Differences between Kolmogorov Complexity and Solomonoff Probability: Consequences for AGI

Alexey Potapov[1], Andrew Svitenkov[2], Yurii Vinogradov[2]

[1]AIDEUS, Russia
[2]National Research University of Information Technology Mechanics and Optics, Russia
potapov@aideus.com

Abstract. Kolmogorov complexity and algorithmic probability are compared in the context of the universal algorithmic intelligence. Accuracy of time series prediction based on single best model and on averaging over multiple models is estimated. Connection between inductive behavior and multi-model prediction is established. Uncertainty as a heuristic for reducing the number of used models without losses of universality is discussed. The conclusion is made that plurality of models is the essential feature of artificial general intelligence, and this feature should not be removed without necessity.

Key words: Universal Agents, Kolmogorov Complexity, Algorithmic Probability, Prediction, Inductive Behavior, Uncertainty

1 Introduction

Solomonoff Algorithmic Probability (ALP) theory of prediction is known to be ideal and universal. Unsurprisingly, it became the main theoretical basis for the models of artificial general intelligence [1, 2]. However, computing algorithmic probabilities implies summation over all possible algorithmic models (programs). Naturally, the two-part Minimum Message Length (MML) or Minimum Description Length (MDL) principles are adopted instead of ALP while developing practically applicable methods of machine perception and learning. These principles also rely on the algorithmic information theory (namely, on Kolmogorov complexity), but they give criteria for selecting single best models in inductive inference tasks. The best model is assumed to be the model that minimizes the sum of the complexity of the model, and the length of the data encoded given this model. These principles are frequently called information-theoretic formalizations of Ockham's Razor, which simplified formulation states that plurality should not be assumed without necessity. The MDL and MML principles are usually treated as the practical approximations of ALP [3]. Even those authors, who utilize ALP in the models of universal agents, refer to Ockham's Razor [1] mixing ALP and MDL in spite of the fact that ALP implies plurality of models.

Besides the practical arguments some authors also claim that the MML (or MDL) principle is much more methodologically appropriate for intelligent agents. In particular, importance of the two-part coding (lossy compression) is pointed out in [4] in the context of multi-agent systems (social environments). Indeed, one can agree that agents should exchange only the first parts of MML messages (models or regularities) with each other, because there is no need to communicate noise. Apparently, social communications are better described by the MML principle than by the ALP theory of prediction. Even optimal prediction methods should really be based on ALP, it is said that ALP gives better results than MML or MDL if many the top models have similar quality [2, 4, 5]. Even 10 bit difference between models makes their probabilities incomparable. It can be seen that there are serious reasons to give up on ALP.

On the other hand, there is also the opinion that human brain prefers to describe observations in many different ways, and it is unlikely that some single model of the world is used. Such redundancy of descriptions contradicts Ockham's Razor [6]. It is also interesting to note that different compositions and mixtures of experts became quite popular in the field of pattern recognition. Their efficiency appeared to be somewhat surprising, because mixture models are very complex and should be subjected to overlearning as it follows from the MDL (MML) principle. In our opinion, these issues can be resolved within ALP.

In this paper, we analyze differences between algorithmic probability and Kolmogorov complexity in the context of the models of universal algorithmic intelligent agents. We argue that ALP not only ensures optimal prediction, but also allows for some essential features of intelligent behavior. In particular, inductive (or knowledge-seeking) behavior can naturally emerge only from consideration of many alternative models. Of course, the mentioned computational and communicational restrictions are valid, but it doesn't mean that one should simply reduce the number of models taken into account. We believe that models should not be just thrown out, but they should be united into some sets leading to uncertain models. That is, the notion of uncertainty absent in the resource-unlimited universal algorithmic intelligent agents originates from the necessity to account for many models while reasoning and communicating with limited resources and time.

These conclusions are illustrated with some particular models of time series forecasting and intelligent agent behavior in Markov environment.

2 Comparison of prediction quality

Consider the notion of algorithmic probability. The algorithmic probability $P_{ALP}(x)$ of some string x is defined as:

$$P_{ALP}(x) = \sum_{q:U(q)=x} 2^{-l(q)} , \tag{1}$$

where U is the Universal Turing Machine, each q is its program, which produces x and has the length $l(q)$.

