Does Algorithmic Probability Solve the Problem of Induction?

Ray Solomonoff

Visiting Professor, Computer Learning Research Center
Royal Holloway, University of London
Mailing Address: P.O.B. 400404, Cambridge, Ma. 02140, U.S.A.
rjsolo@ieee.org

1 Introduction

We will begin with a definition of Algorithmic Probability (ALP), and discuss some of its properties. From these remarks it will become clear that it is extremely effective for computing probabilities of future events — the best technique we have. As such, it gives us an ideal theoretical solution to the problem of inductive inference. I say "theoretical" because any device as accurate as ALP must necessarily be incomputable.

For practical induction we use a set of approximations of increasing power to approach ALP. This set is called Resource Bounded Probability (RBP), and it constitutes a general solution to the problem of *practical* induction. Some of its properties are quite different from those of ALP.

The rest of the paper will discuss philosophical and practical implications of the properties of ALP and RBP.

It should be noted that the only argument that need be considered for the use of these techniques in induction is their effectiveness in getting good probability values for future events. Whether their properties are in accord with our intuitions about induction is a peripheral issue. The main question is "Do they work?" As we shall see, they *do work*.

2 Algorithmic Probability: A Definition and Some Properties

ALP may be regarded as a Bayesian method of extrapolating finite binary strings. The requisite prior probability distribution is obtained by assuming

the strings were generated as the output of a Universal Turing Machine with unidirectional output tape and random bits on its unidirectional input tape.

Let's define a "description of string x with respect to machine M" as an input string to M that results in output x. Usually x will have many different descriptions. Say d_i is the i^{th} description of x and d_i is l_i bits long. Then the probability that d_i will occur as random input and produce x as output is just 2^{-l_i}

To obtain the total probability that s will be produced, we simply sum over all possible descriptions, obtaining

$$P(x) = \sum_{i=1}^{\infty} 2^{-l_i} \tag{1}$$

If l is the length of the shortest description of s, a useful approximation to this sum can be obtained by just taking the largest term, 2^{-l}. See Sol 78, Pp. 422-423 for a more detailed discussion.

For earliest work in this area as well as earliest work on important developements see Sol 60, Sol 64, Kol 65, Cha 66, Mar 66, Wal 68, Wil 70, Zvo 70, Cov 74, Ris 78. Li and Vitányi (Liv 93) gives an excellent review and history.

Early attempts to justify ALP were based on heuristic arguments involving Occam's razor, as well as many examples in which it gave reasonable answers. At the present time, however, we have a much stronger justification than any heuristic argument. ALP is the only induction technique known to be complete.

By this we mean that if there is any describable regularity in a body of data, ALP will discover it using a relatively small sample of the data [1]. As a necessary consequence of its completeness, this kind of probability must be incomputable. Conversely, any computable probability measure cannot be complete.

We are using the term "incomputable" in a special way. ALP is as "computable" as the value of π – but with one important difference; when we make successive approximations to the value of π,we know how large the error in each approximation can be. In the case of ALP, we have a procedure for successive approximations that is guaranteed to converge to the right value. However, at no point in the calculation can we make a useful estimate of the error in the current approximation.

This might be regarded as a serious criticism of the use of approximations to ALP to solve practical problems, but it is not. It is a difficulty shared by all probability evaluation methods. If they are complete, and hence "incomputable", they have unknown error size in any finite approximation of them. If they are computable, then they *must* be incomplete. This incompleteness implies that there have to be regularities that are invisible to them. When used with data having regularities of these kinds, computable methods will have er-

[1] See Appendix A for The Convergence Theorem, which is a more exact description of "completeness". Sol 78 contains a proof of the theorem.

rors of unknown size. It is likely that all quantitative methods of dealing with uncertainty have this associated uncertainty of error size.

It has only been through the analysis of ALP that these very general limitations of knowledge of error size have become known.

Over the years there has been a general impression in the scientific community that the incomputability of ALP would make it impossible to use it for statistical prediction (see for example, Ris 95 p.197).

From the beginning, however, this difficulty was recognized and methods for dealing with it were proposed (Sol 64a section 3.1.2.1). Willis (Wil 70) formalized one of these methods in what we will call Resource Bounded Probability (RBP). He considered various limitations on computational resources, such as the amount of memory or the amount of time available. With time and/or memory limitations, equation 1 can be summed over only a finite number of descriptions, giving us a computable approximation to ALP. As we increase the available resources, we get closer and closer to ALP.

