Simple Algorithmic Principles of Discovery, Subjective Beauty, Selective Attention, Curiosity & Creativity[*]

Jürgen Schmidhuber
TU Munich, Boltzmannstr. 3, 85748 Garching bei München, Germany &
IDSIA, Galleria 2, 6928 Manno (Lugano), Switzerland
juergen@idsia.ch - http://www.idsia.ch/~juergen

Abstract

I postulate that human or other intelligent agents function or should function as follows. They store all sensory observations as they come—the data is 'holy.' At any time, given some agent's current coding capabilities, part of the data is compressible by a short and hopefully fast program / description / explanation / world model. In the agent's subjective eyes, such data is more regular and more *beautiful* than other data. It is well-known that knowledge of regularity and repeatability may improve the agent's ability to plan actions leading to external rewards. In absence of such rewards, however, *known* beauty is boring. Then *interestingness* becomes the *first derivative* of subjective beauty: as the learning agent improves its compression algorithm, formerly apparently random data parts become subjectively more regular and beautiful. Such progress in data compression is measured and maximized by the *curiosity* drive: create action sequences that extend the observation history and yield previously unknown / unpredictable but quickly learnable algorithmic regularity. I discuss how all of the above can be naturally implemented on computers, through an extension of passive unsupervised learning to the case of active data selection: we reward a general reinforcement learner (with access to the adaptive compressor) for actions that improve the subjective compressibility of the growing data. An unusually large compression breakthrough deserves the name *discovery*. The *creativity* of artists, dancers, musicians, pure mathematicians can be viewed as a by-product of this principle. Several qualitative examples support this hypothesis.

[*] Joint Invited Lecture for *Algorithmic Learning Theory* (ALT 2007) and *Discovery Science* (DS 2007), Sendai, Japan (preprint). Variant to appear in: V. Corruble, M. Takeda, and E. Suzuki (Eds.): DS 2007, LNAI 4755, pp. 26-38, Springer-Verlag Berlin Heidelberg 2007. Abstract: M. Hutter, R.A. Servedio, and E. Takimoto (Eds.): ALT 2007, LNAI 4754, pp. 24-25, Springer-Verlag Berlin Heidelberg 2007; see also http://www.springerlink.com/content/y8j3453l0757m637/?p=42fb108af50a4cbf8ec06c12309884f6&pi=2 and http://www.springer.com/west/home/generic/search/results?SGWID=4-40109-22-173760307-0

1 Introduction

A human lifetime lasts about 3×10^9 seconds. The human brain has roughly 10^{10} neurons, each with 10^4 synapses on average. Assuming each synapse can store not more than 3 bits, there is still enough capacity to store the lifelong sensory input stream with a rate of roughly 10^5 bits/s, comparable to the demands of a movie with reasonable resolution. The storage capacity of affordable technical systems will soon exceed this value.

Hence, it is not unrealistic to consider a mortal agent that interacts with an environment and has the means to store the entire history of sensory inputs, which partly depends on its actions. This data anchors all it will ever know about itself and its role in the world. In this sense, the data is 'holy.'

What should the agent do with the data? How should it learn from it? Which actions should it execute to influence future data?

Some of the sensory inputs reflect external rewards. At any given time, the agent's goal is to maximize the remaining reward or reinforcement to be received before it dies. In realistic settings external rewards are rare though. In absence of such rewards through teachers etc., what should be the agent's motivation? Answer: It should spend some time on *unsupervised learning*, figuring out how the world works, hoping this knowledge will later be useful to gain external rewards.

Traditional unsupervised learning is about finding regularities, by clustering the data, or encoding it through a factorial code [2, 14] with statistically independent components, or predicting parts of it from other parts. All of this may be viewed as special cases of data compression. For example, where there are clusters, a data point can be efficiently encoded by its cluster center plus relatively few bits for the deviation from the center. Where there is data redundancy, a non-redundant factorial code [14] will be more compact than the raw data. Where there is predictability, compression can be achieved by assigning short codes to events that are predictable with high probability [3]. Generally speaking we may say that a major goal of traditional unsupervised learning is to improve the compression of the observed data, by discovering a program that computes and thus explains the history (and hopefully does so quickly) but is clearly shorter than the shortest previously known program of this kind.

According to our complexity-based theory of beauty [15, 17, 25], the agent's currently achieved compression performance corresponds to subjectively perceived beauty: among several sub-patterns classified as 'comparable' by a given observer, the subjectively most beautiful is the one with the simplest (shortest) description, given the observer's particular method for encoding and memorizing it. For example, mathematicians find beauty in a simple proof with a short description in the formal language they are using. Others like geometrically simple, aesthetically pleasing, low-complexity drawings of various objects [15, 17].

Traditional unsupervised learning is not enough though—it just analyzes and encodes the data but does not choose it. We have to extend it along the dimension of active action selection, since our unsupervised learner must also choose the actions that influence the observed data, just like a scientist chooses his experiments, a baby its toys, an artist his colors, a dancer his moves, or any attentive system its next sensory input.