At the same time, the Kolmogorov complexity $K(x)$ is defined as:

$$K(x) = \min_{q:U(q)=x} l(q) . \tag{2}$$

Formally, it is obvious that $-\log_2 P_{ALP}(x)<K(x)$. However, Kolmogorov complexity implies that there is the smallest program, which can be used as the most compact description of x and can be sent instead of the original data, while ALP doesn't provide us with an effective compression scheme. Thus, Kolmogorov complexity is the more natural basis to introduce the two-part coding separating models from noise:

$$K(x) = \min_{q:U(q)=x} l(q) = \min_{\mu\delta:U(\mu\delta)=x} (l(\mu)+l(\delta)) = \min_{\mu}(l(\mu)+K(x\,|\,\mu)), \tag{3}$$

where μ is interpreted as the model, and δ is interpreted as noise.

As the result, one can choose the best model μ yielding the minimum description length. This separation can also be performed in the case of ALP, but its meaning will be more vague. Actually, it is somewhat heuristic also in the case of Kolmogorov complexity, but it appears to be rather natural in each specific case.

Now, let's consider separately the task of prediction. Solution of this task can be based on the conditional algorithmic probability and the conditional algorithmic complexity defined as:

$$P_{ALP}(x\,|\,y) = \sum_{q:U(qy)=x} 2^{-l(q)}, \; K(x\,|\,y) = \min_{q:U(qy)=x} l(q). \tag{4}$$

Of course, algorithmically complete solutions are now unachievable both for Kolmogorov complexity and ALP. Thus, we compare them on the restricted subset of algorithms specified by the dynamical artificial neural networks (DANNs). Each DANN can be described by the corresponding system of differential equations:

$$x_i'(t) = \frac{dx_i(t)}{dt} = f\left(\sum_{j=1}^{M} w_{ji}x_j(t)\right), \tag{5}$$

where x_i are activities of M neurons, w_{ji} are connection weights constituting a matrix $\mathbf{W}$, and f is an activation function.

Starting from some initial values $x_i(0)$, activities $x_i(t)$ will evolve producing some functions as an output. One interesting application is the time series forecasting, in which the data $D=\{\mathbf{y}(t_1),\ldots,\mathbf{y}(t_n)\}$ is given, where the values $\mathbf{y}(t_i)=(y_1(t_i),\ldots, y_N(t_i))$ of the N-dimensional vector are observed at some moments of time $t_i\in[0, T_{\max}]$. The task is to predict values $\mathbf{y}(t)$ for $t>T_{\max}$.

Such connection weights w_{ij} and such initial activities $x_i(0)$ should be found that the activities $x_i(t)$ are most precisely correspond to the values $y_i(t)$. Naïve approach leads to minimization of the mean-square error:

$$E^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{N}\left[y_j(t_i)-x_j(t_i)\right]^2. \tag{6}$$

The number of neurons M should be not less than the dimension N of the vector $\mathbf{y}$, but it can be larger. In this case, additional neurons can be treated as hidden dynamic variables. They are not included into the MSE criterion (6). Apparently, increase of the number of additional neurons will result in decrease of the MSE as well as in overfitting. In accordance with the MDL principle, the model complexity should also be taken into account in addition to the description length of the data encoded within the model that can be estimated as $nN\log_2 E$ (accurate to a constant). Here, one can see benefits of the two-part coding.

The ANN model description includes information about the number of neurons, established connections, their weights, and initial values of activity. Total MDL criterion for the ANN with M neurons and K connections requiring $\log_2 \sqrt{n}$ bits per parameter can be roughly estimated as:

$$L = nN\log_2 E + \log_2 M + \log_2 K + \log_2 C_{M^2}^{K} + 0.5(M+K)\log_2 n\,. \qquad (7)$$

To find the best ANN, one should consider and optimize ANNs with different number of neurons and connections. In order to reduce computational complexity of this process, we utilized an iterative scheme, in which new neurons are consequently added and redundant connections are removed if these operations result in reduction of the description length criterion (7). We considered and implemented a combination of several optimization techniques (stochastic gradient descent, genetic algorithms, and simulated annealing) for optimizing ANNs with fixed architecture.

While searching for the solution with the minimum description length, many other ANNs are generated. In any case, extrapolations of the given time series are calculated using these ANNs. Why don't we try computing average result of prediction for all these ANNs taken with weights proportional to 2^{-L} (actually, ALP implies averaging over probabilities, but here averaging over predictions also works)? We will refer to such the plural model as "P-model" (P stands for algorithmic probability). The best found model will be referred to as "K-model" (K stands for Kolmogorov complexity). Let's compare prediction precision for K-models and P-models on some specific data.

