Diverse Applications of Algorithmic Probability Theory

Eray Özkural

Bilkent University

Abstract. We reminisce and discuss applications of Algorithmic Probability Theory to a wide range of problems in Artificial Intelligence, philosophy and technological society. We argue that Solomonoff has effectively axiomatized the field of Artificial Intelligence, therefore establishing it as a rigorous scientific discipline. We also relate to our own work in incremental machine learning and progress in the open AI problems which Solomonoff has defined.

1 Introduction

In this paper, we reminisce and discuss applications of Algorithmic Probability Theory (ALP) to a wide range of problems in Artificial Intelligence (AI), philosophy and technological society. We argue that Solomonoff has effectively axiomatized the field of Artificial Intelligence, therefore establishing it as a rigorous scientific discipline. We also relate to our own work in incremental machine learning and progress in the three major open problems in AI which Solomonoff has defined [1, Section 11].

Let M be a reference machine which corresponds to a probabilistic universal computer with a prefix-free code. By probabilistic universal computer, we mean a universal computer that has access to a true random number generator. In a prefix-free code, no code is a prefix of another. This is also called a self-delimiting code, and most reasonable computer programming languages are self-delimiting. Given such a machine, Solomonoff inquired the probability that an output string x is generated by M considering the whole space of possible programs. By giving each program bitstring p an a priori probability of $2^{-|p|}$, we can ensure that the space of programs meets the probability axioms (by the Kraft inequality). In other words, we imagine that we toss a fair coin to generate each bit of a random program. This probability model entails the following probability mass function (pmf) for strings $x \in \{0,1\}^*$:

$$P_M(x) = \sum_{M(p)=x} 2^{-|p|} \tag{1}$$

therefore assigning every string a certain probability. $P_M(x)$ may be called *algorithmic probability of* x because it assumes the *definition* of program based probability. We use P when M is clear from the context to avoid clutter.

2 Solomonoff Induction

Using this probability model of bitstrings, one can make predictions. Intuitively, we can state that it is impossible to imagine intelligence in the absence of any prediction ability: purely random behavior is decisively non-intelligent. Since, P is a universal probability model, it can be used as the basis of universal prediction, and thus intelligence. Perhaps, Solomonoff's most significant contributions were in the field of AI, as he envisioned a baby machine that can learn anything from scratch. Reviewing his early papers such as [2,3], we see that he has founded the complete groundwork for machine learning and data mining fields. Although much weaker methods have been popular in those fields, few other researchers could make claims about general intelligence. Unfortunately, few of his ideas have yet found fruition in practice; yet there is little doubt that his approach was the correct basis for a *science* of intelligence rather than the ad-hoc approaches of the days in which he started his patient and meticulous work.

His main proposal for machine learning is inductive inference [4,5], for a variety of problems such as sequence prediction, set induction, operator induction and grammar induction [6]. Without loss of generality, we can discuss sequence prediction on bitstrings. Assume that there is a *computable* probability mass function of bitstrings P_1. Given a bitstring x drawn from P_1 (which can be generated by Monte Carlo methods), Solomonoff states that we can predict the next bit by finding the most probable program that produces x and let it generate another bit. Finding the shortest program in general is *undecidable*, however, Levin search [7] can be used for this purpose. There are two important results about Solomonoff induction that we shall mention here. First, Solomonoff induction converges very rapidly to the real probability distribution. The convergence theorem states that the expected total square error is related to only the algorithmic complexity of P_1, which is independent from x. The following bound [1] is discussed at length in [8] with a concise proof:

$$E_P([(P_M(a_{m+1}=1|a_1,a_2,...,_a\, m)-P_1(a_{m+1}=1|a_1,a_2,...,_a\, m))^2]) \leq -\frac{1}{2}lnP_M(P_1)) \tag{2}$$

