string, x, will be identical to the probability of the reference machine's output terminating after it has outputted x. Under this proposal, ALP's semimeasure property doesn't merely cease to be a "bug" in an ALP-based theory of induction, but instead acquires the status of being a useful and necessary "feature", for in order for a three-way method to assign a certain quantity of probability to the possibility that the sequence has terminated, it must leave the selfsame quantity of probability unassigned either to the possibility that the next symbol will be a 0 or to the possibility that it will be a 1. Hence the probability distribution that such a method is based on must be a semimeasure, and cannot be a measure.

We now have two proposals on the table, which I will call *Solomonoff's proposal* (it being the proposal that Solomonoff championed) and the *new proposal* respectively. According to Solomonoff's proposal, the proper goal of an ALP-based theory of induction is to construct a maximally reliable two-way method for predicting the continuation of a binary series, and the normalized measure, $Pm'(x)$, should be used as an estimate of $\mu(x)$. According to the new proposal, on the other hand, ALP is best used to construct a three-way method for making such predictions, and the unnormalized semimeasure, $Pm(x)$, should be used as an estimate of $\mu(x)$. Both proposals circumvent the semimeasure problem, and are on an equal footing in this respect, but I will now offer reasons to believe that the new proposal is nevertheless superior to Solomonoff's.

The first and most important reason concerns Solomonoff's own grounds for thinking that ALP-based predictions about the black box's output are likely to be any good. I will argue that these grounds offer stronger support to the new proposal than they do to Solomonoff's own proposal.

The following concepts and notation will be useful. Let a *padded string* be a binary string with a $\_$ appended to its rightmost end. (E.g., "0110$_$" is a padded string.) Let $\rho(x\_)$ denote the probability assigned by ρ to the possibility that the binary string, x, won't be followed by any more 0s or 1s. That is, $\rho(x\_) = \rho(x) - \rho(x0) - \rho(x1)$. Notice that if ρ assigns a non-zero probability to any padded string, then ρ is a semimeasure. Let's say that the distribution, ρ *dominates* the distribution, ν, iff there is some non-zero probability, p, such that, for any binary or padded string x, $\rho(x) \geq p\nu(x)$. So, for example, if, for any binary or padded x, the probability assigned by ρ to x never undershoots the probability assigned by ν to x by more than a multiplicative factor of, say, 0.3, then ρ dominates ν (to within a factor of 0.3).

Now, let's suppose that we use a distribution, ρ as an estimate of μ. It can be shown [4] that, on assumption that ρ dominates μ, then, as time goes by and as the black box's output-to-date grows in length, the conditional probabilities that we assign to the next symbol by using ρ will rapidly converge to match the true, objective probabilities that are assigned to this symbol by μ. That is, if $o_{1...n}$ denotes the black box's first n symbols of output and o_n denotes its n^{th} symbol of output, then $\lim_{n\to\infty}[\rho(o_n|o_{1...n-1}) - \mu(o_n|o_{1...n-1})] = 0$.

This is encouraging, for it means that our estimate of μ will yield good predictions in the long run provided that it dominates μ. But how can we arrange for our estimate of μ to dominate μ when – ignorant as we are about what is in the black box – we know little or nothing about the nature of μ itself? Solomonoff's answer is that we can maximize our chances of "catching" μ within the set of distributions that are dominated by our estimate simply by casting our net very widely indeed. A probability distribution, ρ is *computable* iff there is a classical Turing machine that will, when given a binary string, x, as input, output an encoding of $\rho(x)$. Solomonoff showed that Pm' is "universal" in the sense that it dominates *every computable measure* [4]. Hence Pm' will dominate μ if μ is a computable measure.

That's the good news. The bad news is that if μ is a semimeasure – which is to say, if the process in the black box has the capacity to produce a terminating output – then Solomonoff's convergence result doesn't provide us with any assurance that Pm' will yield accurate probabilistic predictions in the long run. This limitation of Pm' is unsurprising, for, after all, Pm' was designed by Solomonoff to provide us with a two-way method for making predictions, and as such it is to be used only when it is known that the black box's output *won't terminate*. But it is still a very serious limitation, for, after all, it is perfectly possible that the process in the black box will stop outputting 0s and 1s at some point.[3] Ideally, we would like our estimate of μ to yield accurate predictions irrespective of what is in the black box, and irrespective of the nature of μ. The more distributions that are dominated by our estimate of μ, the smaller the risk of μ escaping domination by it, and so the greater the chances that the estimate will lead us to the true probabilities [17, p. 28]. In order for the estimate to be able to lead us to the true probabilities even if the black box can produce

[3] For example, C.S. Wallace [2, p. 407] imagines a process that examines all the stable isotopes of the chemical elements, one by one, in order of their atomic weight, outputting extensive data about their physical, chemical and spectroscopic properties as it goes. After examining lead-208 the process will stop, lead-208 being the last stable isotope, and so the sequence of data it is producing will terminate at this point. A predictor who has observed a sufficiently long initial segment of this data sequence should ideally be able to predict both that the sequence will eventually terminate, and the point at which it will do so.

a terminating output, we need it to dominate, not just distributions that are measures, but also distributions that are semimeasures.