Consider the well-known Wolf annual sunspot time series (see [7] as an example of application of the MDL-based ANN learning). We used the Wolf numbers till 1979 as the training sample. The search algorithm was launched for several times. Table 1 shows the result for 3 best runs (K-models and P-models assigned the same indices were obtained during the same runs). MSE_{int} stands for the MSE on the training sample, MSE_9 and MSE_{22} stand for the prediction MSE for 1980–1988 years and 1980–2001 years correspondingly.

Table 1. Comparison of prediction precision for some P-models and K-models

Model	L or $-\log_2 P$	MSE_{int}	MSE_9	MSE_{22}
K-model #1	798.9	398	900	4010
P-model #1	790.6	382	795	3078
K-model #2	799.0	388	904	3359
P-model #2	789.6	369	815	2926
K-model #3	796.7	382	907	3956
P-model #3	789.4	383	875	3705

It can be seen that prediction precision of the K-models is usually worse than of the corresponding P-models, although the optimization procedure wasn't specially designed to search for alternative models with close weights. Actually, corresponding K- and P-models produce functions with similar shape meaning that primarily the best K-model and some nearby models influence the P-models. It is interesting to merge different P-models (in order to merge two P-models, one should simple calculate averaged prediction using corresponding weights, and sum probabilities of these models). One can consider even P-models belonging to different model spaces.

To check this idea the P-model #4 was found using another activation function representing another subset of algorithms. This model has $–\log_2 P = 784.5$; $\text{MSE}_{\text{int}}=204$; $\text{MSE}_9=834$; $\text{MSE}_{22}=529$. Table 2 shows the prediction precision of the consequently merged P-models.

Table 2. MSE values for the merged P-models

Model	MSE_{int}	MSE_9	MSE_{22}
P-model #4	204	834	529
P-model #4+1	204	820	521
P-model #4+1+2	204	796	510
P-model #4+1+2+3	205	769	506

In this case, the final P-model showed the best prediction accuracy. Examples of the K- and P-model predictions are given on Fig. 1.

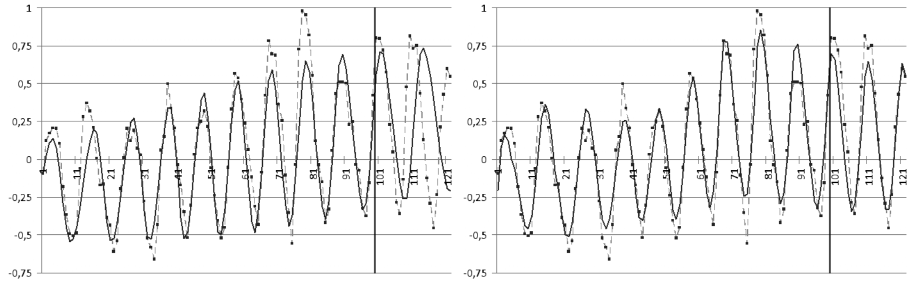

Fig. 1. Initial data (*dotted curves*) and reconstructed time series with the K-model #1 (*left*) and the merged P-model (*right*)

More interesting (but less reproducible) results can be obtained on such non-stationary data as financial time series. An example of such time series extrapolated with three best P-models (found on separate runs of the search algorithm) and the merged P-model are shown on Fig. 2. This is the case, when several the top models have similar weights, but give absolutely different predictions. 50 points ahead forecasting MSE for these models is given in Table 3.

Table 3. MSE values for the P-models

Model	#1	#2	#3	#1+#2	#1+#2+#3
MSE_{int}	0.0263	0.0258	0.0270	0.0250	0.0251
MSE_{50}	0.157	0.264	0.097	0.146	0.067