The most efficient way to implement RBP is to approximate equation 1 by the largest lower bound on $P(x)$ that can be demonstrated in time, T. This is usually done by finding many short codes for x that give terms summing to that bound. This kind of approximation to ALP is an example of a "Time limited optimization problem" [2].

By getting as large a value of the sum as we can, we get as close as possible to ALP in the allotted time. This seems like the best possible way to spend our time computing probabilities, since, at present, ALP is the best theoretical induction system we have.

From the forgoing, it might appear that we need not consider any other methods of induction. Though this is indeed true, it does not limit us to any great extent. Just about all methods of induction can be regarded as approximations to ALP, and from this point of view, we can criticize and optimize them. Criticism is always easier if you have an idealized (possibly unrealizable) standard for comparison. ALP provides such a standard.

When it was first defined, one of the attractive properties of ALP was its relative lack of dependence upon just what universal machine was used for reference. If $P_{M1}(x)$ and $P_{M2}(x)$ are the algorithmic probabilities associated with string x for machines $M1$ and $M2$ respectively, then

$$\log_2 P_{M2}(x) - \log_2 P_{M1}(x) \leq b$$

Here b is the number of bits needed for Machine $M1$ to simulate Machine $M2$. $\log_2 P_{M1}(x)$ is within 1 bit of the length of the encoding for x that would result

[2]Time (or resource) limited optimization problems are one of the commonest types of problems occurring in science and engineering. To formalize the idea, suppose we are given a machine (or program) whose inputs are finite strings s, and outputs $M(s)$ are real numbers. We are given a fixed time limit, T. The problem is to find within time T, a string, s, such that $M(s)$ is as large as possible. If no time limit is given, the problem generalizes to "Anytime Algorithms."(Dea 88). The Sigart Bulletin of the ACM Vol 7, No.2 April 1996, has references and recent work on "Anytime Algorithms".

if $P_{M1}(x)$ were used for optimum data compression. For transmitting a few megabytes of compressed data, the 100 kilobytes needed to translate from one machine to another is not significant - the reference machines $M1$ and $M2$ are about equally good for data compression.

l00 Kilobytes may not be a significant difference in data compression, but it translates to a factor of 2^{800000} between the probabilities $P_{M1}(x)$ and $P_{M2}(x)$ – a very large factor indeed! Choice of reference machine is *much* more important for probability calculations than it is for information compression.

The completeness property of ALP is also sensitive to choice of reference machine, but much less than in the previous discussion. This property requires only that the reference machine be universal, but the rate at which its probability estimates converge to the correct values, as a function of corpus length (or sample size), depends much on just what reference machine was used. The convergence is most rapid if the reference machine is able to describe the mechanism generating the data by a short string of bits.

The reference machine defines the a priori probability distribution of ALP. If this machine finds it "easy" to describe a particular stochastic generator, then its a priori probability distribution is "close" to that of the generator.

For quite some time I felt that the dependency of ALP on the reference machine was a serious flaw in the concept, and I tried to find some "objective" universal device, free from the arbitrariness of choosing a particular universal machine. When I finally found a device of this sort, I realized that I really didn't want it - that I had no use for it at all! Let me explain:

In doing inductive inference, one begins with two kinds of information: First, the data itself, and Second, the a priori data - the information one had before seeing the data. It is possible to do prediction without data, but one cannot do prediction without a priori information. In choosing a reference machine we are given the opportunity to insert into the a priori probability distribution any information about the data that we know before we see the data.

If the reference machine were somehow "objectively" chosen for all induction problems, we would have no way to make use of our prior information. This lack of an objective prior distribution makes ALP very subjective - as are all Bayesian systems.

Application of any formal probability model to real world problems gives rise to another source of subjectivity. I have described how to do prediction of a binary string. To do prediction of events in the real world, it is necessary to map these events and the events that preceded them to binary strings in a reversible way. While it is always possible to do this, some methods are much better than others. A mapping technique that is particularly bad is one that omits important real world information. Whenever we formulate a real world problem for formal analysis and prediction, it is almost always necessary to omit information. This is because our world is very complex and contains an enormous amount of information. For prediction problems in the physical sciences, we have very good ideas as to which information must be included,

and we usually seem to be right — the apparent success of the physical sciences has been closely associated with this fact.

In many cases it is also possible to include much irrelevant information, obscuring the signal in a sea of noise. This increases the amount of relevant data ALP needs to discover regularities. RBP will in addition need more computing time to discover the regularity – perhaps beyond what is available.