Is there a distribution we might use as our estimate of μ, which dominates, not just all computable measures, but also all computable semimeasures? Indeed, there is, and it is none other than the original, unnormalized version of ALP, Pm. As a first step to understanding why Pm dominates such a large class of distributions, it will help to introduce the notion of a *Solomonoff distribution*, this simply being a distribution that is associated with some particular (universal or non-universal) Solomonoff machine. That is, ρ is a Solomonoff distribution iff there is some Solomonoff machine, Sy, such that, for all binary strings x, $\rho(x) = Py(x)$. Some Solomonoff distributions are measures, while others are semimeasures. The Solomonoff distribution, Px, is a measure iff, regardless of which randomly generated input the corresponding Solomonoff machine, Sx, is supplied with, its binary output will never stop. On the other hand, if there is at least one possible input to Sx that will result in its output terminating at some point, then Px is a semimeasure.

It is easily shown that Pm dominates all Solomonoff distributions, including all those that are measures and all those that are semimeasures. To see this, suppose ρ is some Solomonoff distribution. This being so, there will be some Solomonoff machine, St, whose probability of producing an output prefixed by x is $\rho(x)$. Since our reference machine, Sm, is universal, there will be some program, g, that will cause it to simulate St. The probability of g occurring as a prefix of Sm's randomly generated input is simply $1/2^{|g|}$. If g does occur as a prefix of Sm's input, then, when Sm has read in g, it will begin simulating St, and from this point forward it will exhibit output-producing propensities indistinguishable from those of St. Thus Sm has a probability of at least $1/2^{|g|}$ of simulating St, and if it does simulate St then it will, like St, have a probability of $\rho(x)$ of producing an output prefixed by x. This means that Sm's own probability of producing an output prefixed by x must be at least $1/2^{|g|}\rho(x)$, from which it follows that Pm dominates ρ (to within a factor of $1/2^{|g|}$). As for ρ, so for any Solomonoff distribution, whether it be a measure or a semimeasure.

Every computable distribution – whether it be a measure or a semimeasure – is a Solomonoff distribution. In order to see this, recall that if ρ is computable then there is a classical Turing machine that, when given a binary string x as input, will produce an encoding of $\rho(x)$ as output. Given this classical Turing machine, we can easily engineer a Solomonoff machine, Sh, that simulates the classical Turing machine in order to determine, for any binary string x, the value of $\rho(x)$, and which then outputs a sequence prefixed by x with a probability of $\rho(x)$, while using the randomly generated sequence of 0s and 1s on its own input tape as a source of indeterminism. Since Sh's design ensures that $Sh(x) = \rho(x)$ for all binary x, ρ is a Solomonoff distribution.

Since Pm dominates all Solomonoff distributions, and since all computable distributions are Solomonoff distributions, Pm dominates all computable distributions, including not just all the computable measures that are dominated by Pm', but also all computable semimeasures. This being so, a three-way inductive reasoning method based on Pm is inherently less risky than a two-way method based on Pm'. The conditions that μ must satisfy in order for the former method to be guaranteed to yield accurate probabilistic predictions in the long run are considerably weaker and less demanding than those it must satisfy in order for the latter method to yield the same guarantee. The point might be put by saying that Pm is "more universal" than Pm', in the sense that Pm dominates a much larger class of computable distributions than Pm' does.

The second reason for preferring the new proposal to Solomonoff's proposal is pragmatic, and concerns the practical utility of the inductive methods they prescribe. We can safely use Solomonoff's two-way method only when we can be certain, ahead of time, that the process we are predicting will keep outputting symbols forever. It is, however, surely impossible to find any real-life example of a process that satisfies this condition. Science teaches us that the universe we inhabit is governed by the second law of thermodynamics, and that it is fated, if not to a Big Crunch, then to heat-death. Hence, far from it being the case that we can ever be *perfectly certain* that a symbol source we are dealing with will keep producing output for eternity, the smart money will always be on its output eventually terminating. Our background knowledge about our world may provide us with but little information about the likely symbol-producing propensities of the black box, but it does at least tell us that terminations of output are probably in the offing. This being so, we must, when we are doing induction in the real world, use a method that will yield acceptable results if μ is a semimeasure. A three-way method based on Pm satisfies this requirement, while a two-way method based on Pm' does not.

