The Probability of “Undefined” (Non–Converging) Output in Generating the Universal Probability Distribution

Ray J. Solomonoff

Visiting Professor, Computer Learning Research Centre
Royal Holloway, University of London

IDSIA, Galleria 2, CH–6928 Manno–Lugano, Switzerland
rjsolo@ieee.org http://world.std.com/~rjs/pubs.html

Abstract

In order to generate a universal probability distribution to extrapolate a binary string x of length i, we feed random bits into a universal device, M. When we find an input string that gives an output matching x, we continue the successful input with random bits until M produces a zero or one as output. The relative probabilities of these two continuations can give a normalized prediction for the probability of the symbol following x. There is, however, a probability, $P_{i+1}(u)$ that the continued random input string will not generate any output for the $i+1^{th}$ symbol.

We will show

$$\underset{\mu}{E}\sum_{i=1}^{n} P_i(u) \leq k_\mu \ln 2$$

Here $\underset{\mu}{E}$ is the expected value with respect to μ, the probability distribution that created x.

k_μ is the Kolmogorov complexity of μ with respect to M.

n is any positive integer.

Usually we don’t know μ and so we don’t know k_μ. However, if μ is the uniform distribution, we can usually find a good upper bound for k_μ.

Introduction:

A universal distribution on binary strings, x can be defined using a universal Turing machine with unidirectional input tape, unidirectional output tape and a bidirectional work tape. If we feed random bits into this machine, there is some probability $P(x)$, that M will print an output beginning with x. $P(x)$ is the universal probability of x with respect to M. This probability, however, is not a *measure*, it is a *semimeasure*, and many inputs that produce x, will never print anything after that, but will continue to compute forever. We will investigate the probability of this occurring.

Section I

Suppose we have an ensemble of strings produced by a probabilistic source, μ. We do not know μ, but we are given x_n, the first n bits of a string produced by it. We can use the universal distribution to estimate the probabilities of the next bit of x_n being zero or one.

We use the universal Turing machine M, to make these estimates by feeding it random strings. It sometimes prints x_n, then after printing x_n, it will with probability $P_n(0)$, print a zero, with probability $P_n(1)$ it will print a 1, and with probability $P_n(u)$ it will never print again — so the next symbol is "undefined".

We will show that

$$\underset{\mu}{E} \sum_{i=1}^{n} P_i(u) \leq k_\mu \ln 2 \tag{1}$$

$\underset{\mu}{E}$ is the expected value with respect to μ, the generator of x_n.

k_μ is the length of the shortest program with which M can describe μ.

k_μ is a kind of "Kolmogorov Complexity" of μ with respect to M so that $k_\mu \ln 2 = \ln 2^{k_\mu}$ may be regarded as "the natural complexity of μ with respect to M. The appendix has more details on the relation of μ to k_μ.

To prove 1: Consider the semimeasure $P(x_n)$. $P(x_n)$ is the probability with respect to the Universal Machine M, of an output string beginning with x_n, when M has random input.

Let x_n^i be the i^{th} bit of x_n, so $x_n = x_n^1 \, x_n^2 \cdots x_n^n$.

Let $P_i(x_n^i)$ be the probability of x_n^i, given the previous $i-1$ bits of x_n.

Then $P(x_n) = \prod_{i=1}^{n} P_i(x_n^i)$ will be the unnormalized probability (semimeasure) assigned to x_n by M. The normalization factor for the i^{th} factor will be $\frac{1}{P_i(0)+P_i(1)} = \frac{1}{1-P_i(u)}$, since $P_i(0) + P_i(1) + P_i(u) = 1$.

The normalized form of $P(x_n)$ becomes

$$P'(x_n) = P(x_n) \prod_{i=1}^{n} \frac{1}{1 - P_i(u)} \tag{2}$$

Willis and Levin have shown (Sol78 p. 424; Theorem 2.) that

$$\frac{P(x_n)}{\mu(x_n)} \geq 2^{-k_\mu} \tag{3}$$

$\mu(x_n)$ is the probability assigned to x_n by the probabilistic source, μ.

Since both μ and P' are normalized probability distributions, $\underset{\mu}{E} \ln \frac{P'(x_n)}{\mu(x_n)}$ is the Kullback-Leibler distance between $\mu(x_n)$ and $P'_M(x_n)$. This distance [1] must be ≤ 0, so

$$\underset{\mu}{E} \ln \frac{P'(x_n)}{\mu(x_n)} \leq 0 \tag{4}$$

Substituting P' from (2) into (4) gives

$$\underset{\mu}{E} \ln \frac{P(x_n)}{\mu(x_n)} + \underset{\mu}{E} \sum_{i=1}^{n} \ln \frac{1}{1 - P_i(u)} \leq 0 \tag{5}$$

