Algorithmic Probability

From Scholarpedia

From Scholarpedia, the free peer-reviewed encyclopedia p.19046

Curator: Marcus Hutter, Australian National University
Curator: Shane Legg, Dalle Molle Institute for Artificial Intelligence (IDSIA)
Curator: Paul M.B. Vitanyi, CWI , and Computer science, University of Amsterdam

Algorithmic "Solomonoff" Probability (AP) assigns to objects an *a priori* probability that is in some sense universal. This prior distribution has theoretical applications in a number of areas, including inductive inference theory and the time complexity analysis of algorithms. Its main drawback is that it is not computable and thus can only be approximated in practice.

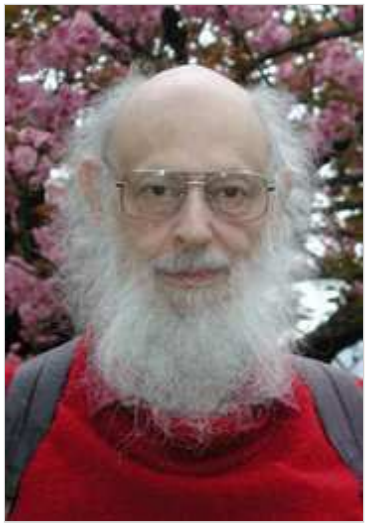

Figure 1: Ray Solomonoff in Bioggio, Switzerland, 2001.

Contents

- 1 Bayes, Occam and Epicurus
- 2 Discrete Universal A Priori Probability
- 3 Continuous Universal A Priori Probability
- 4 Applications
  - 4.1 Solomonoff Induction
  - 4.2 AIXI and a Universal Definition of Intelligence
  - 4.3 Expected Time/Space Complexity of Algorithms under the Universal Distribution
  - 4.4 PAC Learning Using the Universal Distribution in the Learning Phase
  - 4.5 Halting Probability
- 5 References
- 6 External Links
- 7 See Also

Bayes, Occam and Epicurus

In an inductive inference problem there is some observed data $D = x_1, x_2, \ldots$ and a set of hypotheses $H = h_1, h_2, \ldots$, one of which may be the true hypothesis generating D. The task is to decide which hypothesis, or hypotheses, are the most likely to be responsible for the observations. An elegant solution to this problem is Bayes' rule,

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}.$$

To compute the relative probabilities of different hypotheses $P(D)$ can be dropped as it is an independent constant. Furthermore, for a given hypothesis h it is often straight forward to compute $P(D|h)$. A conceptually more difficult problem is to assign the prior probability $P(h)$. Indeed, how can one assign a probability to a hypothesis h *before* observing any data? A theoretically sound solution to this problem was not known until R.J. Solomonoff founded algorithmic probability theory in the 1960s.

Algorithmic probability rests upon two philosophical principles. The first is Epicurus' (342? B.C. - 270 B.C.) principle of multiple explanations which states that one should *keep all hypotheses that are consistent with the data*. From Bayes' rule it is clear that in order to keep consistent hypotheses what is needed is

$\forall h : P(h) > 0$.

The second philosophical foundation is the principle of Occam's razor (1285 - 1349, sometimes spelt Ockham). Occam's razor states that when inferring causes *entities should not be multiplied beyond necessity*. This is widely understood to mean: Among all hypotheses consistent with the observations, choose the simplest. In terms of a prior distribution over hypotheses, this is the same as giving simpler hypotheses higher a priori probability, and more complex ones lower probability.

Using Turing's model of universal computation, Solomonoff (1964) produced a universal prior distribution that unifies these two principles. This work was later generalized and extended by a number of researchers, in particular L. A. Levin who formalized the initial intuitive approach in a rigorous mathematical setting and supplied the basic theorems of the field. There now exist a number of strongly related universal prior distributions, however this article will describe only the discrete universal a priori probability, and its continuous counterpart. See Chapters 4 and 5 of Li and Vitányi (1997) for a general survey.

Discrete Universal A Priori Probability

Consider an unknown process producing a binary string of one hundred 1s. The probability that such a string is the result of a uniform random process, such as fair coin flips, is just $2^{-100} \approx 10^{-30}$, like that of any other string of the same length. Intuitively, we feel that there is a difference between a string that we can recognize and distinguish, and the vast majority of strings that are indistinguishable to us. In fact, we feel that a far more likely explanation is that some deterministic algorithmic simple process has generated this string. This observation was already made by P.-S. Laplace in about 1800, but the techniques were lacking for that great mathematician to quantify this insight. Presently we do this using the lucky confluence of combinatorics, information theory and theory of algorithms. It is clear that, in a world with computable processes, patterns which result from simple processes are relatively likely, while patterns that can only be produced by very complex processes are relatively unlikely. Formally, a computable process that produces a string x is a program p that when executed on a universal Turing machine U produces the string x as output.