This bound characterizes the divergence of the ALP solution from the real probability distribution P_1. $P(P_1)$ is the a priori probability of P_1 pmf according to our universal distribution P_M. $-lnP_M(P_1)$ is roughly $kln2$ where k is the Kolmogorov complexity of P_1, thus the entire error is bounded by a constant, which guarantees that the error decreases very rapidly as example size increases. Secondly, there is an optimal search algorithm to approximate Solomonoff induction, which is an application of Levin Search to the kind of prediction problem outlined [7,9]. Levin Search time-shares all candidate programs according to their a priori probability with a clever watchdog policy [9]. It starts with a time limit $t = t_0$, tries all candidate programs c with that would run in $t.P(c)$ exceeding a time quantum, and while a solution is not found, it doubles the time limit t in the next iteration of search. The time $t(s)/P(s)$ for a solution program s taking time $t(s)$ is called the Conceptual Jump Size, and it has been shown that

Levin Search terminates in at most $2.CJS$. This result too is a straightforward application of ALP that we will not discuss here (as well as its optimal order of complexity), but more discussion can be found in [7,9,10].

3 The axiomatization of Artificial Intelligence

We believe in fact that Solomonoff's work was seminal in that he has single-handedly *axiomatized* AI, discovering the sufficient and necessary conditions for any machine to attain general intelligence.

Informally, these axioms are

AI1 AI must have in its possession a universal computer M (Universality).
AI2 AI must be able to learn any solution expressed in M's code (Learning recursive solutions).
AI3 AI must use probabilistic prediction (Bayes' theorem).
AI4 AI must embody in its learning a principle of induction (Occam's razor).

While it may be possible to give a more compact characterization, these are ultimately what is necessary for the kind of general learning that Solomonoff induction achieves. ALP can be seen as a complete formalization of Occam's razor [11] and thus serve as the foundation of general purpose (universal) induction, capable of solving all AI problems of significance. The axioms are important because they allow us to assess whether a system is capable of general intelligence or not.

4 Incremental Machine Learning

Levin's universal search algorithm has been adopted for solving the problem of universal induction. This algorithm is based on time-sharing all problems according to their a priori probability as mentioned above. To avoid the practical impact of the undecidability of the halting problem, the said procedure time-shares all candidate programs within a time-limit, and then doubles the time-limit at each iteration until the solution is found. Levin's universal search algorithm (and its variant discussed here) has an optimal order of complexity. In solving a problem of induction, these methods suffer from the huge computational complexity of trying to compress the entire input sequence. For instance, if the complexity of the pmf P_1 is about 400 bits, Levin search would take on the order of 2^{400} times the running time of the solution program, which is infeasible (quite impossible in observed universe). Therefore, Solomonoff has suggested using an *incremental* machine learning algorithm, which can re-use information found in previous solutions.

The following argument illustrates the situation more clearly. Let P_1 and P_2 be the pmf's corresponding to a training sequence of two induction problems (any of them, not necessarily sequence prediction, to which others can be reduced easily) with data $< d_1, d_2 >$. Assume that the first problem has been solved with

Levin search. It has taken at most $2.CJS_1 = 2.t(s_1)/P(s_1)$ time. If the second problem is solved in an *incremental* fashion, making use of the information from P_1, then the running time of discovering a solution s_2 for d_2 reduces, depending on the success of *information transfer* across problems. Here, we quantify how much in familiar probabilistic terms.

In [8], Solomonoff describes an information theoretic interpretation of ALP, which suggests the following entropy function:

$$H^*(x) = -lgP(x) \tag{3}$$

This entropy function has perfect sub-additivity of information according to the corresponding conditional entropy definition:

$$P(y|x) = \frac{P(x,y)}{P(x)} \tag{4}$$

$$H^*(y|x) = -lgP(y|x) \tag{5}$$

$$H^*(x,y) = H^*(x) + H^*(y|x) \tag{6}$$

This definition of entropy thus does not suffer from the additive constant terms as in Chaitin's version. We can instantly define mutual entropy:

$$H^*(x:y) = H^*(x) + H^*(y) - H^*(x,y) = H^*(y) - H^*(y|x) \tag{7}$$

which trivially follows.