For very complex problems, like predicting earthquakes, or effects of medicines on humans, we have very poor understanding of the phenomena of interest. Although we may have an enormous amount of information, we may have no idea as to what part of it is relevant. This difficulty becomes very clear in the application of artificial neural nets to problems of this sort. The big problem isn't finding a relationship between a known set of variables and an unknown variable to be predicted. The big problem is to discover which variables are the relevant predictors.

Another kind of subjectivity in mapping from the real world derives from the method of coding. The representation can make certain regularities in the world quite obvious, or without omitting essential information, the representation can badly obscure these regularities. In the latter case, ALP will need more data before it discovers the regularity - or in the case of RBP, it may not have enough time to discover the regularity at all.

This last kind of subjectivity seems close to that which is introduced in the choice of a reference machine.

Another feature of RBP is the subjective bias it introduces in the choice of approximation methods. In practical applications of probability , we are usually interested in *ratios* of probabilities, and very rarely in the probability values themselves. Suppose we want to compute the ratio of probabilities of events X and Y, with the same reference machine for both, using RBP: If we use the same resources on both X and Y, but our approximation techniques for X are more efficient than those used for Y, then for small resources, X will have spuriously high probability. For sufficiently large resources, this kind of bias becomes negligible, and we approach the ratio assigned by ALP. We can never know, however, how big "sufficiently large" must be.

Both choice of reference machine and mode of representation can be viewed as modifications of the apriori probabilities assigned to sequences of events by ALP. As such, they afford an opportunity to bring into the extrapolation process information the user had before seeing the data.

This certainly makes the results "subjective". If we value objectivity, we can routinely reduce the choice of machine and representation to certain universal "default" values – but there is a tradeoff between objectivity and accuracy. To obtain the best extrapolation, we must use whatever information is available, and much of this information may be subjective.

Consider two physicians, A and B: A is a conventional physician: He diagnoses ailments on the basis of what he has learned in school, what he has read about and his own experience in treating patients. B is not a conventional

physician. He is "objective". His diagnosis is entirely "by the book" – things he has learned in school that are universally accepted. He tries as hard as he can to make his judgements free of any bias that might be brought about by his own experience in treating patients.

As a lawyer, I might prefer defending B's decisions in court, but as a patient, I would prefer A's intelligently biased inductions.

To the extent that a statistician uses objective techniques, his recommendations may be easily defended, but for accuracy in prediction, the additional information afforded by subjective information can be a critical advantage.

At this point, it may seem that RBP is much like ALP, but is a bit less accurate. This is occasionally true, but often the two are worlds apart. In a complex prediction problem important qualitative changes can occur in the solution, depending on whether one has a millisecond, a minute, an hour, a month, a lifetime or an infinite amount of time (as in the case of ALP) to analyze the problem.

A feeling for the difference between the two can be obtained if we ask for the probability that the $10^{20}th$ digit of π is 7. ALP in its infinite wisdom would assign a probability close to zero or close to one. At our present state of mathematical and technical development we don't know which. For RBP, the answer is quite different. For small resources, the answer is .1, but as soon as the machine has enough resources to compute the exact value of the digit, the probability gets close to zero or one, just as it does when ALP is used.

3 The Dimensions of Uncertainty

Both ALP and RBP have the usual uncertainty in probability assignment due to finite sample size. If we have data on n flips of a biased coin, our uncertainty in the probability of "heads" would be proportional to $n^{-1/2}$. The less data we have, the more uncertain we are of the probability value.

If we consider only Bernoulli models then the cardinality of the data will be an adequate characterization of uncertainty due to finite sample size. If any other models are used, we must characterize the uncertainty in other ways. The only way that is adequate for all model classes is to give all of the data in full detail.

Both kinds of probability have in addition what is called "model uncertainty" (Las 96). Loosely speaking, ALP may be viewed as a weighted average of the predictions of an infinite set of models of the data. Each model is able to assign a probability to the data, as well as a probability to each possible continuation of the data. The weight assigned to each model is proportional to the a priori probability of that model (obtained approximately from the length of its shortest description) multiplied by the probability that the model assigns to the data. [3]

[3] In addition to the conventional models described, ALP also uses partial recursive models. Consideration of such models will not materially modify the present discussion.

In ALP, a large source of uncertainty is in the a priori weights of the models, which are dependent upon the very subjective choice of reference Turing machine. We will come back to this issue again in the section on "The Aesthetics of Science".