Which data should the agent select by executing appropriate actions? Which are the *interesting* sensory inputs that deserve to be targets of its curiosity? I postulate [25] that in the absence of external rewards or punishment the answer is: Those that yield *progress* in data compression. What does this mean? New data observed by the learning agent may initially look rather random and incompressible and hard to explain. A good learner, however, will *improve* its compression algorithm over time, using some application-dependent learning algorithm, making parts of the data history subjectively more compressible, more explainable, more regular and more 'beautiful.' A beautiful thing is interesting only as long as it is new, that is, as long as the algorithmic regularity that makes it simple has not yet been fully assimilated by the adaptive observer who is still learning to compress the data better. So the agent's goal should be: create action sequences that extend the observation history and yield previously unknown / unpredictable but quickly learnable algorithmic regularity or compressibility. To rephrase this principle in an informal way: maximize the *first derivative* of subjective beauty.

An unusually large compression breakthrough deserves the name *discovery*. How can we motivate a reinforcement learning agent to make discoveries? Clearly, we cannot simply reward it for executing actions that just yield a compressible but boring history. For example, a vision-based agent that always stays in the dark will experience an extremely compressible and uninteresting history of unchanging sensory inputs. Neither can we reward it for executing actions that yield highly informative but uncompressible data. For example, our agent sitting in front of a screen full of white noise will experience highly unpredictable and fundamentally uncompressible and uninteresting data conveying a lot of information in the traditional sense of Boltzmann and Shannon [32]. Instead, the agent should receive reward for creating / observing data that allows for *improvements* of the data's subjective compressibility.

The appendix will describe formal details of how to implement this principle on computers. The next section will provide examples of subjective beauty tailored to human observers, and illustrate the learning process leading from less to more subjective beauty. Then I will argue that the *creativity* of artists, dancers, musicians, pure mathematicians as well as unsupervised *attention* in general is just a by-product of our principle, using qualitative examples to support this hypothesis.

2 Visual Examples of Subjective Beauty and its 'First Derivative' Interestingness

Figure 1 depicts the drawing of a female face considered *'beautiful'* by some human observers. It also shows that the essential features of this face follow a very simple geometrical pattern [17] to be specified by very few bits of information. That is, the data stream generated by observing the image (say, through a sequence of eye saccades) is more compressible than it would be in the absence of such regularities. Although few people are able to immediately see how the drawing was made without studying its grid-based explanation (right-hand side of Figure 1), most do notice that the facial features somehow fit together and exhibit some sort of regularity. According to our postulate, the observer's reward is generated by the conscious or subconscious discovery of this compressibility. The face remains interesting until its observation does not reveal any additional previously unknown regularities. Then it becomes boring even in the eyes of those who think it is beautiful—beauty and interestingness are two different things.

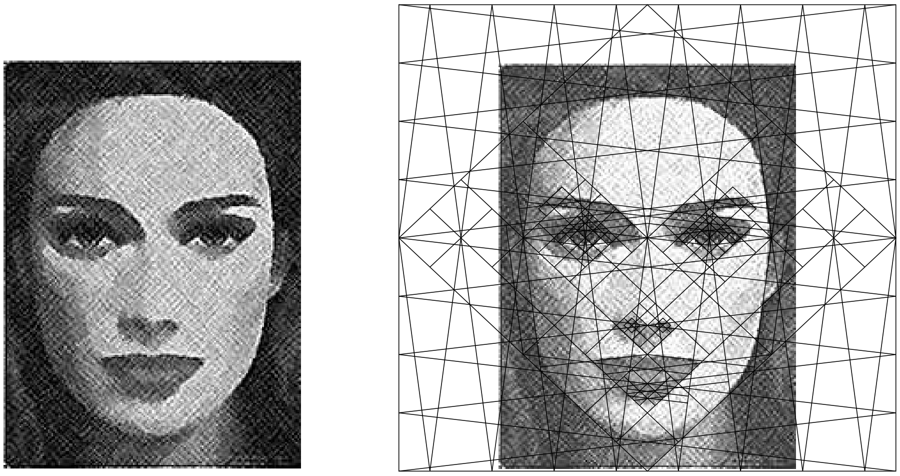

Figure 1: **Left:** *Drawing of a female face based on a previously published construction plan [17] (1998). Some human observers report they feel this face is 'beautiful.' Although the drawing has lots of noisy details (texture etc) without an obvious short description, positions and shapes of the basic facial features are compactly encodable through a very simple geometrical scheme. Hence the image contains a highly compressible algorithmic regularity or pattern describable by few bits of information. An observer can perceive it through a sequence of attentive eye movements or saccades, and consciously or subconsciously discover the compressibility of the incoming data stream.* **Right:** *Explanation of how the essential facial features were constructed [17]. First the sides of a square were partitioned into 2^4 equal intervals. Certain interval boundaries were connected to obtain three rotated, superimposed grids based on lines with slopes ± 1 or $\pm 1/2^3$ or $\pm 2^3/1$. Higher-resolution details of the grids were obtained by iteratively selecting two previously generated, neighbouring, parallel lines and inserting a new one equidistant to both. Finally the grids were vertically compressed by a factor of $1 - 2^{-4}$. The resulting lines and their intersections define essential boundaries and shapes of eyebrows, eyes, lid shades, mouth, nose, and facial frame in a simple way that is obvious from the construction plan. Although this plan is simple in hindsight, it was hard to find: hundreds of my previous attempts at discovering such precise matches between simple geometries and pretty faces failed.*