A KUSP machine is briefly a universal computer that can store data and methods from previous experience. In 1984, Solomonoff observed that KUSP machines are especially suitable for incremental learning. In our work [12,13] we found that, using a KUSP machine was indeed useful (as in OOPS[14]). Here is how we interpreted incremental learning. After each induction problem, the pmf P is updated, thus for every new problem a new probability distribution is obtained. Although we are using the same M reference machine for trial programs, we are referring to *implicit* KUSP machines which store information about the experience of the machine so far, in subsequent problems. In our example of two induction problems, let the updated P be called P', naturally there will be an update procedure which takes time $t_u(P, s_1)$: the time of update operation given P and solution s_1. Just how much time can we expect to save if we use incremental learning instead of independent learning? First, let us write the time bound $2.t(s)/P(s)$ as $t(s).2^{H^*(s)+1}$. If s_1 and s_2 are not algorithmically independent, then $H^*(s_2|s_1)$ is smaller than $H^*(s_2)$. Independently, we would have $t(s_1).2^{H^*(s_1)+1} + t(s_2).2^{H^*(s_2)+1}$, together, we will have, in the best case $t(s_1).2^{H^*(s_1)+1} + t(s_2).2^{H^*(s_2|s_1)+1}$ for the search time, assuming that recalling s_1 takes no time for the latter search task. This would be the case, for instance, if it were an extension of a previous solution (somewhat similar to OOPS). Therefore in total, we would be saving $t(s_2).2^{H^*(s_1:s_2)+1} - t_u(P, s_1)$ in the best case (which is *unlikely* since we did not account for recall time). Note that the maximum temporal gain is related to both how much mutual information is discovered across solutions (and consequently P_i's), and how much time the update procedure

takes. Clearly, if the update time dominates overall, incremental learning is in vain. However, if updates are effective and efficient, there is enormous potential in incremental machine learning.

During the experimental tests of our Stochastic Context Free Grammar based search and update algorithms [13], we have observed that in practice we can realize fast updates, and we can still achieve actual code re-use and tremendous speed-up. Using only 0.5 teraflop/sec of computing speed and a reference machine choice of R5RS Scheme, we were able to solve 6 simple operator induction problems in 245.1 seconds. Scaled to human-level processing capacity of 100 teraflop/sec, this would mean that our system could learn and solve the entire training sequence in 1.25 seconds, which is (arguably) better than most human students. This running time is compared to 7150 seconds without any updates. In one particular operator induction problem (fourth power, x^4), we saw actual code re-use: `(define (pow4 x ) (define (sqr x ) (* x x)) (sqr (sqr x ) ))`, and an actual speedup of 272. The gains that we saw confirmed the incremental learning proposals of Solomonoff, mentioned in a good number of his publications, but most clearly [9,10,15]. Based on our work and OOPS [14], we have come to believe that incremental learning has the epistemological status of an additional AI axiom:

AI5 AI must be able to use its previous experience to speed up subsequent prediction tasks (Transfer Learning).

This axiom is justified by observing that many induction problems are completely unsolvable by a system that does not have the adequate sort of *algorithmic* memory.

We should also account our brief correspondence with Solomonoff. We expressed that the algorithms were very powerful but it seemed that too little memory was used. Solomonoff responded by mentioning the potential stochastic grammar and genetic programming approaches. Our present research was motivated by a problem he posed during the discussions of his 2006 seminars in Istanbul: "We can use grammar induction for updating a stochastic context free grammar, but there is a problem. We already know the grammar of the reference machine.". The 2009 implementation of gigamachine algorithms [12] were designed in response to this problem in late 2006. Solomonoff has also guided our research by making a valuable suggestion, that it is not so significant to solve a difficult problem by spending lots of supercomputer time, but it is important to show whether incremental learning works over a sequence of simpler problems.

5 Cognitive Architecture

Another important discussion is whether a cognitive architecture is necessary. The axiomatic approach was seen counter-productive by some leading researchers in the past. However, we think that their opinion can be expressed as follows: the minimal program that realizes these axioms is not automatically intelligent,

because in practice an intelligent system requires a good deal of algorithmic information to take off the ground. This is not a bad argument, since obviously, the human brain is well equipped genetically. However, we cannot either rule out that a somewhat compact system may achieve human-level general intelligence. The question therefore, is whether a simply described system like AIXI(t,s) [16] is sufficient *in practice*, or is there a need for a modular/extensible cognitive architecture that has been designed in particular ways to promote certain kinds of mental growth and operation. Some proponents of general purpose AI research think that such a cognitive architecture is necessary such as OpenCog [17]. Schmidhuber has suggested the famous Godel Machine which has a mechanical model of machine consciousness [18]. Solomonoff himself has proposed early on in 2002, the design of Alpha, a generic AI architecture which can ultimately solve free-form time-limited optimization problems [10]. Although in his later works, Solomonoff has not made much mention of Alpha and has instead focused on the particulars of the required basic induction and learning capability, nonetheless his proposal remains as one of the most extensible and elegant self-improving AI designs. Therefore, this point is open to debate, though some researchers may want to assume another, entirely optional, axiom:

AI6 AI must be arranged such that self-improvement is feasible in a realistic environment (Cognitive Architecture).

It is doubtful for instance whether a combination of incremental learning and time/space bounded $AIXI$ will result in a practical reinforcement learning agent. Neither it is well understood whether autonomous systems with built-in utility/goal functions are suitable for all practical purposes. We anticipate that such questions will be settled by experimenters, as the complexity of interesting experiments will quickly overtake the analysis that can be furthered in theoretical papers.

We presented these important capabilities as axioms, so that they may be referred to more easily, as in subscribing to one axiom and rejecting another in a particular AI system.

6 Foundations of Mathematics

An unexpected consequence of the algorithmic probability approach is the halting probability of a random program of M, which is denoted as Ω_M. Chaitin has shown that Ω_M has an infinite amount of information, and thus it corresponds to a strong definition of mathematical randomness. Therefore, ALP does not merely build upon the probability axioms, but also serves as a foundation for probability theory itself. Furthermore, it is well known that classical information theory can be derived from algorithmic information theory, further clenching the epistemological status of ALP and AIT (which are converse ways of looking at the same problem).

An interesting question is whether ALP encourages a realist or a constructivist attitude towards mathematics. No theorem in ALP requires accepting

constructively unjustifiable claims, therefore we think that ALP is compatible with constructivism. This is most strikingly viewed in the question of whether Ω exists. It is true that the bits of Ω are both certain and are essentially the same when we change the reference machine. This has led some thinkers to assume a Platonist position with regards to Ω. However, we think that position is unjustified, for the approximation procedure for Ω suggests that novel algorithmic information can be generated by spending adequate physical resources (time, energy, space). Therefore, we suggest that algorithmic information evolves; it has not been created before evolution. The certainty of machine description must not be confused with independent existence.

7 Intellectual Property Towards Infinity Point

Solomonoff has proposed that infinity point, also known as the singularity, is the result of accelerated progress caused by trans-human AI's, to accelerate our progress even further ad infinitum [19]. Solomonoff has proposed five milestones of AI development: A: modern AI phase (1956 Dartmouth conference), B: general theory of problem solving (our interpretation: Solomonoff Induction, Levin Search), C: self-improving AI (our interpretation: Alpha architecture, 2002), D: AI that can understand English (our interpretation: not realized yet), E: human-level AI, F: an AI at the level of entire computer science (CS) community, G: an AI many times smarter than entire CS community.

A weak condition for infinity point may be obtained by an economic argument, also covered in [19] briefly. The human brain produces 5 teraflops/watt roughly. Current NVIDIA GPGPU architectures achieve about 6 gflops/watt. Assuming 85% improvement in power efficiency per year, in 12 years, the human-level computing power-efficiency will be achieved. After that date, even if AI fails, we will be able to run our "uploads" faster than us, using less energy than humans, effectively creating a bioinformation based AI which meets the basic requirement of infinity point. This weaker condition rests on an economic observation: the economic incentive of cheaper intellectual work will drive the proliferation of personal use of uploads. According to NVIDIA's own projections, thus, we can expect the necessary conditions for the infinity point to materialize by 2023, after which point technological progress will accelerate very rapidly.