In RBP — which is the only kind of probability that we actually use — there is an additional source of uncertainty. Our limited resources make it necessary for us to omit many models from our weighted average. For the set of models *not considered*, we do not know any of the associated predictions or their weights. If one of the models not considered had very accurate predictions, and much weight- the error associated with our omitting this model would be quite large.

No matter how good the predictions of RBP seem to be, it is possible that by using more resources (say 10 more minutes of computation) we would discover a model of much weight that would greatly change our predictions. Furthermore, at the present time, it seems that we have no way to assign a probability to such an occurrence. This makes it impossible to apply decision theory in any exact way to problems based on empirical data.

We are uncertain of the probabilities needed for the decision and we cannot express this uncertainty probabilistically.

This is a serious cause for concern. Many people (myself among them) regard decision making as the only legitimate practical application of probability.

One way to characterize our model uncertainty: we can simply describe our reference Turing machine and tell which models we've used in our approximation. This information makes it possible for a new researcher to continue where the old one left off.

It also enables us to implement a kind of partial ordering of prediction reliability. If two predictions are based upon the same universal machine, but the first one considers all models that the second one considers plus some extra models — then the first prediction is at least as good as the second (and probably better).

4 The Mechanics of ALP

Before going further, it would be well to look at just how ALP works. It looks for short descriptions of the data, so it can assign as large a probability to the data as possible. Suppose we have the sequence x, and we want to know the probability of the continuation 1 rather than 0. If ALP assigns a probability $P(x0)$ to the sequence $x0$ and $P(x1)$ to the sequence $x1$, then the relative probability of the continuations 0 and 1 is just $P(x0)/P(x1)$.

ALP assigns these probabilities by finding short codes for $x0$ and $x1$. If no short codes exist, the shortest code is the sequence itself. In this case if $x0$ and $x1$ are both of length n, we would assign probabilities of about 2^{-n} to both $x0$ and $x1$, giving a probability ratio of 1 for the two possible continuations of x.

One common method of devising a short code uses definitions. If a certain

subsequence occurs many times in the data, we can shorten the data description by defining a specially designed short code for that subsequence whenever it occurs. The "bit cost" of this encoding is not only the accumulation of the many occurrences of the special code — we must add in the bit cost of the definition itself. In general a sequence must occur at least twice before we can save any bits by defining it.

More generally, suppose there are several subsequences of symbols in the data, each with its own frequency of occurrence. We can assign an equivalent code length of *exactly* $-\log_2 f$ to a subsequence whose relative frequency is f. Huffman coding obtains somewhat similar results.

5 Paradoxes in Induction

One of the great values of ALP is that it enables us to construct a reasonable a priori distribution, based on whatever language we have found useful in the past. This construct enables us to deal with various classical difficulties in induction theory.

An important class of difficulties in our understanding of induction, was pointed out by Nelson Goodman in his "Grue Paradox". (Goo 51).

He first defines the predicate "Grue" to mean, "Green before Jan. 1, 1999 — blue after that date".

We observe some emeralds. They seem to be green. They are also seem to be grue. We observe them many times. They *still* seem green, and *still* seem grue. We expect that in the year 2000 they will still be green — but they should just as well be blue, since they were grue in the past and we expect a quality to continue in time.

Grue may seem a bit contrived, but many things in the real world display similar discontinuities — a caterpillar discontinuously changing into a butterfly, or ordinary water changing to steam as its temperature is raised.

Goodman's example showed that all hypothetical models for data are not of equal a priori likelihood and that we had no good way to assign more likelihood to some and less to others. His proposed solution to the problem was the concept of "entrenchment" — the idea that a model is more likely, to the extent that it has been useful in the past in prediction. ALP quantifies this in the same way , by noting the high frequency of use of a concept in the past gives it a short equivalent code length. Descriptions using it are shorter and are given correspondingly more weight in prediction.

Another classical problem in prediction that is quite similar, is the problem of geometric probability. Suppose we want to do induction involving the density of an object, and know only that its density is between 1 and 2. In absence of further information, we might assume a uniform density distribution from 1 to 2.

However, the densities of 1 and 2 are equivalent to specific volumes of 1 and

1/2 respectively. If we assume a uniform specific volume distribution between 1 and 1/2, this is quite different from a uniform density distribution between 1 and 2.

We can use the quantification of "entrenchment" to deal with this ambiguity. If, as in the Grue problem, "density" had been more useful in the past, then we would give it a short code and more weight in the present problem.

6 The Aesthetics of Science

ALP gives us a way to begin to understand the very important role of "Beauty" in science.