Figure 2 provides another example: a butterfly and a vase with a flower. The image to the left can be specified by very few bits of information; it can be constructed through a very simple procedure or algorithm based on fractal circle patterns [15]. People who understand this algorithm tend to appreciate the drawing more than those who do not. They realize how simple it is. This is not an immediate, all-or-nothing, binary process though. Since the typical human visual system has a lot of experience with circles, most people quickly notice that the curves somehow fit together in a regular way. But few are able to immediately state the precise geometric principles underlying the drawing. This pattern, however, is learnable from the right-hand side of Figure 2. The conscious or subconscious discovery process leading from a longer to a shorter description of the data, or from less to more compression, or from less to more subjectively perceived beauty, yields reward depending on the first derivative of subjective beauty.

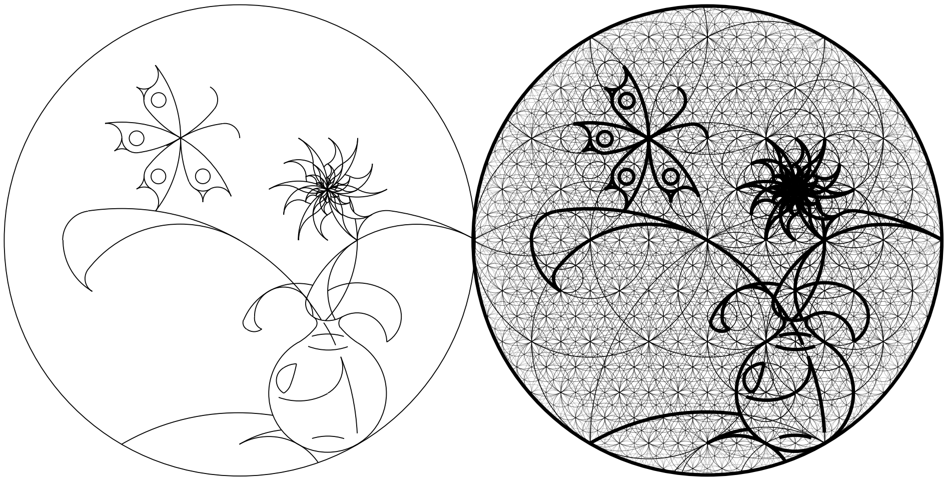

Figure 2: **Left:** *Image of a butterfly and a vase with a flower, reprinted from* Leonardo *[15, 25].* **Right:** *Explanation of how the image was constructed through a very simple algorithm exploiting fractal circles [15]. The frame is a circle; its leftmost point is the center of another circle of the same size. Wherever two circles of equal size touch or intersect are centers of two more circles with equal and half size, respectively. Each line of the drawing is a segment of some circle, its endpoints are where circles touch or intersect. There are few big circles and many small ones. In general, the smaller a circle, the more bits are needed to specify it. The drawing to the left is simple (compressible) as it is based on few, rather large circles. Many human observers report that they derive a certain amount of pleasure from discovering this simplicity. The observer's learning process causes a reduction of the subjective complexity of the data, yielding a temporarily high derivative of subjective beauty. (Again I needed a long time to discover a satisfactory way of using fractal circles to create a reasonable drawing.)*

3 Compressibility-Based Rewards of Art and Music

The examples above indicate that works of art and music may have important purposes beyond their social aspects [1] despite of those who classify art as superfluous [10]. Good observer-dependent art deepens the observer's insights about this world or possible worlds, unveiling previously unknown regularities in compressible data, connecting previously disconnected patterns in an initially surprising way that makes the combination of these patterns subjectively more compressible, and eventually becomes known and less interesting. I postulate that the active creation and attentive perception of all kinds of artwork are just by-products of my curiosity principle yielding reward for compressor improvements.

Let us elaborate on this idea in more detail, following the discussion in [25]. Artificial or human observers must perceive art sequentially, and typically also actively, e.g., through a sequence of attention-shifting eye saccades or camera movements scanning a sculpture, or internal shifts of attention that filter and emphasize sounds made by a pianist, while surpressing background noise. Undoubtedly many derive pleasure and rewards from perceiving works of art, such as certain paintings, or songs. But different subjective observers with different sensory apparati and compressor improvement algorithms will prefer different input sequences. Hence any objective theory of what is good art must take the subjective observer as a parameter, to answer questions such as: Which action sequences should he select to maximize his pleasure? According to our principle he should select one that maximizes the quickly learnable compressibility that is new, relative to his current knowledge and his (usually limited) way of incorporating or learning new data.

