Jim Portegies
Department of Mathematics and Computer Science
Eindhoven University of Technology
j.w.portegies@tue.nl

Ergo learning

How to build machines that learn and think like humans or animals? In this article Jim Portegies discusses the approach of Misha Gromov, the so-called 'ergo project'. Then he will highlight some closely related research in developmental robotics and artificial curiosity. Finally, Portegies will discuss what he thinks are next steps in the search for universal learning programs, and will make a case for the study of learning by imitating, or analysis by synthesis.

A few years ago, a friend told me about the online chat program CleverBot. The next few days I (embarrassingly) spent hours chatting with the bot, trying to prove in various ways that it wasn't actually a human I was chatting with, playing an interrogator at a Turing test. The conversations were absurd. CleverBot practically only reacted to the last sentence I said. I asked for its name several times and it gave me different answers. But this didn't rule out the possibility of chatting with a human (who was joking and couldn't hold the thread of a conversation). I was trying to make CleverBot say something that a human would never say. But I was doomed to fail, since CleverBot works with a system that reuses previous conversations with actual humans and so everything CleverBot said was basically said by a human at some point.

Surely, a Turing test would be set up differently: The human or machine behind the screen, or at the other end of the chat interface, would both try to convince the interrogator of their humanity. CleverBot is one of the best human-imitating chat programs around. But it doesn't pass such a Turing test by a long shot.

It means that Turing was too optimistic when he believed in 1950, that "...in about 50 years' time, it will be possible to programme computers, with a storage capacity of about 10^9, to make them play the imitation game so well that an average interrogator will not have more than 70% chance of making the right identification after five minutes of questioning" [24].

It intrigues me that, despite the many recent successes in machine learning and artificial intelligence, humans and animals still outperform machines on a large number of tasks. Humans are much better at generalizing from a small number of examples and carrying skills over from one task to another, and they are much more efficient doing so in terms of usage of energy, computational power and memory [13].

How to build machines that learn and think like humans or animals? How to design machines that pass the Turing Test? How to solve the grand problem of artificial intelligence, of designing artificial agents that interact optimally with real-world environments machines, given *limited resources* [19]? How to interpret machine-learning models [4, 11]?

It was another friend who told me about the approach of Misha Gromov to these questions. Until then, I had known Misha Gromov as a mathematician, for his work in geometry, and I did not know about his interest in artificial intelligence. I looked up the article on Gromov's website, and it somehow resonated with me. Over the last years, Gromov has posted and updated his articles several times [7, 8]. Recently, he bundled his thoughts in a book *Great Circle of Mysteries: Mathematics, the World, the Mind*, and at the time of writing I am waiting for the translation in English, to read Gromov's new account on what he calls the 'ergo project'.

The ergo project revolves around the conjecture that inside the human brain runs a simple, efficient, *universal learning algorithm* that applies indiscriminately to *any* incoming signal. That is, whether the incoming signal originates from the eyes,

the ears, or the sensorimotor system, it is processed in the same way. The algorithm finds *structure* in the incoming signals, finds *meaning* and *interpretation* and eventually leads to *understanding*.

A crucial part of the ergo project is to give *mathematical incarnations* to these last concepts. And whereas the idea is not new that inside the human brain runs a simple, efficient, universal learning algorithm, the vision of capturing a concept such as *meaning* in purely mathematical terms is rather original, and I think it is one of the aspects that sets the ergo project apart.

Another distinguishing factor of the ergo project is the big role of mathematics, and moreover, for mathematics yet to be developed. As such, it holds the promise to stimulate the development of a new mathematical field.

In this article I will discuss Gromov's ergo project in more detail. I will then highlight some closely related research in developmental robotics and artificial curiosity. Finally, I will discuss what I think are next steps in the search for universal learning programs, and in particular I will make a case for the study of *learning by imitating*, or *analysis by synthesis*.

Non-universal versus universal learning

Before I try to explain what universal learning is, let me first illustrate *non-universal* learning. Suppose we want to program a robot with a camera to navigate through a room. We could read the signals recorded by the robot's camera and immediately interpret them as light intensities on a grid. When we combine it with all our understanding of elementary optics and geometry of three-dimensional and projective space, it will be possible to hardcode a fairly successful navigation procedure. And this is currently done, of course, in many practical and toy robots.