Assume that we are progressing towards infinity point. Then, the entire human civilization may be viewed as a global intelligence working on technological problems (And from another perspective, the whole evolution may be viewed as computing the bits of Ω_M as in Chaitin's work). The practical necessity of incremental learning suggests that when faced with more difficult problems, better information sharing is required. If no information sharing is present between researchers (i.e., different search programs) then, they will lose time traversing overlapping program subspaces. This is most clearly seen in the case of *simultaneous inventions* when an idea is said to be "up in the air" and is invented by different parties on near dates. If, intellectual property laws are too rigid and costly, this would entail that there is minimal information sharing, and after

some point, the global efficiency of solving non-trivial technological problems would be severely hampered. Therefore, to utilize the infinity point effects better, knowledge sharing must be encouraged in the society. Maximum efficiency in this fashion can be provided by free software licenses, and a reform of the patent system. The conclusion is that no single company can (or should) have a monopoly on the knowledge resources to attack problems with truly large algorithmic complexity. Therefore, sharing of facts and technology is the most efficient path towards the infinity point (i.e., singularity).

8 Conclusion

We have mentioned diverse applications of ALP in artificial intelligence, axiomatization of AI, philosophy of mathematics and technological society. We have related our own research to Solomonoff's open AI problems. We interpret ALP as a fundamentally new world-view which allows us to approach a plethora of complex subjects more scientifically, bridging the gap between complex natural phenomena and positive sciences more closely than ever. This paradigm shift has resulted in various breakthrough applications and is likely to benefit the society in the foreseeable future.

References

1. Solomonoff, R.J.: Algorithmic probability: Theory and applications. In Dehmer, M., Emmert-Streib, F., eds.: Information Theory and Statistical Learning, Springer Science+Business Media, N.Y. (2009) 1–23
2. Solomonoff, R.J.: An inductive inference machine. Privately circulated report (1956)
3. Solomonoff, R.J.: An inductive inference machine. In: IRE Convention Record, Section on Information Theory, Part 2. (1957) 56–62
4. Solomonoff, R.J.: A formal theory of inductive inference, part i. Information and Control **7**(1) (1964) 1–22
5. Solomonoff, R.J.: A formal theory of inductive inference, part ii. Information and Control **7**(2) (1964) 1–22
6. Solomonoff, R.J.: Three kinds of probabilistic induction: Universal distributions and convergence theorems. The Computer Journal **51**(5) (2008) 566–570
7. Levin, L.A.: Universal sequential search problems. Problems of Information Transmission **9**(3) (1973) 265–266
8. Solomonoff, R.J.: Complexity-based induction systems: Comparisons and convergence theorems. IEEE Trans. on Information Theory **IT-24**(4) (1978) 422–432
9. Solomonoff, R.J.: Optimum sequential search (1984)
10. Solomonoff, R.J.: A system for incremental learning based on algorithmic probability. In: Proceedings of the Sixth Israeli Conference on Artificial Intelligence, Tel Aviv, Israel (1989) 515–527
11. Hutter, M.: Algorithmic randomness as foundation of inductive reasoning and artificial intelligence. CoRR **abs/1102.2468** (2011)
12. Özkural, E.: Gigamachine: incremental machine learning on desktop computers. `http://examachine.net/papers/gigamachine-draft.pdf` (2009) Draft.

13. Özkural, E.: Teraflop-scale incremental machine learning. CoRR **abs/1103.1003** (2011) `http://arxiv.org/abs/1103.1003`.
14. Schmidhuber, J.: Optimal ordered problem solver. Machine Learning **54** (2004) 211–256
15. Solomonoff, R.J.: Progress in incremental machine learning. In: NIPS Workshop on Universal Learning Algorithms and Optimal Search. (2002)
16. Hutter, M.: Universal algorithmic intelligence: A mathematical top→down approach. In Goertzel, B., Pennachin, C., eds.: Artificial General Intelligence. Cognitive Technologies. Springer, Berlin (2007) 227–290
17. Goertzel, B.: Opencogprime: A cognitive synergy based architecture for artificial general intelligence. In Baciu, G., Wang, Y., Yao, Y., Kinsner, W., Chan, K., Zadeh, L.A., eds.: IEEE ICCI, IEEE Computer Society (2009) 60–68
18. Schmidhuber, J.: Ultimate cognition *à la* Gödel. Cognitive Computation **1**(2) (2009) 177–193
19. Solomonoff, R.J.: The time scale of artificial intelligence: Reflections on social effects. Human Systems Management **5** (1985) 149–153