Published by Oxford University Press on behalf of The British Computer Society 2008.
Advance Access publication on February 8, 2008 doi:10.1093/comjnl/bxm120

Three Kinds of Probabilistic Induction: Universal Distributions and Convergence Theorems

RAY J. SOLOMONOFF[1,2,*]

[1]*Computer Learning Research Centre Royal Holloway, University of London, London, UK*
[2]*IDSIA, Galleria 2, CH-6928 Manno-Lugano, Switzerland*
**Corresponding author: rjsolo@ieee.org http://world.std.com/~rjs/pubs.html*

We will describe three kinds of probabilistic induction problems, and give general solutions for each, with associated convergence theorems which show that they tend to give good probability estimates. The first kind extrapolates a sequence of strings and/or numbers. The second extrapolates an unordered set of strings and/or numbers. The third extrapolates an unordered set of ordered pairs of elements that may be strings and/or numbers. Given the first element of a new pair, to get a probability distribution on possible second elements of the pair. Each of the three kinds of problems is solved using an associated universal distribution. In each case a corresponding convergence theorem is given, showing that as sample size grows, the expected error in probability estimate decreases rapidly. The solutions given are very general and cover a great variety of induction problems. Time series prediction, grammar discovery (for both formal and natural languages), curve fitting, the identification problem and the categorization problem, are a few of the kinds of problems amenable to the methods described.

Keywords: algorithmic Probability, universal probability distribution, machine learning, statistical learning theory, classification, identification problem, curve fitting, prediction, regression, grammatical induction, time series prediction, proof of convergence

1. INTRODUCTION

Problems in probabilistic induction are of three general kinds. In the first, we are given a linearly ordered sequence of symbols to extrapolate. There is a very general solution to this problem using the universal probability distribution, and much has been written on finding good approximations to it [1–8]. It has been shown that for long sequences, the expected error in probability estimates converge rapidly toward zero [9].

In the second kind of problem, we want to extrapolate an *unordered* sequence of finite strings and/or numbers. A universal distribution has been defined that solves this problem [10]. We will give a convergence theorem that shows it to give small errors as the number of examples increases—just as with sequential predictions.

In the third kind, operator induction, we have an unordered sequence of ordered pairs of elements (Q_i, A_i) (that may be strings and/or numbers). Given a new Q_i, to obtain the probability distribution over possible A_is. The Qs can be questions in some formal or natural language, the As can be associated answers. The Qs can be inputs to some unknown stochastic device and the As can be outputs (The Identification Problem). The Qs can be description of mushrooms, the As can tell if they are edible or poisonous (The Categorization Problem). The Qs can be numbers and the As can be exact or noisy values of some unknown function of those numbers (The Curve Fitting Problem).

We will give two solutions to this problem based on universal distributions, and give associated convergence theorems that affirm their precision in prediction.

Section 1 deals with the sequential prediction and its universal distribution. This is followed by a convergence theorem for the normalized distribution and some more recent generalizations of it.

Section 2 deals with the extrapolation of a set of unordered strings and/or numbers, and gives an associated convergence theorem.

Section 3 deals with operator induction, and gives the associated convergence theorem.

2. SEQUENTIAL PREDICTION

The universal distribution for sequential prediction is a probability distribution on strings obtained by assuming the

strings are the output of a universal machine with random input. We will at first consider only universal Turing machines with binary unidirectional input and output tapes and an infinite bidirectional work tape. It is possible to get equivalent distributions using more general kinds of universal devices with less constrained input and output.

How can we use this definition to get an expression of the probability of a particular finite string, x?

Let $[S_k]$ be the set of all binary programs for our reference machine, M, such that $M(S_k)$ gives an output with x as prefix. To prevent double counting we have an additional constraint on the set $[S_k]$: dropping the last bit of the string S_k, will give a program with output that does not have x as prefix. With this condition the probability of x becomes the sum of the probabilities of all of its programs:

$$P_M(x) = \sum_k 2^{-|S_k|}, \tag{1}$$

where $|S_k|$ is the number of bits in S_k and $2^{-|S_k|}$, the probability of an input that has S_k as prefix.

Because certain of the codes, S_k do not result in useful output (i.e. the machine prints out part of x, but continues to calculate without printing anything else), the resultant probability distribution is not a measure, but a semimeasure—i.e.

$$P_M(x0) + P_M(x1) < P_M(x).$$

For our first prediction method, we will normalize P_M to create P'_M

$$\begin{aligned} P'_M(x0) &= \frac{P_M(x0)}{P_M(x0) + P_M(x1)} P'_M(x), \\ P'_M(x1) &= \frac{P_M(x1)}{P_M(x0) + P_M(x1)} P'_M(x), \\ P'_M(\wedge) &= 1. \end{aligned} \tag{2}$$

Though there are other possible methods of normalization, it is not difficult to show that equations (2) give us maximum $P'_M(\mathrm{x})/P_M(x)$ for all x. Later we will show that this condition minimizes the expected prediction error of P'_M.