The third reason has to do with the comparative simplicity and elegance of the two proposals. The fundamental idea behind an ALP-based theory of induction is that an hypothesis that attributes certain output-producing dispositions to a symbol source can be represented by a program that causes the reference machine to itself manifest these selfsame output-producing dispositions. Some programs, of course, cause the reference machine to have a non-zero probability of producing a terminating output. Solomonoff thought that such terminating programs "do not result in useful output" [13, p. 567]. But if we take to its natural conclusion the idea that programs represent hypotheses about the output-producing dispositions of the symbol source, then why should we not hold instead that such a program, which causes the reference machine to have a certain chance of producing an output that ends, represents a hypothesis that says the symbol source has this same chance of producing an output that ends? Why not indeed! There is no principled reason not to, and if we do then the resulting theory of induction is both more elegant, in view of the fact that it doesn't arbitrarily treat terminations of output differently than 0s and 1s, and more predictively powerful, since it yields a three-way method that can cope with terminations of output, rather than a two-way method that can't. On the other hand, if, like Solomonoff, we treat the reference machine's terminating outputs as being predictively meaningless, then we are left to confront the semimeasure problem, and must wheel in a normalization operation to surmount it. The inclusion of a normalization operation further complicates the theory and detracts from its elegance.

The final reason I offer for thinking that the new proposal should be preferred over Solomonoff's proposal concerns certain technical objections to the normalization operation that the new proposal avoids by simply dispensing with normalization altogether. These include, for instance, the objection that there are several rival methods of normalizing, each yielding measures with different properties, and no very compelling reason to choose one over another [12, p. 281]; the objection (due to Robert M. Solovay) that every choice of normalization operation has an unboundedly large impact on the relative probabilities assigned to some particular sequence [12, p. 301] (but c.f. [18]); and the objection that, while Pm is at least lower semicomputable, Pm' is not even computable in this restricted sense.

To conclude, it is my contention that ALP's property of being a semimeasure appears to be a "bug" in an ALP-based theory of induction only if one insists on trying to whack the round peg of ALP into the square hole of a two-way method for predicting a black box's output. The semimeasure problem evaporates entirely if one accepts terminations of output as being events worthy of prediction, and therefore uses ALP to construct a three-way method for making predictions.

References

1. Wallace, C., Dowe, D.: Minimum message length and Kolmogorov complexity. Computer Journal 42, 270–283 (1999)
2. Wallace, C.: Statistical and Inductive Inference by Minimum Message Length. Information Science and Statistics. Springer (2005)
3. Solomonoff, R.: A formal theory of inductive inference: Parts 1 and 2. Information and Control 7, 1–22, 224–254 (1964)
4. Solomonoff, R.: Complexity-based induction systems: Comparisons and convergence theorems. IEEE Trans. Infor. Theory IT-24, 422–432 (1978)
5. Wallace, C., Boulton, D.: An information measure for classification. Computer Journal 11(2), 185–194 (1968)
6. Boulton, D., Wallace, C.: The information content of a multistate distribution. J. Theoret. Biol. 23, 269–278 (1969)
7. Boulton, D., Wallace, C.: A program for numerical classification. The Computer Journal 13(1), 63–69 (1970)
8. Boulton, D., Wallace, C.: An information measure for hierarchic classification. The Computer Journal 16(3), 254–261 (1973)
9. Wallace, C., Boulton, D.: An invariant Bayes method for point estimation. Classification Society Bull. 3(3), 11–34 (1975)
10. Boulton, D., Wallace, C.: An information measure for single-link classification. The Computer Journal 18(3), 236–238 (1975)
11. Dowe, D.L.: MML, hybrid Bayesian network graphical models, statistical consistency, invariance and uniqueness. In: Bandyopadhyay, P., Forster, M. (eds.) Philosophy of Statistics. Handbook of the Philosophy of Science, vol. 7, Elsevier (2011)
12. Li, M., Vitányi, P.: An introduction to Kolmogorov complexity and its applications, 2nd edn. Springer (1997)

13. Solomonoff, R.: Three kinds of probabilistic induction: universal distributions and convergence theorems. The Computer Journal: Christopher Stewart Wallace (1933-2004); Memorial Special Issue 51, 566–570 (2008)
14. Solomonoff, R.: The Kolmogorov lecture: The universal distribution and machine learning. Computer Journal 46, 598–601 (2003)
15. Solomonoff, R.: The discovery of algorithmic probability. Journal of Computer and System Sciences 55, 73–88 (1997)
16. Solomonoff, R.: Does algorithmic probability solve the problem of induction? Technical Report (1997)
17. Hutter, M.: On generalized computable universal priors and their convergence, IDSIA-05-05 (2005)
18. Solomonoff, R.: The probability of "undefined" (non-converging) output in generating the universal probability distribution. Inf. Process. Lett. 106, 238–240 (2008)