For example, which song should some human observer select next? Not the one he just heard ten times in a row. It became too predictable in the process. But also not the new weird one with the completely unfamiliar rhythm and tonality. It seems too irregular and contain too much arbitrariness and subjective noise. He should try a song that is unfamiliar enough to contain somewhat unexpected harmonies or melodies or beats etc., but familiar enough to allow for quickly recognizing the presence of a new learnable regularity or compressibility in the sound stream. Sure, this song will get boring over time, but not yet.

The observer dependence is illustrated by the fact that Schönberg's twelve tone music is less popular than certain pop music tunes, presumably because its algorithmic structure is less obvious to many human observers as it is based on more complicated harmonies. For example, frequency ratios of successive notes in twelve tone music often cannot be expressed as fractions of very small integers. Those with a prior education about the basic concepts and objectives and constraints of twelve tone music, however, tend to appreciate Schönberg more than those without such an education.

All of this perfectly fits our principle: The current compressor of a given subjective observer tries to compress his history of acoustic and other inputs where possible. The action selector tries to find history-influencing actions that improve the compressor's performance on the history so far. The interesting musical and other subsequences are those with previously unknown yet learnable types of regularities, because they lead to compressor improvements. The boring patterns are those that seem arbitrary or random, or whose structure seems too hard to understand.

Similar statements not only hold for other dynamic art including film and dance (taking into account the compressibility of controller actions), but also for painting and sculpture, which cause dynamic pattern sequences due to attention-shifting actions [31] of the observer.

Just as observers get intrinsic rewards from sequentially focusing attention on artwork that exhibits new, previously unknown regularities, the 'creative' artists get reward for making it. For example, I found it extremely rewarding to discover (after hundreds of frustrating failed attempts) the simple geometric regularities that permitted the construction of the drawings in Figures 1 and 2. The distinction between artists and observers is not clear though. Artists can be observers and vice versa. Both artists and observers execute action sequences. The intrinsic motivations of both are fully compatible with our simple principle. Some artists, however, crave *external* reward from other observers, in form of praise, money, or both, in addition to the *internal* reward that comes from creating a new work of art. Our principle, however, conceptually separates these two types of reward.

From our perspective, scientists are very much like artists. They actively select experiments in search for simple laws compressing the observation history. For example, different apples tend to fall off their trees in similar ways. The discovery of a law underlying the acceleration of all falling apples helps to greatly compress the recorded data.

The framework in the appendix is sufficiently formal to allow for implementation of our principle on computers. The resulting artificial observers will vary in terms of the computational power of their history compressors and learning algorithms. This will influence what is good art / science to them, and what they find interesting.

A Appendix

This appendix is a compactified, compressibility-oriented variant of parts of [25].

The world can be explained to a degree by compressing it. The compressed version of the data can be viewed as its explanation. Discoveries correspond to large data compression improvements (found by the given, application-dependent compressor improvement algorithm). How to build an adaptive agent that not only tries to achieve externally given rewards but also to discover, in an unsupervised and experiment-based fashion, explainable and compressible data? (The explanations gained through explorative behavior may eventually help to solve teacher-given tasks.)

Let us formally consider a learning agent whose single life consists of discrete cycles or time steps $t = 1, 2, \ldots, T$. Its complete lifetime T may or may not be known in advance. In what follows, the value of any time-varying variable Q at time t ($1 \leq t \leq T$) will be denoted by $Q(t)$, the ordered sequence of values $Q(1), \ldots, Q(t)$ by $Q(\leq t)$, and the (possibly empty) sequence $Q(1), \ldots, Q(t-1)$ by $Q(< t)$. At any given t the agent receives a real-valued input $x(t)$ from the environment and executes a real-valued action $y(t)$ which may affect future inputs. At times $t < T$ its goal is to maximize future success or *utility*

$$u(t) = E_\mu \left[\sum_{\tau=t+1}^{T} r(\tau) \;\middle|\; h(\leq t) \right], \tag{1}$$

where $r(t)$ is an additional real-valued reward input at time t, $h(t)$ the ordered triple $[x(t), y(t), r(t)]$ (hence $h(\leq t)$ is the known history up to t), and $E_\mu(\cdot \mid \cdot)$ denotes the conditional expectation operator with respect to some possibly unknown distribution μ from a set $\mathcal{M}$ of possible distributions. Here $\mathcal{M}$ reflects whatever is known about the possibly probabilistic reactions of the environment. For example, $\mathcal{M}$ may contain all computable distributions [33, 34, 9, 4]. There is just one life, no need for predefined repeatable trials, no restriction to Markovian interfaces between sensors and environment, and the utility function implicitly takes into account the expected remaining lifespan $E_\mu(T \mid h(\leq t))$ and thus the possibility to extend it through appropriate actions [23, 26, 24].