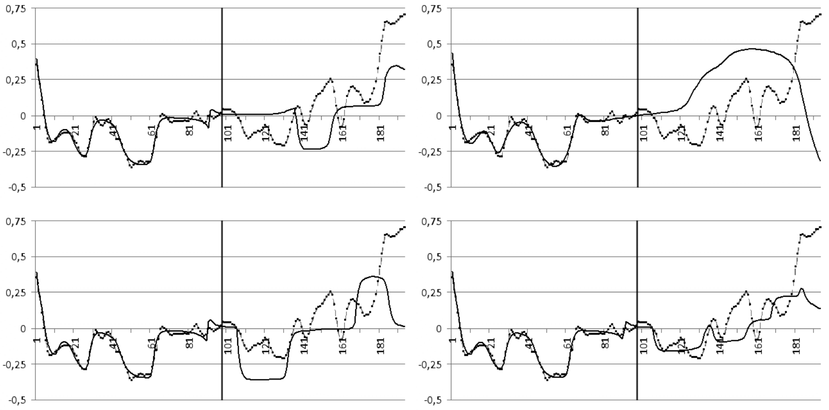

Fig. 2. Three initial P-models and the merged P-model (*bottom right*)

The shown prediction quality increase for the merged P-models is rather frequent. Of course, the prediction quality of a merged model is not always better than the quality of both models before merging. Sometimes it lies between them meaning that the quality of the merged model is worse than the quality of one of the models. However, the final P-model is almost always better than the best K-model. This is why different "mixtures of experts" in machine learning appeared to be so useful.

It should be pointed out that this increase of prediction quality is achieved almost without additional computation costs. Also, two-part coding was rather naturally used with the plural model prediction derived from ALP. At the same time, further usage of the plural models can be indeed computationally costly, e.g. in sequential decision making or in multi-agent communications.

3 Inductive behavior

The disputable question is whether reinforcement learning is the appropriate framework for generally intelligent agents or not. Will the universal agent, which simply tries to maximize rewards received from the environment, show all types of behavior typical for humans? Here, we don't try to give a complete answer to this question. Instead, we focus on a specific behavior, namely the inductive behavior (knowledge seeking or active learning).

Different authors have considered necessity to extend (or even replace) the reward based utility function with the term expressing increase of agent's knowledge about environment. Then, the agent will be curious and will try to obtain new information. The reinforcement-learning agent has no direct motivation for inductive behavior.

Authors of [8] even claim that if this agent is allowed to arbitrarily modify its own inputs, it will do so. They call this situation the "delusion box". That is, the agent will prefer to live in illusion maximizing his utility function without obtaining information about the real world. However, the reinforcement-learning agent will choose to use the delusion box only if it will be able to predict that this choice will increase integral future reward taking into account predicted lifespan. If the agent is based on ALP, there will be models with non-zero probability predicting shorter lifespan in the case of the delusion box. Thus, the expected rewards will not be the highest possible, and the choice will depend on circumstances. For example, if the agent expects near death, it may try to use the delusion box.

On the other hand, if the agent uses only the best model for prediction, it will immediately use the delusion box (and ignore the real world), when probability of the lifespan decrease is lower. Consequently, one may suggest that inductive behavior in general can be derived from sequential decision making with ALP-based prediction. Indeed, if the agent refines predictions on each step of sequential decision making depending on the hypothesized answer of the environment, it will "automatically" account for the benefits of knowledge acquisition. Of course, one can also agree that "additional" explicit bias towards exploring previously unknown environmental regularities can be a useful heuristic [9].

Difference in the agent's behavior depending on usage of a single or multiple models can be experimentally checked on the example of the simplest Markov environment. Let environment be described by some probability distribution $P(x'|x, y)$, where x is the previous state of the environment, x' is the current state, and y is the last agent's action. We can even consider fully observable environments.

The agent tries to estimate the model of the world in the form of the distribution $P^*(x'|x, y)$ on the base of the history $xy_{\leq t}$. Obviously, the best model will be the model with probabilities simply equal to the frequencies of the corresponding transitions estimated on the base of the history, if complexities of different distributions P are assumed to be equal. When the history is empty, all the models have the same quality. Arbitrary model can be chosen depending on implementation details.

When the agent performed the action y at the state x for the first time, and this action leaded to the state x', the best model would contain $P^*(x'|x, y)=1$. Imagine that the state x appeared twice, and the agent performed actions y_1 and y_2 with the results x'_1 and x'_2. Obviously, the agent will choose the action that previously leaded to the best outcome. Situation will be more complex for sequential decision making, but the general result will be the same – the agent will choose the action that simply gave the best reinforcement in the past. Of course, the next try of this action in the given situation can lead to different states, and statistics for this action will be enriched. The agent can reject to use the action that seemed to be good on the first try, but appeared to be worse later. But this agent will not try such action that leaded to bad states unless all the other actions would be even worse. Thus, one can expect that the "single-model" agent will accumulate very inhomogeneous statistics for different actions.