It is easy to use P'_M for prediction:

$$P(x1|x) = \frac{P'_M(x1)}{P'_M(x)} \quad \text{and} \quad \frac{P(x0|x) = P'_M(x0)}{P'_M(x)}. \tag{3}$$

Just how accurate are the predictions of P'_M?

Suppose you have a device μ, generating binary sequences according to some finitely describable stochastic rules. It gives a probability for each of the bits it generates. If you use the universal distribution to get probabilities for each of the bits, there will be a difference between the two probabilities. If you square these probability differences and add them up, the expected value of the sum is bounded by $-1/2 \ln P'_{M,\mu}$. $P'_{M,\mu}$ is the probability that the universal distribution assigns to μ, the generator of the data [9, p. 426].

More exactly:

$\mu(x_{n+1} = 1|x_1,\ x_2,\ x_3 \cdots x_n)$ is the conditional probability distribution according to μ that the $(n+1)$th bit of a binary string is 1, given the previous n bits, $x_1, x_2, x_3 \cdots x_n$.

$P'_M(x_{n+1} = 1|x_1,\ x_2,\ x_3 \cdots x_n)$ is the corresponding probability for P'_M

$x = x_1,\ x_2,\ x_3 \cdots x_n$ is a string constructed using μ as a stochastic source.

Both μ and P'_M are able to assign probabilities to the occurrence of the symbol 1 at any point in the sequence x based on the previous symbols in x.

The convergence theorem says that the total expected squared error between μ and P'_M is given by

$$\begin{aligned} &\underset{\mu}{E} \sum_{m=1}^{n} (P'_M(x_{m+1} = 1|x_1,\ x_2,\ x_3 \cdots x_m) \\ &- \mu(x_{m+1} = 1|x_1,\ x_2,\ x_3 \cdots x_m))^2 < -\frac{1}{2} \ln P'_{M,\mu}. \end{aligned} \tag{4}$$

The expected value is with respect to probability distribution, μ.

$\ln P'_{M,\mu}$ is dependent on just what universal device generated the universal distribution. It is approximately $K \ln 2$, where K is the Kolmogorov complexity of the generator—the length of the shortest program needed to describe it.

Since this total error is independent of the size of the data string being predicted it is clear that the errors in the individual bits must decrease more rapidly than $1/n$, n being the length of the data sequence.

This is a very powerful result. It is clear that the universal distribution gives *very good* probability estimates.

The truth of (4) hinges on the fact that if μ is a computable probability measure then there exists a positive constant $P'_{M,\mu}$ such that

$$P'_M(x)/\mu(x) > P'_{M,\,\mu}$$

and that while $P'_{M,\mu}$ will depend on $\mu(\cdot)$ and $P'_M(\cdot)$, it will be independent of x.

Equation (4) can be usefully generalized:

IF

P_1 and P_2 are any normalized measures on x.

$x(n)$ is a string of length n.

$$\frac{P_2(x(n))}{P_1(x(n))} > \alpha(n) > 0,$$

where $\alpha(n)$ is a function of $P_1(\cdot)$, $P_2(\cdot)$ and n, but not of x
THEN

$$\underset{P_2}{E}\sum_{m=1}^{n}(P_1(x_{m+1}=1|x_1,\ x_2,\ x_3\cdots x_m)$$

$$-P_2(x_{m+1}=1|x_1,\ x_2,\ x_3\cdots x_m))^2 < -\frac{1}{2}\ \ln\ \alpha(n). \quad (5)$$

The convergence theorem of (4) is true if P'_M is a *normalized* universal measure. Peter Gács [11] has shown it to be true for the *unnormalized* semimeasure, P_M, but the associated convergence constant $-1/2\ \ln\ P_{M,\ \mu}$ is much larger than the corresponding constant, $-\ 1/2\ \ln\ P'_{M,\mu}$ for P'_M.

The difference between them is

$$\frac{1}{2}\ \ln\left(\frac{P'_{M,\ \mu}}{P_{M,\ \mu}}\right),$$

where $P'_{M,\mu}/P_{M,\mu}$ is the ratio of the values of the normalization factors for $n=\infty$. We have selected a normalization technique to make it as large as possible.