From (3) we know that

$$\underset{\mu}{E} \ln \frac{P(x)}{\mu(x)} \geq -k_\mu \ln 2 \tag{6}$$

and hence

$$-\underset{\mu}{E} \ln \frac{P(x)}{\mu(x)} \leq k_\mu \ln 2 \tag{7}$$

Adding inequalities (5) and (7) we get

$$\underset{\mu}{E} \sum_{i=1}^{n} - \ln(1 - P_i(u)) \leq k_\mu \ln 2 \tag{8}$$

Then, since $-\ln(1-z) = z + \frac{z^2}{2} + \frac{z^3}{3} \cdots$, we have

$$\underset{\mu}{E} \sum_{i=1}^{n} \sum_{j=1}^{\infty} \frac{(P_i(u))^j}{j} \leq k_\mu \ln 2 \tag{9}$$

Since all of the terms of (9) are ≥ 0,

$$\underset{\mu}{E} \sum_{i=1}^{n} P_i(u) \leq k_\mu \ln 2 \tag{10}$$

Which was to be proved.

[1]That the $K - L$ distance is always ≤ 0 was shown by Gibbs about a century ago, and more recently by Kullback and Leibler. (CT91) pp. 18-26, gives a very readable discussion and proof. It may also be shown using Lagrange multipliers, as in (Sol78) Theorem 4, Lemma 2

Section 2: Some Implications of These Results

(8) tells us that the expected value of the log of the normalization factor is $\leq k_\mu$. This is the strongest result. (8) implies (10), but (10) does not imply (8).

(10) assures us that $P_i(u)$ converges more rapidly than $\frac{1}{i}$. We do *not* suggest that $P_i(u) < c/i$ for some finite c, but rather that the partial sums for $P_i(u)$ are less than those of $\frac{1}{i}$ for large enough i.

It is notable that the forgoing results are valid when M has an arbitrarily large alphabet, and may or may not have a "stop" state.

How does this relate to Solovay's result of Sept. 12, 1989 (LV97 p. 301)? Solovay showed that there were an infinite number of cases in which $P(x)/(P(x0)+P(x1))$ was very large — So $P(u)$ was close to 1. However these events occur *very infrequently* and the theorem says that for a long sequence they don't have much weight.

The results of the present paper are for upper bounds on $P_i(u)$. For discussion of lower bounds see (Hu05) Problem 2.7, p.62.

When a universal monotone machine is used for induction, we usually don't know μ and so we don't know k_μ. However, if we are simply interested in $P_i(u)$ for a particular universal machine, M, we note that our previous results are for *any* normalized distribution, μ. If we set μ to be the uniform distribution (see appendix for discussion of the uniform distribution), then

$$\underset{\mu}{E}\ P_i(u) = \sum_{|x|=i} 2^{-i} P(u|x) \tag{11}$$

$P(u|x)$ being the probability that x will be followed by u. Equation (10) then gives

$$\sum_{i=1}^{n} \sum_{|x|=i} 2^{-i}\ P(u|x) \leq k_\mu \ln 2 \tag{12}$$

k_μ is the length in bits (usually small) of the shortest program, p, such that $M(px) = x$ for all x.

Appendix: Monotonic Machines and Associated Probability Distributions.

A "Monotonic Machine" is a Turing machine with unidirectional input and output tapes and a bidirectional work tape. The input tape is "read only". The output tape is "write only". A *universal* monotone machine was used in the introduction to define a probability distribution over all finite strings. If the monotone machine is *not* universal, we still get a probability distribution in this way, but it is *not* a universal probability distribution. Willis and Levin

(Sol78 p. 424) have shown that just as each monotone machine has an associated probability distribution on strings, conversely each computable probability distribution on strings has an associated monotone machine.

Suppose $\mu(x)$ is a probability distribution on strings and M_μ is the associated monotone machine.

If M is a universal monotone machine then it can simulate M_μ using a finite string, p_μ — so for all x,

$M(p_\mu x) = M_\mu(x)$

In this case it is easy to show (Sol78 p. 424) that $P_M(x) \geq 2^{-p_\mu}\mu$

In section 2 we let $\mu(x)$ be the uniform distribution: i.e. $\mu(x) = 2^{-|x|}$

Here $|x|$ is the length of string x. In this case $M_\mu(x) = x$ and usually M's instructions to simulate M_μ are very simple — they involve nothing more than telling M to copy symbols from the input tape directly onto the output tape.

Acknowledgement: Much thanks to Paul Vitányi and Marcus Hutter for good discussions enabling improvements in the paper.

References

[1] (CT 91) Cover, T., and Thomas, J. *Elements of Information Theory*, Wiley & Sons, N. Y., 1991.

[2] (Hu 05) Hutter, M. *Universal Artificial Intelligence*, Springer-Verlag, Berlin, 2005.

[3] (LV 97) Li, M. and Vitányi, P. *An Introduction to Kolmogorov Complexity and Its Applications*, Springer-Verlag, N.Y.,2nd edition, 1997.

[4] (Sol 78) Solomonoff, R.J. "Complexity-Based Induction Systems: Comparisons and Convergence Theorems," *IEEE Trans. on Information Theory*, Vol IT–24, No. 4, pp. 422 - 432, July 1978. (http://www.world.std.com/~rjs/pubs.html)