Recent work has led to the first learning machines that are universal and optimal in various very general senses [4, 23, 26, 27, 28, 29]. Such machines can in principle find out by themselves whether curiosity and world model construction are useful or useless in a given environment, and learn to behave accordingly. The present appendix, however, will assume *a priori* that compression / explanation of the history is good and should be done; here we shall not worry about the possibility that 'curiosity may kill the cat.' Towards this end, in the spirit of our previous work [12, 11, 35, 16, 18], we split the reward signal $r(t)$ into two scalar real-valued components: $r(t) = g(r_{ext}(t), r_{int}(t))$, where g maps pairs of real values to real values, e.g., $g(a, b) = a + b$. Here $r_{ext}(t)$ denotes traditional *external* reward provided by the environment, such as negative reward in response to bumping against a wall, or positive reward in response to reaching some teacher-given goal state. But I am especially interested in $r_{int}(t)$, the internal or intrinsic or *curiosity* reward, which is provided whenever the data compressor / internal world model of the agent improves in some sense. Our initial focus will be on the case $r_{ext}(t) = 0$ for all valid t. The basic principle is essentially the one we published before in various variants [11, 12, 35, 16, 18, 22, 25]:

Principle 1 *Generate curiosity reward for the controller in response to improvements of the history compressor.*

So we conceptually separate the goal (explaining / compressing the history) from the means of achieving the goal. Once the goal is formally specified in terms of an algorithm for computing curiosity rewards, let the controller's reinforcement learning (RL) mechanism figure out how to translate such rewards into action sequences that allow the given compressor improvement algorithm to find and exploit previously unknown types of compressibility.

A.1 Predictors vs Compressors

Most of our previous work on artificial curiosity was prediction-oriented, e. g., [11, 12, 35, 16, 18, 22, 25]. Prediction and compression are closely related though. A predictor that correctly predicts many $x(\tau)$, given history $h(< \tau)$, for $1 \leq \tau \leq t$, can be used to encode $h(\leq t)$ compactly: Given the predictor, only the wrongly predicted $x(\tau)$ plus information about the corresponding time steps τ are necessary to reconstruct history $h(\leq t)$, e.g., [13]. Similarly, a predictor that learns a probability distribution of the possible next events, given previous events, can be used to efficiently encode observations with high (respectively low) predicted probability by few (respectively many) bits [3, 30], thus achieving a compressed history representation. Generally speaking, we may view the predictor as the essential part of a program p that re-computes $h(\leq t)$. If this program is short in comparison to the rad data $h(\leq t)$, then $h(\leq t)$ is regular or non-random [33, 7, 9, 19], presumably reflecting essential environmental laws. Then p may also be highly useful for predicting future, yet unseen $x(\tau)$ for $\tau > t$.

A.2 Compressor Performance Measures

At any time t ($1 \leq t < T$), given some compressor program p able to compress history $h(\leq t)$, let $C(p, h(\leq t))$ denote p's compression performance on $h(\leq t)$. An appropriate performance measure would be

$$C_l(p, h(\leq t)) = l(p), \tag{2}$$

where $l(p)$ denotes the length of p, measured in number of bits: the shorter p, the more algorithimic regularity and compressibility and predictability and lawfulness in the observations so far. The ultimate limit for $C_l(p, h(\leq t))$ would be $K^*(h(\leq t))$, a variant of the Kolmogorov complexity of $h(\leq t)$, namely, the length of the shortest program (for the given hardware) that computes an output starting with $h(\leq t)$ [33, 7, 9, 19].

$C_l(p, h(\leq t))$ does not take into account the time $\tau(p, h(\leq t))$ spent by p on computing $h(\leq t)$. An alternative performance measure inspired by concepts of optimal universal search [8, 21] is

$$C_{l\tau}(p, h(\leq t)) = l(p) + \log\ \tau(p, h(\leq t)). \tag{3}$$

Here compression by one bit is worth as much as runtime reduction by a factor of $\frac{1}{2}$.

A.3 Compressor Improvement Measures

The previous Section A.2 only discussed measures of compressor performance, but not of performance *improvement*, which is the essential issue in our curiosity-oriented context. To repeat the point made above: *The important thing are the improvements of the compressor, not its compression performance per se.* Our curiosity reward in response to the compressor's progress (due to some application-dependent compressor improvement algorithm) between times t and $t+1$ should be

$$r_{int}(t+1) = f[C(p(t+1), h(\leq t+1)), C(p(t), h(\leq t+1))], \tag{4}$$

where f maps pairs of real values to real values. Various alternative progress measures are possible; most obvious is $f(a, b) = a - b$.

Note that both the old and the new compressor have to be tested on the same data, namely, the complete history so far.