On the contrary, for the Solomonoff prediction any distribution $P^*(x'|x, y)$ can be considered as a possible environment model for any history with some probability that can be easily estimated. Difference in probabilities will increase as the history length increases, but it will be small for short histories. Because prediction is based on averaging over all models, all expected reinforcements will be very similar at first. If some action was performed one or few times, its quality will be near average value, and preferences in actions will change very frequently until statistics for almost all of them are gathered.

Knowledge-seeking is “automatically” modeled in sequential decision making with the use of multi-model prediction. Indeed, some “unknown” action can have good outcome. In this case, this action will be repeated many times, and summed future outcome will be increased. The action can have bad outcome. In this case, this action will not be repeated many times, and summed future outcome will decrease only slightly. Because these both possibilities for “unknown” action have similar probabilities, it will be better in average to try such action (if there is no well-known action that has reliable outcome better than some average value). It can be seen that this agent will show knowledge-seeking behavior, when it is not “satisfied”. Of course, it may be useful to boost knowledge-seeking behavior (or even make it the main “drive”) by modifying the value function, but our goal was to show that this form of behavior naturally appears due to the multiplicity of environment models.

4 Uncertainty

As it was shown above, it is inadmissible to use the only one best model in AGI. Not only is multi-model prediction more accurate, but also it allows for such forms of behavior, which are essential for universal intelligence. At the same time, usage of too many models is practically impossible in sequential decision making and communications. Is it possible to reduce computational costs of multi-model approach without loosing its important features? We suppose that the number of models should be reduced not simply by eliminating worse models, but by uniting them into some sets.

Let’s divide the whole set of models $Q=\{q\colon U(q)=x\}$ into finite number of disjoint subsets Q_i. Thus, one can write

$$P_{ALP}(x)=\sum_{q\in Q}2^{-l(q)}=\sum_{Q_i}\sum_{q\in Q_i}2^{-l(q)}. \tag{8}$$

We want to deal with subsets of models without addressing individual models in order to reduce complexity of their further usage. The simplest way is to use the best model within a subset instead of all models in this subset:

$$P_{ALP}(x)=\sum_{Q_i}\sum_{q\in Q_i}2^{-l(q)}\geq\sum_{Q_i}2^{-\min\limits_{q\in Q_i}l(q)}\geq 2^{-\min\limits_{q\in Q}l(q)}=2^{-K(x)}. \tag{9}$$

This will be better than usage of the single best model, but still is not good enough. One needs not only to use one representative model instead some subset, but to describe the structure of this subset in more details.

To illustrate this idea, we analyze the simplest non-universal, but useful way of enriching descriptions of model subsets. Consider the subset, in which all models have the structure $q_j=\mu\pi_j\delta_j$, where μ is their common part (general model), π_j are the strings of particular parameter values, δ_j are the strings of deviations of j-th model $\mu\pi_j$ from the data x. One can write

$$\sum_{q\in Q_i} 2^{-l(q)} = \sum_{\pi_j\delta_j:\ \mu\pi_j\delta_j\in Q_i} 2^{-l(\mu)-l(\pi_j)-l(\delta_j)} = 2^{-l(\mu)}\sum_{\pi_j\delta_j} 2^{-l(\pi_j)-l(\delta_j)}. \tag{10}$$

Because all δ_j are interpreted as noise, it is not necessary to use them in prediction and decision-making. We also don't want to account for all possible values of π_j, but we are interested in the distribution:

$$P_{\mu,x}(\pi_j) = 2^{-l(\pi_j)-l(\delta_j)}. \tag{11}$$

If the set of parameters π_j constitute some metric space, one can estimate some statistical moments of this distribution. In the other case, assuming independence of distributions of each sign in π_j one can directly estimate these distributions. As the result, it is possible to represent the distribution $P_{\mu,x}(\pi_j)$ compactly. Such compact representation will contain information about uncertainty in the parameter values π of some best model from the subset Q_i.

Usage of such uncertain models allows estimating uncertainty in prediction caused by the simple fact that different models in the set Q_i produce different outputs $U(\mu\{\pi\delta\})=\{x\}$ (of course, the set of predictions $\{x\}$ cannot be known precisely unless all models are explicitly computed). More complex type of uncertainty can be considered, when one tries to reduce the number of models further uniting subsets Q_i containing models with different structures.