As p is itself a binary string, we can define the discrete universal a priori probability, $m(x)$, to be the probability that the output of a universal prefix Turing machine U is x when provided with fair coin flips on the input tape. Formally,

$$m(x) := \sum_{p\,:\,U(p)=x} 2^{-\ell(p)},$$

where the sum is over all halting programs p for which U outputs the string x. As U is a *prefix* universal Turing machine the set of valid programs forms a prefix-free set and thus the sum is bounded due to Kraft's inequality. In fact, the boundedness of the sum should be an obvious consequence from realizing that the above expression is indeed a probability, which hinges on the fact that a prefix machine is one with only {0,1} input symbols and no end-of-input marker of any kind. It is easy to see that this distribution is strongly related to Kolmogorov complexity in that $m(x)$ is at least the maximum term in the summation, which is $2^{-K(x)}$. The Coding Theorem of L.A. Levin (1974) states that equality also holds in the other direction in the sense that $m(x) = \Theta(2^{-K(x)})$, or equivalently,

$$-\log\, m(x) \;=\; K(x) + O(1).$$

As every string can be computed by at least one program on U, it follows that m assigns a non-zero probability to all computable hypotheses and thus this distribution respects Epicurus' principle. Furthermore, if we measure the complexity of x by its Kolmogorov complexity, we see that the simplest strings have the highest probability under $m(x)$, while complex ones have a correspondingly low probability. In this way m also formalises the principle of Occam's razor.

Unfortunately, it is impossible in general to know whether any given program p will halt when run on U, due

to the halting problem. This means that the sum can not be computed, it can only be approximated from below. Stated technically, the function $m()$ is only lower semi-computable. Furthermore, m is not a proper probability measure but rather a semi-measure as $\sum_x m(x) < 1$. This can be corrected by normalising, however this introduces other theoretical weaknesses and so will not be discussed here.

The universal a priori probability m has many remarkable properties. A theorem of L.A. Levin states that among the lower semi-computable semi-measures it is the largest such function in the following sense: If p is a lower semi-computable semi-measure, then there is a constant c such that $cm(x) \geq p(x)$ for all x. In fact, we can take $c = 2^{-K(p)}$ with $K(p)$ the prefix Kolmogorov complexity of the distribution p (defined as the least complexity of a prefix Turing machine lower semi-computing the distribution). This justifies calling m a universal distribution.

Thus, the same mathematical object m arises in three different ways:

- The family of lower semi-computable semi-measures contains an element that multiplicatively dominates every other element: a `universal' lower semi-computable semi-measure m;
- The probability mass function defined as the probability that the universal prefix machine U outputs x when the input is provided by fair coin flips, is the `a priori' probability m; and
- The probability $m(x)$ has $K(x)$ as its associated Shannon-Fano code ($m(x) = 2^{K(x)+O(1)}$).

When three very different definitions (or properties) turn out to define the same object, then this is usually taken in Mathematics to mean that this object is objectively important.

Continuous Universal A Priori Probability

The universal distribution m is defined in a discrete domain, its arguments are finite binary strings. For applications such as the prediction of growing sequences it is necessary to define a similar distribution on infinite binary sequences. This leads to the universal semi-measure M defined as the probability that the output of a monotone universal Turing machine U starts with x when provided with fair coin flips on the input tape. Formally,

$$M(x) := \sum_{p\,:\,U(p)=x*} 2^{-\ell(p)},$$

where the sum is over all minimal programs p for which U 's output starts with x (denoted by *).

Like m, it can be shown that M is only a semi-computable semi-measure. Due to a theorem of L.A. Levin it is the largest such function: If μ is a lower semi-computable semi-measure, then there is a constant c such that $cM(x) \geq \mu(x)$ for all x. The precise definition of these notions is similar but more complicated than in the discrete setting above and thus we refer the reader to the literature (see for example chapter 4 of Li and Vitanyi 1997).

The semi-measure M has similar remarkable properties to m, for example, $-\log M(x)$ $= K(x) - O(\log \ell(x)) = Km(x) - O(\log \ell(x))$, where $Km(x) := \min\{\ell(p) : U(p) = x*\}$ is the monotone complexity. Also M divided by the uniform measure $2^{-\ell(x)}$ is the maximal semicomputable supermartingale. Additionally, if the infinite binary sequences are distributed according to a computable measure μ, then the predictive distribution $M(x_{n+1}|x_1...x_n) := M(x_1...x_{n+1})/M(x_1...x_n)$ converges rapidly to $\mu(x_{n+1}|x_1...x_n) := \mu(x_1...x_{n+1})/\mu(x_1...x_n)$ with μ-probability 1. Hence, M predicts almost as well as does the true distribution μ.

Applications

Algorithmic probability has a number of important theoretical applications; some of them are listed below. For a more complete survey see Li and Vitanyi (1997).