The result is that the probability errors for the normalized measure, $P'_M(\cdot)$ can converge much more rapidly than those for the semimeasure, $P_M(\cdot)$.

Gacs [11] also shows that the generalization corresponding to equation 5 holds if $P_2(\cdot)$ is an unnormalized semimeasure.

Marcus Hutter [12] shows that these results hold if we use alphabets with any finite number of symbols.

In the foregoing convergence theorems the total squared probability difference is used as loss function. The proofs of the theorems also show the same convergence for the Kullback–Liebler loss function (which is greater than or equal to the square loss function—resulting in stronger theorems).

Hutter [12] considers more general loss functions for the universal distribution and obtains associated convergence theorems.

3. INDUCTION ON UNORDERED SETS

3.1. The problem and a solution

We have an unordered set of n finite strings of symbols, D_1, D_2, ... ,D_n. Given a new string, D_{n+1}, what is the probability that it belongs to the set? Or given a string, a, how must it be completed so it is most likely to be a member of the set? Or, given a string a and a set of possible completions, $[ab_j]$, what is the relative probability of each of these completions?

A common example of unordered set prediction occurs in natural and formal languages. We are given a set of examples of strings that are acceptable sentences. Given a new string, what is the probability that it is acceptable? A common solution technique is to devise a well-fitting stochastic grammar for the known set of strings. The universal distribution gives a criterion for goodness of fit of such grammars [3, pp.240–251; 13].

The universal distribution P_M, is a weighted sum of all finitely describable semimeasures on finite strings:

$$P_M([D_i]) = \sum_j \alpha_j \prod_{i=1}^{n} P_j(D_i), \quad (6)$$

where n is the number of strings in the set $[D_i]$ and

α_j is the weight of the jth semimeasure on finite strings.

$\alpha_j = 2^{-|a_j|}$, where a_j is the shortest description of $P_j(\cdot)$ and $|a_j|$ is the number of bits in a_j

The M subscript of P_M indicates that the functions P_j are to be described with reference to machine, M. Since M is universal, it can be used to describe any describable function.

Suppose that $[D_i]\ i=1, \ldots ,n$ is a set of n strings generated by some unknown stochastic device, $\mu(\cdot)$. What is the probability that our universal distribution assigns to a new string, D_{n+1} ?

It is just

$$P(D_{n+1}) = P_M\frac{\left([D_i]\bigcup D_{n+1}\right)}{P_M([D_i])}. \quad (7)$$

The probability assigned to $[D_i]$ by its creator, $\mu(\cdot)$, is

$$\mu([D_i]) = \prod_{i=1}^{n}\mu(D_i). \quad (8)$$

For a suitable set of strings, $[D_i]$, the probability assigned by P_M in (6) can be very close to those assigned by $\mu(\cdot)$, the generator of $[D_i]$, in (8). In section 3, we will discuss Operator Induction and prove an associated convergence theorem. Section 3.3 shows that this convergence theorem implies a convergence theorem for (6), insuring small expected errors between the probability estimates of $P_M(\cdot)$ and those of $\mu(\cdot)$.

4. OPERATOR INDUCTION

In the Operator Induction problem, we are given an unordered set of n strings and/or number pairs, $[Q_i, A_i]$. Given a new Q_{n+1}, what is the probability distribution over all possible A_{n+1}? We will give two solutions.

4.1. First Solution

In the first, we consider the problem to be an extrapolation of an unordered set of finite strings, D_i, in which $D_i = (Q_i, A_i)$

Equation 6 is used to obtain a probability distribution on all unordered sets of Q_i, A_i pairs and (7) gives us a probability

distribution over (Q_{n+1}, A_{n+1}) — i.e. $P(Q_{n+1}, A_{n+1})$ for all possible A_{n+1}.

Then

$$P(A_{n+1}) = \frac{P(Q_{n+1}, A_{n+1})}{\sum_i P(Q_{n+1}, A_i)}. \quad (9)$$

4.2. Second Solution

For the second solution to the operator problem, we express the probability of an arbitrary A_{n+1} directly as a function of the data set, $[Q_i, A_i]$. For this data set, the probability distribution of A_{n+1} is

$$\sum_{j=1} a_0^j \prod_{i=1}^{n+1} O^j(A_i|Q_i). \quad (10)$$

Here $O^j(\cdot|\cdot)$ is the jth possible conditional probability distribution relating its two arguments. $O^j(A_i|Q_i)$ is the probability of A_i, given Q_i, in view of the function O^j.