As researchers, we are all familiar with the "Ah Ha" phenomenon. We work on a problem for a long time, trying to integrate a mass of data that seems to point everywhere and nowhere. Then, sometimes much later, things begin suddenly to fit together, into a simple general scheme — "Ah Ha!" — We have a Beautiful understanding of the whole problem!

What are the dimensions of Beauty? How do we characterize it, how dow we measure it? I propose that beauty in science is a short code. It's a significant local minimum in the search for a way to express all of the data in a simple, unified way — to make it all hang together.

Concepts of beauty differ significantly among scientists. One scientist will consider a theory beautiful if it expresses the data using a few concepts that he *himself* has found useful in the past. This is related to Goodman's entrenchment, and ALP's short abbreviations for concepts that have been useful in the past.

Another scientist, having had somewhat different experiences and having coded his data somewhat differently, will have a different set of code lengths for the concepts he uses to describe the data. As a result, a theory that is very beautiful, very simple, and has a very short code for one scientist will be ugly, ad-hoc, and have a very long code for another.

How can the scientific community decide between the two?

Ideally what should be done is to give each scientist's opinion a weight proportional to his or her success in framing successful theories in the past [4].

What is *actually* done, is only approximately what I've described. Einstein's opinions were given enormous weight because of his past success in framing good, unifying theories. In the simpler sciences, such as physics and chemistry, evaluation of a scientist's work by his or her colleagues is usually based on relatively objective measures of performance. In the more complex sciences, [5]

[4] A good way of doing this might be based on T. Cover's Universal portfolios (Cov 96), an optimized method of betting on stock market price fluctuations. It would take into account not only past success or failure of the two scientists, but cross correlations between their work.

[5] I'm using "simple and complex" to refer to the general methodological state of a science. In physics and chemistry, we usually know what variables affect what other variables. We have good ideas as to the general forms of the laws relating them. In the complex sciences, there are many things to observe and measure, but little certain knowledge of which are related to

such as psychology, geology, and economics, evaluations of scientists are more strongly related to their rhetorical skills. This is, however, but one of the many serious barriers to progress in the complex sciences.

Schmidhuber (Sch 95) has used short codes to define "beauty" of more conventional art forms.

7 ALP v.s Induction

Our discussions thus far have not distinguished between induction, prediction, and probability distributions on the future. Most scientists think of prediction through induction only. They examine the data, trying to find a regularity or "law" that best fits the data. They then make predictions or probability distributions on the future, using the regularity they have found.

ALP does not explicitly use induction. It goes directly from the data to the probability distribution for the future. This is equivalent to using a weighted average of the predictions of all possible theories - the weight depending on how well the theory fits past data, as well as the shortness of the description of theory itself.

In the physical sciences, the single "best" Theory, is usually much better than the others, so selecting the single best law is not much different from ALP. In the complex sciences - such as sociology, psychology and geology, - the tenth best theory may be not far behind the best, and ALP's weighing of all of them can be considerably different from choosing the single best one.

Do we need scientific laws at all? ALP can give us predictions at least as good as any of them – and in most sciences, it does better. Scientific laws, however, have other functions, and one of them is closely related to the sociology of science. Scientific laws are usually compactly expressed, enabling easy communication of good ideas.

8 ALP and AI

Since about 1980, Machine Learning has become one of the major branches of Artificial Intelligence.

There are four factors that Learning and Prediction systems of this kind must address:

- The prediction itself.
- The reliability of that prediction.
- The sample size.

which. The general forms of the relationships are as yet unclear. All sciences in their early stages are quite complex. Simplicity is acquired (if ever) with maturity.

- Computation cost.

RLP deals with each of the factors in what appears to be an optimum manner.

Only a small fraction of the early work on Machine Learning seriously considered all of these factors. This sharply limited its value , both in practical applications and in shedding light on human problem solving.

More recently there has been much good work in applying machine learning to practical problems such as protein folding and financial market prediction — areas in which all four factors are critical.

In the protein folding problem, the data is finite and slowly growing. It is fairly accurate, and easy to obtain. Though there is much known about the physical and chemical constraints in folding, this information is not always used in the models.

In the financial markets the amount of data is enormous and grows rapidly. For the most part, it is not very accurate, and may contain systematic errors.