Solomonoff Induction

Solomonoff (1964) developed a quantitative formal theory of induction based on universal Turing machines (to compute, quantify and assign codes to all quantities of interest) and algorithmic complexity (to define what simplicity/complexity means). The important Prediction Error Theorem is due to Solomonoff (1978): Let

$$S_n = \sum_{\ell(x)=n-1} \mu(x)(M(0|x) - \mu(0|x))^2,$$

be the expected squared error made in the n -th prediction of the next bit of an infinite binary sequence $x = x_1 x_2 x_3 ...$ in the ensemble of infinite binary sequences distributed according to computable measure μ when predicting according to the fixed semi-measure M . Then,

$$\sum_n S_n \leq K(\mu)(\ln 2)/2,$$

that is, the total summed expected squared prediction error is bounded by a constant, and therefore, if smooth enough, typically decreases faster than $1/n$. In particular it implies $M(x_{n+1}|x_1...x_n) \to \mu(x_{n+1}|x_1...x_n)$ with μ -probability 1, stated above. This is remarkable as it means that Solomonoff's inductive inference system will learn to correctly predict *any* computable sequence with only the absolute minimum amount of data. It would thus, in some sense, be the perfect universal prediction algorithm, if only it were computable.

AIXI and a Universal Definition of Intelligence

It can also be shown that M leads to excellent predictions and decisions in more general stochastic environments. Essentially *AIXI* is a generalisation of Solomonoff induction to the reinforcement learning setting, that is, where the agent's actions can influence the state of the environment. AIXI can be proven to be, in some sense, an optimal reinforcement learning agent embedded in an arbitrary unknown environment (Hutter 2004). Like Solomonoff induction, the AIXI agent is not computable and thus can only be approximated in practice.

Instead of defining an optimal general agent, it is possible to turn things around and define a universal performance measure for agents acting in computable environments, known as universal intelligence.

Expected Time/Space Complexity of Algorithms under the Universal Distribution

For every algorithm whatsoever, the m -average-case complexity is always of the same order of magnitude as the worst-case complexity for computational resources such as time and storage space. This result of Li and Vitanyi means that, if the inputs are distributed according to the universal distribution, rather than according to the uniform distribution as is customary, then the expected case is as bad as the worst case. For example, the sorting procedure Quicksort has worst-case running time n^2, but expexted running time $n \log n$ on a list of n keys for permutations drawn at random from uniformly distributed permutations of n keys. But if the permutations are distributed according to the universal distribution, that is, according to Occam's dictum that the simple permutations have the highest probabilities, as is often the case in practice, then the expected running time rises to the worst-case of n^2. Experiments appear to confirm this theoretical prediction.

PAC Learning Using the Universal Distribution in the Learning Phase

Using m as the sampling distribution in the learning phase in PAC-learning properly increases the learning power for the family of discrete concept classes under computable distributions, and similarly for PAC learning

of continuous concepts over computable measures by using M. This model of Li and Vitanyi is akin to using a teacher in the learning phase who gives the simple examples first. See Applications of AIT.

Halting Probability

A formally related quantity is the probability that U halts when provided with fair coin flips on the input tape (i.e., that a random computer program will eventually halt). This halting probability $\Omega = \sum_x m(x)$ is also known as *Chaitin's constant* or "the number of wisdom", and has numerous remarkable mathematical properties. For example, it can be used to quantify Goedel's Incompleteness Theorem.

References

M. Hutter. *Universal artificial intelligence: Sequential Decisions based on algorithmic probability*. Springer, Berlin, 2004. http://www.springer.com/chl/home/generic/search/results?SGWID=2-40109-22-34425910-0

M. Li and P. M B. Vitanyi. *An introduction to Kolmogorov complexity and its applications*. Springer, New York, 2nd edition 1997, and 3rd edition 2007. http://www.springer.com/chl/home/generic/search/results?SGWID=2-40109-22-165245230-0

L.A. Levin. Laws of information conservation (non-growth) and aspects of the foundation of probability theory. *Problems Information Transmission*, 10(3):206-210, 1974.

R. J. Solomonoff. A formal theory of inductive inference: Parts 1 and 2. *Information and Control*, 7:1--22 and 224--254, 1964.

R. J. Solomonoff. Complexity-based induction systems: Comparisons and convergence theorems. *IEEE Transactions on Information Theory*, IT-24:422-432, 1987.

A. K. Zvonkin and L. A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25(6):83--124, 1970.

External Links

- Homepages of
  - Leonid A. Levin,
  - Ray Solomonoff
  - Ming Li
  - Paul Vitanyi
  - Marcus Hutter
  - Gregory Chaitin

See Also

Algorithmic information theory, Algorithmic Complexity, Algorithmic Randomness

Marcus Hutter, Shane Legg, Paul M.B. Vitanyi (2007) Algorithmic Probability. Scholarpedia, p.19046
Created: 4 December 2006, reviewed: 23 August 2007, accepted: 23 August 2007
Editor: Marcus Hutter, Australian National University
Retrieved from "http://www.scholarpedia.org/article/Algorithmic_Probability"

Categories: Algorithmic Information Theory | Computational Intelligence