A.4 Asynchronous Framework for Creating Curiosity Reward

Let $p(t)$ denote the agent's current compressor program at time t, $s(t)$ its current controller, and do:

Controller: At any time t ($1 \leq t < T$) do:

1. Let $s(t)$ use (parts of) history $h(\leq t)$ to select and execute $y(t+1)$.
2. Observe $x(t+1)$.
3. Check if there is non-zero curiosity reward $r_{int}(t+1)$ provided by the separate, asynchronously running compressor improvement algorithm (see below). If not, set $r_{int}(t+1) = 0$.
4. Let the controller's reinforcement learning (RL) algorithm use $h(\leq t+1)$ including $r_{int}(t+1)$ (and possibly also the latest available compressed version of the observed data—see below) to obtain a new controller $s(t+1)$, in line with objective (1).

Compressor: Set p_{new} equal to the initial data compressor. Starting at time 1, repeat forever until interrupted by death T:

1. Set $p_{old} = p_{new}$; get current time step t and set $h_{old} = h(\leq t)$.
2. Evaluate p_{old} on h_{old}, to obtain $C(p_{old}, h_{old})$ (Section A.2). This may take many time steps.
3. Let some (application-dependent) compressor improvement algorithm (such as a learning algorithm for an adaptive neural network predictor) use h_{old} to obtain a hopefully better compressor p_{new} (such as a neural net with the same size but improved prediction capability and therefore improved compression performance). Although this may take many time steps, p_{new} may not be optimal, due to limitations of the learning algorithm, e.g., local maxima.
4. Evaluate p_{new} on h_{old}, to obtain $C(p_{new}, h_{old})$. This may take many time steps.
5. Get current time step τ and generate curiosity reward

   $$r_{int}(\tau) = f[C(p_{old}, h_{old}), C(p_{new}, h_{old})], \tag{5}$$

   e.g., $f(a, b) = a - b$; see Section A.3.

Obviously this asynchronuous scheme may cause long temporal delays between controller actions and corresponding curiosity rewards. This may impose a heavy burden on the controller's RL algorithm whose task is to assign credit to past actions (to inform the controller about beginnings of compressor evaluation processes etc., we may augment its input by unique representations of such events). Nevertheless, there are RL algorithms for this purpose which are theoretically optimal in various senses, to be discussed next.

A.5 Optimal Curiosity & Creativity & Focus of Attention

Our chosen compressor class typically will have certain computational limitations. In the absence of any external rewards, we may define *optimal pure curiosity behavior* relative to these limitations: At time t this behavior would select the action that maximizes

$$u(t) = E_\mu \left[\sum_{\tau=t+1}^{T} r_{int}(\tau) \;\middle|\; h(\leq t) \right]. \tag{6}$$

Since the true, world-governing probability distribution μ is unknown, the resulting task of the controller's RL algorithm may be a formidable one. As the system is revisiting previously uncompressible parts of the environment, some of those will tend to become more compressible, that is, the corresponding curiosity rewards will decrease over time. A good RL algorithm must somehow detect and then *predict* this decrease, and act accordingly. Traditional RL algorithms [6], however, do not provide any theoretical guarantee of optimality for such situations. (This is not to say though that sub-optimal RL methods may not lead to success in certain applications; experimental studies might lead to interesting insights.)

Let us first make the natural assumption that the compressor is not super-complex such as Kolmogorov's, that is, its output and $r_{int}(t)$ are computable for all t. Is there a best possible RL algorithm that comes as close as any other to maximizing objective (6)? Indeed, there is. Its drawback, however, is that it is not computable in finite time. Nevertheless, it serves as a reference point for defining what is achievable at best.

A.6 Optimal But Incomputable Action Selector

There is an optimal way of selecting actions which makes use of Solomonoff's theoretically optimal universal predictors and their Bayesian learning algorithms [33, 34, 9, 4, 5]. The latter only assume that the reactions of the environment are sampled from an unknown probability distribution μ contained in a set $\mathcal{M}$ of all enumerable distributions—compare text after equation (1). More precisely, given an observation sequence $q(\leq t)$, we only assume there exists a computer program that can compute the probability of the next possible $q(t+1)$, given $q(\leq t)$. In general we do not know this program, hence we predict using a mixture distribution

$$\xi(q(t+1) \mid q(\leq t)) = \sum_i w_i \mu_i(q(t+1) \mid q(\leq t)), \tag{7}$$

a weighted sum of *all* distributions $\mu_i \in \mathcal{M}$, $i = 1, 2, \ldots$, where the sum of the constant weights satisfies $\sum_i w_i \leq 1$. This is indeed the best one can possibly do, in a very general sense [34, 4]. The drawback of the scheme is its incomputability, since $\mathcal{M}$ contains infinitely many distributions. We may increase the theoretical power of the scheme by augmenting $\mathcal{M}$ by certain non-enumerable but limit-computable distributions [19], or restrict it such that it becomes computable, e.g., by assuming the world is computed by some unknown but deterministic computer program sampled from the Speed Prior [20] which assigns low probability to environments that are hard to compute by any method.