The researcher is adrift in a vast ocean of data, and he doesn't know what part of it is relevant to his problem. The science of economics gives only a few useful clues on just what kinds of variables affect what others or just what functional forms might be relevant. The prediction problem is very competitive and newly discovered regularities in the data disappear as they are exploited by traders in the markets. This gives a training sequence of constantly increasing difficulty — an ideal environment for learning to solve very difficult problems in the complex sciences. An important factor slowing down progress in this area is related to its competitive nature. A prediction technique is usually published only when it is obsolete — no longer relevant to today's markets.

9 Appendix A: The Convergence Theorem

Suppose x is an infinite binary string.

$P_S(x_n, 1)$ is the probability that the bit 1 will follow the first n bits of x, as given by a certain stochastic model, S.

$P_A(x_n, 1)$, is this probability as given by ALP, using reference machine M. Suppose the stochastic source S is describable by M, in b bits. Then

$$E \sum_{n=1}^{k} (P_S(x_n, 1) - P_A(x_n, 1))^2 \leq \frac{b}{2} \ln 2$$

The "Expected Value" is with respect to the stochastic source, S, which induces a probability density on all strings, x.

The expected value of the sum of the squared errors, between the conditional probabilities assigned by ALP and by S, is less than or equal to $\frac{b}{2} \ln 2$. This is true for any finite value of k. It implies that the squared error converges more rapidly than n^{-1}.

References

[1] (Cha 66) Chaitin, G.W, "On the Length of Programs for Computing Finite Binary Sequences," *Journal of the Assoc. of Comp. Mach.*, 13, pp. 547–569, 1966.

[2] (Cov 74) Cover, T.M. "Universal Gambling Schemes and the Complexity Measures of Kolmogorov and Chaitin," Rep. 12, Statistics Dept., Stanford Univ., Stanford, Ca, 1974.

[3] (Cov 96) Cover, T.M. and Ordentlich, E. "Universal Portfolios with Side Information," *IEEE Trans. on Information Theory*, Vol 42, No. 2, pp. 348–363, March 1996.

[4] (Dea 88) Dean, Thomas and Boddy, "An Analysis of Time-Dependent Planning," *Proceedings of the Seventh National Conference on Artificial Intelligence*, pp 49-54, Minneapolis Minnesota, 1988.

[5] (Goo 51) Goodman, N. *The Structure of Appearance*, Harvard University Press, Cambridge, 1951.

[6] (Kol 65) Kolmogorov, A.N. "Three Approaches to the Quantitative Definition of Information." *Problems Inform. Transmission*, 1(1): 1–7, 1965

[7] (Las 96) Laskey, K.B. "Model Uncertainty: Theory and Practical Implications" *IEEE Trans. on Systems, Man, and Cybernetics*, Vol 26, No. 3, pp. 340–348, May 1996.

[8] (Liv 93) Li, M. and Vitányi, P. *An Introduction to Kolmogorov Complexity and Its Applications*, Springer-Verlag, N.Y., 1993.

[9] (Mar 66) Martin-Löf, P. "The Definition of Random Sequences," *Information and Control*, 9:602–619, 1966.

[10] (Ris 78) Rissanen, J. "Modeling by the Shortest Data Description," *Automatica*, 14:465–471, 1978.

[11] (Ris 95) Rissanen, J. "Stochastic Complexity in Learning," in Vitányi, Paul (ed.), *Computational Learning Theory, Second European Conference, EuroCOLT '95*, Springer- Verlag, Berlin, 1995.

[12] (Sch 95) Schmidhuber, J.H. "Low-Complexity Art" Technical Report FKI–197–94 (revised), Fakultät für Informatik, Technische Universität München, 1995.

[13] (Sol 60) Solomonoff, R.J. "A Preliminary Report on a General Theory of Inductive Inference," Report V- -131, Zator Co., Cambridge, Mass., Feb. 4, 1960.

[14] (Sol 64) Solomonoff, R.J. "A Formal Theory of Inductive Inference," *Information and Control*, Part I: Vol 7, No. 1, pp. 1–22, March 1964.

[15] (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.

[16] (Wal 68) Wallace, C.S and Boulton, D.M. "An Information Measure for Classification," *Computing Journal*, 11:185–195, 1968.

[17] (Wil 70) Willis, D.G. "Computational Complexity and Probability Constructions," *Journal of the Assoc. of Comp. Mach.*, pp. 241–259, April 1970.

[18] (Zvo 70) Zvonkin, A.K.,and Levin, L.A., "The Complexity of Finite Objects and the Development of the Concepts of Information and Randomness by Means of the Theory of Algorithms," *Russ. Math. Survs*, Vol. 25, No. 6, pp. 83- -124, 1970.