Uncertainty in the predicted x propagates through sequential decision making and becomes much larger in future. Obviously, if the agent has such a history that leads to models with high uncertainty, it will not be possible to guarantee high future rewards. Thus, actions aimed to decrease uncertainty will allow increasing future rewards in average. Thus, they can be chosen even in the case, when few models are used in sequential decision making, but uncertainty is taken into account.

In the case of simplest Markov environment, introduction of uncertainty leads to bias towards more uniform distribution $P^*(x'|x, y)$. Unsurprisingly, experiments show that more diverse actions are tried in presence of this bias, while the agent prefers exploitation in absence of this bias. The biased agent gains slightly smaller rewards at the beginning, but it has some chances to outperform unbiased single-model agent on long time intervals. Correct introduction of uncertainty as a heuristic in adoption of ALP can hopefully give optimal solution of the "exploration vs. exploitation" problem. This possibility has not been considered within algorithmic information theory.

It can be seen that uncertainty should be introduced as a heuristic that helps to greatly reduce computational costs of ALP without violating inductive behavior. It is frequently said that uncertainty and probability are different categories. However, theories of uncertainty usually rely on the combinatorial basis. However, if we follow Kolmogorov and Solomonoff, the notion of probability should be inferred from the notion of information, which should also have pure combinatorial (algorithmic) basis. Solomonoff induction doesn't include the notion of uncertainty, but it naturally appears in attempt to reduce the number of used models. Thus, the complete theory of uncertainty should be built on the base of the algorithmic information theory. Unfortunately, detailed analysis of this problem goes beyond the scope of the paper.

5 Conclusions

Some methodological aspects of usage of Kolmogorov complexity and algorithmic probability in universal intelligent agents were discussed. At first, the task of time series forecasting was considered. The dynamic artificial neural networks were used as a subset of algorithmic models. Accuracy of prediction given by the best ANN selected on the base of the MDL criterion was compared with accuracy of prediction derived from ALP (weighted sum of predictions made by all the models constructed during the search was calculated). MSE of the latter kind of prediction appeared to be stably lower. Decrease of MSE varied from 10% to 50% depending on data.

Then, the problem of information-seeking behavior was considered. It was shown that such inductive behavior naturally appears in the ALP-based agent, while the "single-model" agent will have a strong bias towards exploitation of actions with well-known good outcome. In order to reduce complexity of usage of multiple models in decision making and communications, subsets of models is proposed to replace with some "uncertain" models. A theory of uncertainty as one of meta-heuristics meant for considerable reduction of computational complexity of ALP without losses of universality is to be developed in future.

References

1. Hutter, M.: Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability. Springer (2005).
2. Solomonoff, R.: Algorithmic Probability, Heuristic Programming and AGI. In: Baum, E., Hutter, M., Kitzelmann, E. (eds). Advances in Intelligent Systems Research, vol. 10 (proc. 3rd Conf. on Artificial General Intelligence), pp. 151–157 (2010).
3. Solomonoff, R.: The Discovery of Algorithmic Probability. In: J. of Computer and System Sciences, vol. 55, no. 1, pp. 73–88 (1997).
4. Dowe, D., Hernández-Orallo, J., Das, P.: Compression and Intelligence: Social Environments and Communication. In: Lecture Notes in Computer Science 6830 (proc. Artificial General Intelligence – 4th International Conference), pp. 204–211. Springer (2011).
5. Poland, J., Hutter, M.: MDL convergence speed for Bernoulli sequences. In: Statistics and Computing, vol. 16, pp. 161–175 (2006).
6. Buchanan B.G.: What Do We Know about Knowledge? In: AI Magazine, vol. 26, no. 4, pp. 35–46 (2005).
7. Small, M., Tse, C.K.: Minimum Description Length Neural Networks for Time Series Prediction. In: Physical Review E. 66, 066701 (1–12) (2002).
8. Ring, M., Orseau, L. Delusion, Survival, and Intelligent Agents. In: Lecture Notes in Computer Science 6830 (proc. Artificial General Intelligence – 4th International Conference), pp. 11–20. Springer (2011).
9. Schmidhuber, J. Artificial Scientists & Artists Based on the Formal Theory of Creativity. In: Baum, E., Hutter, M., Kitzelmann, E. (eds). Advances in Intelligent Systems Research, vol. 10 (proc. 3rd Conf. on Artificial General Intelligence), pp. 145–150 (2010).