Once we have such an optimal predictor, we can extend it by formally including the effects of executed actions to define an optimal action selector maximizing future expected reward. At any time t, Hutter's theoretically optimal (yet uncomputable) RL algorithm AIXI [4] uses an extended version of Solomonoff's prediction scheme to select those action sequences that promise maximal future reward up to some horizon T, given the current data $h(\leq t)$. That is, in cycle $t+1$, AIXI selects as its next action the first action of an action sequence maximizing ξ-predicted reward up to the given horizon, appropriately generalizing eq. (7). AIXI uses observations optimally [4]: the Bayes-optimal policy p^{ξ} based on the mixture ξ is self-optimizing in the sense that its average utility value converges asymptotically for all $\mu \in \mathcal{M}$ to the optimal value achieved by the Bayes-optimal policy p^{μ} which knows μ in advance. The necessary and sufficient condition is that $\mathcal{M}$ admits self-optimizing policies. The policy p^{ξ} is also Pareto-optimal in the sense that there is no other policy yielding higher or equal value in *all* environments $\nu \in \mathcal{M}$ and a strictly higher value in at least one [4].

A.7 Computable Selector of Provably Optimal Actions, Given Current System

AIXI above needs unlimited computation time. Its computable variant AIXI*(t,l)* [4] has asymptotically optimal runtime but may suffer from a huge constant slowdown. To take the consumed computation time into account in a general, optimal way, we may use the recent Gödel machines [23, 26, 24] instead. They represent the first class of mathematically rigorous, fully self-referential, self-improving, general, optimally efficient problem solvers. They are also applicable to the problem embodied by objective (6).

The initial software $\mathcal{S}$ of such a Gödel machine contains an initial problem solver, e.g., some typically sub-optimal method [6]. It also contains an asymptotically optimal initial proof searcher based on an online variant of Levin's *Universal Search* [8], which is used to run and test *proof techniques*. Proof techniques are programs written in a universal language implemented on the Gödel machine within $\mathcal{S}$. They are in principle able to compute proofs concerning the system's own future performance, based on an axiomatic system $\mathcal{A}$ encoded in $\mathcal{S}$. $\mathcal{A}$ describes the formal *utility* function, in our case eq. (6), the hardware properties, axioms of arithmetic and probability theory and data manipulation etc, and $\mathcal{S}$ itself, which is possible without introducing circularity [26].

Inspired by Kurt Gödel's celebrated self-referential formulas (1931), the Gödel machine rewrites any part of its own code (including the proof searcher) through a self-generated executable program as soon as its *Universal Search* variant has found a proof that the rewrite is *useful* according to objective (6). According to the Global Optimality Theorem [23, 26, 24], such a self-rewrite is globally optimal—no local maxima possible!—since the self-referential code first had to prove that it is not useful to continue the search for alternative self-rewrites.

If there is no provably useful optimal way of rewriting $\mathcal{S}$ at all, then humans will not find one either. But if there is one, then $\mathcal{S}$ itself can find and exploit it. Unlike the previous *non*-self-referential methods based on hardwired proof searchers [4], Gödel machines not only boast an optimal *order* of complexity but can optimally reduce (through self-changes) any slowdowns hidden by the $O()$-notation, provided the utility of such speed-ups is provable.

A.8 Consequences of Optimal Action Selecton

Now let us apply any optimal RL algorithm to curiosity rewards as defined above. The expected consequences are: at time t the controller will do the best to select an action $y(t)$ that starts an action sequence expected to create observations yielding maximal expected compression *progress* up to the expected death T, taking into accunt the limitations of both the compressor and the compressor improvement algorithm. In particular, ignoring issues of computation time, it will focus in the best possible way on things that are currently still uncompressible but will soon become compressible through additional learning. It will get bored by things that already are compressible. It will also get bored by things that are currently uncompressible but will apparently remain so, given the experience so far, or where the costs of making them compressible exceed those of making other things compressible, etc.

References

[1] M. Balter. Seeking the key to music. *Science*, 306:1120–1122, 2004.

[2] H. B. Barlow, T. P. Kaushal, and G. J. Mitchison. Finding minimum entropy codes. *Neural Computation*, 1(3):412–423, 1989.

[3] D. A. Huffman. A method for construction of minimum-redundancy codes. *Proceedings IRE*, 40:1098–1101, 1952.

[4] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2004. (On J. Schmidhuber's SNF grant 20-61847).

[5] M. Hutter. On universal prediction and Bayesian confirmation. *Theoretical Computer Science*, 2007.

[6] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: a survey. *Journal of AI research*, 4:237–285, 1996.

[7] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1:1–11, 1965.

[8] L. A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.

[9] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications (2nd edition)*. Springer, 1997.

[10] S. Pinker. *How the mind works*. 1997.