We would like to sum over all *total* recursive functions, but since this set of functions is not effectively enumerable, we will instead sum over all *partial* recursive functions, which *are* effectively enumerable.

a_0^j is the a priori probability of the function $O^j(\cdot|\cdot)$. It is approximately $2^{-l(O^j)}$, where $l(O^j)$ is the length in bits of the shortest description of O^j.

We can rewrite (10) in the equivalent form

$$\sum_{j=1} a_n^j O^j(A_{n+1}|Q_{n+1}). \quad (11)$$

Here,

$$a_n^j = a_0^j \prod_{i=1}^{n} O^j(A_i|Q_i).$$

In (11), the distribution of A_{n+1} is a weighted sum of all of the O^j distributions—the weight of each O^j being the product of its *a priori* probability and the probability of the observed data in view of O^j.

Section 3.3 shows that even with a relatively short sequence of Q, A pairs, these distributions tend to be very accurate. If we use the a_0^j to express all of our *a priori* information about the data, they are, perhaps, the most accurate possible.

Since we cannot compute this infinite sum using finite resources, we approximate it using a finite number of large terms—terms that in (11) have large a_n^j values. While it would seem ideal to include the terms of maximum weight, it has been shown to be impossible to *know* if a particular term is of maximum weight. The best we can do is to find a set of terms of largest total weight in whatever time we have available.

We can completely characterize the problem of operator induction to be finding, in whatever time is available, a set of functions, $O^j(\cdot|\cdot)$ such that $\sum_j a_n^j$ is as large as possible.

4.3. Convergence Proof

We will show that for an adequate sequence of (Q_i, A_i) pairs, the predictions obtained by the probability distribution of (10) can be expected to be extremely good.

To do this, we hypothesize that the sequence of A_i answers that have been observed, were created by a probabilistic algorithm, $\mu(A_i|Q_i)$ and that μ can be described with k bits.

Any probability distribution that assigns probabilities to every possible A_i, must also assign probabilities to each bit of A_i:

Suppose that a_r is a string of the first r bits of A_i. Then the probability given by μ that the $(r+1)$th bit of A_i is 1 is

$$\frac{\sum_j \mu(a_r 1 x^j|Q_i)}{\sum_j \mu(a_r x^j|Q_i)},$$

where x^j ranges over all finite strings.

Similarly, $P(\cdot)$ the algorithm of (10), can be used to assign a probability to every bit of every A_i. We will represent the sequence of A_is by a string, Z, that is formed by concatenating these A_is then separating them by the symbols, s; denoting 'space'. Z, then, is a sequence of symbols from the ternary alphabet 0, 1, s. Using an argument similar to the foregoing, it is clear that both μ and P are able to assign probabilities to the space symbol, s as well as to 0, and 1, since each of them must be able to specify when each A_i string terminates.

We have, then, two probability distributions on the ternary strings, Z. In the first distribution, μ is the creator of the observed sequence, and in the second distribution, P, through (10), tries to predict the symbols of Z.

For two such probability distributions on *ternary* strings, we can apply Hutter's [12] generalization to arbitrary alphabet, of the generalized convergence theorem, (5). The expected value, with respect to μ (the 'generator'), of the sum of the squares of the differences in probabilities assigned by μ and P to the symbols of the string are less than $-\ln c$, c being the largest positive number such that $P/\mu > c$ for all arguments of P and μ.

More exactly,

$$\sum_l \mu(Z_l) \sum_{i=1}^{n} \sum_{j=0}^{h_i^l+1} \sum_{t=0,1,s} \left(P_{i,j}^l(t) - \mu_{i,j}^l(t)\right)^2 < k \ln 2, \quad (12)$$

where l sums over all strings Z_l that consist of n finite binary strings separated by s symbols (spaces), A_i^l is the ith A of Z_l,

$P^l_{i,j}(t)$ is the probability as given by P that the jth symbol of A^l_i will be t, conditional on previous symbols of A^l_is in the sequence, Z_l and the corresponding Qs, t takes the values 0,1 and s, $\mu^l_{i,j}(t)$ is defined similarly to $P^l_{i,j}(t)$, but it is independent of previous A^l_is in the sequence and h^l_i is the number of bits in A^l_i. The $(h^l_i + 1)$th symbol of A^l_i is always s.

The total number of symbols in Z_l is $\sum_{i=1}^{n}(h^l_i + 1)$.

$\mu(Z_l)$ is the probability that μ assigns to Z_l in view of the sequence of Qs, k is the length in bits of the shortest description of μ.