[11] J. Schmidhuber. Adaptive curiosity and adaptive confidence. Technical Report FKI-149-91, Institut für Informatik, Technische Universität München, April 1991. See also [12].

[12] J. Schmidhuber. Curious model-building control systems. In *Proceedings of the International Joint Conference on Neural Networks, Singapore*, volume 2, pages 1458–1463. IEEE press, 1991.

[13] J. Schmidhuber. Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242, 1992.

[14] J. Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.

[15] J. Schmidhuber. Low-complexity art. *Leonardo, Journal of the International Society for the Arts, Sciences, and Technology*, 30(2):97–103, 1997.

[16] J. Schmidhuber. What's interesting? Technical Report IDSIA-35-97, IDSIA, 1997. ftp://ftp.idsia.ch/pub/juergen/interest.ps.gz; extended abstract in Proc. Snowbird'98, Utah, 1998; see also [18].

[17] J. Schmidhuber. Facial beauty and fractal geometry. Technical Report TR IDSIA-28-98, IDSIA, 1998. Published in the Cogprint Archive: http://cogprints.soton.ac.uk.

[18] J. Schmidhuber. Exploring the predictable. In A. Ghosh and S. Tsuitsui, editors, *Advances in Evolutionary Computing*, pages 579–612. Springer, 2002.

[19] J. Schmidhuber. Hierarchies of generalized Kolmogorov complexities and nonenumerable universal measures computable in the limit. *International Journal of Foundations of Computer Science*, 13(4):587–612, 2002.

[20] J. Schmidhuber. The Speed Prior: a new simplicity measure yielding near-optimal computable predictions. In J. Kivinen and R. H. Sloan, editors, *Proceedings of the 15th Annual Conference on Computational Learning Theory (COLT 2002)*, Lecture Notes in Artificial Intelligence, pages 216–228. Springer, Sydney, Australia, 2002.

[21] J. Schmidhuber. Optimal ordered problem solver. *Machine Learning*, 54:211–254, 2004.

[22] J. Schmidhuber. Overview of artificial curiosity and active exploration, with links to publications since 1990, 2004. http://www.idsia.ch/˜juergen/interest.html.

[23] J. Schmidhuber. Completely self-referential optimal reinforcement learners. In W. Duch, J. Kacprzyk, E. Oja, and S. Zadrozny, editors, *Artificial Neural Networks: Biological Inspirations - ICANN 2005, LNCS 3697*, pages 223–233. Springer-Verlag Berlin Heidelberg, 2005. Plenary talk.

[24] J. Schmidhuber. Gödel machines: Towards a technical justification of consciousness. In D. Kudenko, D. Kazakov, and E. Alonso, editors, *Adaptive Agents and Multi-Agent Systems III (LNCS 3394)*, pages 1–23. Springer Verlag, 2005.

[25] J. Schmidhuber. Developmental robotics, optimal artificial curiosity, creativity, music, and the fine arts. *Connection Science*, 18(2):173–187, 2006.

[26] J. Schmidhuber. Gödel machines: fully self-referential optimal universal problem solvers. In B. Goertzel and C. Pennachin, editors, *Artificial General Intelligence*, pages 199–226. Springer Verlag, 2006. Variant available as arXiv:cs.LO/0309048.

[27] J. Schmidhuber. The new AI: General & sound & relevant for physics. In B. Goertzel and C. Pennachin, editors, *Artificial General Intelligence*, pages 175–198. Springer, 2006. Also available as TR IDSIA-04-03, arXiv:cs.AI/0302012.

[28] J. Schmidhuber. New millennium AI and the convergence of history. In W. Duch and J. Mandziuk, editors, *Challenges to Computational Intelligence*. Springer, in press, 2006. Also available as arXiv:cs.AI/0606081.

[29] J. Schmidhuber. 2006: Celebrating 75 years of AI - history and outlook: the next 25 years. In *Proceedings of the "50th Anniversary Summit of Artificial Intelligence" at Monte Verita, Ascona, Switzerland*. Springer Verlag, 2007. Variant available as arXiv:0708.4311.

[30] J. Schmidhuber and S. Heil. Sequential neural text compression. *IEEE Transactions on Neural Networks*, 7(1):142–146, 1996.

[31] J. Schmidhuber and R. Huber. Learning to generate artificial fovea trajectories for target detection. *International Journal of Neural Systems*, 2(1 & 2):135–141, 1991.

[32] C. E. Shannon. A mathematical theory of communication (parts I and II). *Bell System Technical Journal*, XXVII:379–423, 1948.

[33] R. J. Solomonoff. A formal theory of inductive inference. Part I. *Information and Control*, 7:1–22, 1964.

[34] R. J. Solomonoff. Complexity-based induction systems. *IEEE Transactions on Information Theory*, IT-24(5):422–432, 1978.

[35] J. Storck, S. Hochreiter, and J. Schmidhuber. Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the International Conference on Artificial Neural Networks, Paris*, volume 2, pages 159–164. EC2 & Cie, 1995.