This implies that the expected value with respect to μ of the squared 'error' between P and μ, summed over the individual symbols of all of the A_i, will be less than $k \ln 2$

Since the total number of symbols in all of the answers can be very large for even a small number of questions, the error per symbol decreases rapidly as n, the number of Q, A pairs increases.

Equation (12) gives a very simple measure of the accuracy of equation (10). There are no 'order of one' constant factors or additive terms. A necessary uncertainty is in the value of k. We normally will not know its value, but if the generator of the data has a long and complex description, we are not surprised that we should need more data to make good predictions—which is just what (12) specifies.

The value of the constant, k, depends critically on just what universal reference machine is being used to assign *a priori* probability to the O_j and to μ. Any *a priori* information that a researcher may have can be expressed as a modification of the reference machine—by inserting low cost definitions of concepts that are believed to be useful in the needed induction—resulting in a shorter codes for the $O_j(\cdot)$, for μ, (a smaller k), and less error.

We believe that if all of the needed a priori information is put into the reference machine, then (10) *is likely to be the best probability estimate possible*.

At first glance, this result may seem unreasonable. Suppose we ask the system many questions about Algebra, until its mean errors are quite small—then we suddenly begin asking questions about Linguistics—certainly we would not expect the small errors to continue! However, what happens when we switch domains suddenly, is that k suddenly increases. A μ that can answer questions on both Algebra and Linguistics has a much longer description than one familiar with Algebra only. This sudden increase in k accommodates large expected errors in a new domain in which only a few questions have been asked.

4.4. Alternative Induction Techniques

If we set $Q_i = \wedge$ $(i = 1, \ldots, n)$ in (10), it becomes clear that the Equation (6) for induction on unordered sets is a special case of operator induction, and that the convergence theorem (12) holds for (6) as well. This also assures convergence of the operator induction technique of Section 2.1.

Is there any advantage in using (9) rather than (10) for operator induction?

Equation (9) exploits regularities in the set $[Q_i, A_i]$. It includes regularities in the set $[Q_i]$—which we do not use—so it would seem that we are doing more work than is necessary. In (10), we only find regularities in functions relating Q_i to A_i. Such regularities may be easier to find than regularities in the more complex object $[Q_i, A_i]$. In general, however, the finding of regularities for either of the techniques will depend critically on just what problem is being solved.

5. FUNDING

This research was supported by AFOSR Contract No. AF 49(638)-376, Grant No. AF–AFOSR 62–377, and Public Health Service Grant No. GM 11021-01.

REFERENCES

[1] Solomonoff, R.J. (1960) A Preliminary Report on a General Theory of Inductive Inference. Report ZBT-138, Zator Co., Cambridge, MA. http://world.std.com/~rjs/pubs.html

[2] Solomonoff, R.J. (1964) A formal theory of inductive inference. *Inf. Control Part I*, **7**, 1–22. http://world.std.com/~rjs/pubs.html

[3] Solomonoff, R.J. (1964) A formal theory of inductive inference. *Inf. Control Part II*, **7**, 224–254. http://world.std.com/~rjs/pubs.html

[4] Wallace, C.S. and Boulton, D.M. (1968) An information measure for classification. *Comput. J.*, **11**, 185–194.

[5] Wallace, C.S. and Freeman, P.R. (1987) Estimation and inference by compact coding. *J. Roy. Stat. Soc. Se. B Stat. Methodol*, **49**, 240–252.

[6] Willis, D.G. (1970) Computational complexity and probability constructions. *J. Assoc. Comput. Mach.*, **17**, 241–259

[7] Rissanen, J. (1978) Modeling by the shortest data description. *Automatica*, **14**, 465–471.

[8] Rissanen, J. (1987) Stochastical complexity. *J. Roy. Stat. Soc. Se. Stat. B Methodol*. **49**, 223–239.

[9] Solomonoff, R.J. (1978) Complexity-based induction systems: comparisons and convergence theorems. *IEEE Trans. Inf. Theory*, **IT–24**, 422–432. http://world.std.com/~rjs/pubs.html

[10] Solomonoff, R.J. (1999) Two kinds of probabilistic induction. *Comput. J.*, **64**, 256–259. http://world.std.com/~rjs/pubs.html

[11] Gács, P. (1997) Theorem 5.2.1. In Li, M. and Vitányi, P. (eds) *An Introduction to Kolmogorov Complexity and Its Applications*, pp. 328–331. Springer, NY.

[12] Hutter, M. (2002) *Optimality of universal Bayesian sequence prediction for general loss and alphabet*. http://www.idsia.ch/~marcus/ai/

[13] Li, M. and Vitányi, P. (1997) *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, NY.