In a theory on human learning called nativism, just like we know how to deal with the signals coming from the camera, the brain of an infant is assumed to be already preprogrammed to deal with the incoming signals.

The ergo philosophy is different. It is very similar to how Turing envisioned that a child's brain is a rather little mechanism, with lots of blank sheets: the ergo project is also surrounded by the expectation that in fact, very little preprogramming is present in the infant's brain. Without such preprogramming, the flow of signals entering a child's brain is much like a chaos of electrochemical sparks. Yet then, gradually and miraculously, an immensely powerful universal learning algorithm brain finds redundancies, patterns, structure in the chaos, starts to autonomously build interpretation, and finally starts understanding the incoming flow of signals.

Ego versus Ergo

Since this process of building understanding seems much stronger in early years of life, the expectation is that at least for adults, the ergo brain, that powerful universal learning algorithm, is dominated by other processes in the mind. According to Gromov, the mind roughly decomposes into two competing parts, the ego mind and the ergo brain, and this is certainly a helpful, and at times inspiring, metaphor.

The ego-mind is responsible for a person's primary needs, such as the needs for survival and reproduction. The ego-mind is at the surface of the brain, in our consciousness, and the part that we are most used to.

In the depths of our minds, as if hidden behind a wall, runs the ergo brain. Within this metaphor, we can explain the extraordinary capabilities of savants and of highly gifted mathematicians such as Ramanujan by cracks in the wall, through which the ergo brain shines through. And finally, children are all little Ramanujans, their task of finding structure in the chaos of electrochemical sparks harder and more abstract than the hardest mathematics.

The expectation is therefore that if we want to build universal learning algorithms, we should mimic how children learn. Children explore the world in an active, playful and curious manner. Their learning seems to be goal-free and independent of external rewards. They get bored of situations that they understand well, and stay away from situations which are too unpredictable.

To get an idea of the competition between ego and ergo, it is illustrative to look at the list of words Gromov associates with the ego mind,

intuitive, intelligent, rational, serious, objective, important, productive, efficient, successful, useful,

and those that belong to the ergo vocabulary,

interesting, meaningful, informative, funny, beautiful, curious, amusing, amazing, surprising, confusing, perplexing, predictable, nonsensical, boring.

With this division in ego and ergo comes of course a belief that in order to find universal learning algorithms one needs to follow an approach fitting the latter list of adjectives.

Let me now describe how we can go from metaphors closer to mathematics.

The ergo-learning conjecture

Gromov's ergo-learning conjecture states that (rephrased):

There exists a simple, *efficient*, universal *learning algorithm that finds structure*, *meaning and interpretation in any incoming stream of signals and finally converges to understanding*.

Universal means that no matter whether the signals come from visual, auditory or sensory input, or the signals are even generated internally, they are all processed using the same algorithm. Simple means logically simple, of low complexity. Efficient means that it can be realized within the human limits on computational, memory and energy resources.

What the words structure, meaning, interpretation and understanding mean is at this point unclear. So while working towards proving the conjecture, one needs to develop a mathematical theory of structure in signals. Similarly, one would need to develop mathematical concepts capturing the terms *understanding*, *interpretation*, *meaning* and *learning*.

The ultimate aim of the ergo project is to prove the conjecture by actually designing and implementing a universal learning algorithm. Such an ergo system would, on encountering *any* incoming flow of signals, start to interact with the flow and find meaning and understanding inside.

How plausible is ergo-learning conjecture?

This question can spawn a discussion that fits right into the nature-nurture debate. It is the discussion about how much preprogramming there is in an infant's brain, and how much understanding of the world is acquired through experience.

The main argument in favor of ergo learning is that evolution has not had

enough time to construct targeted learning algorithms for every possible signal and every possible task. Instead, an infant's brain is endowed with a simple, universal program that works for several signals. This is very similar to Turing's vision: he wrote that "Our hope is that there is so little in the child brain that something like it can be easily programmed."

As a reaction to psychological theories that were hard to test experimentally, scientists in the beginning of the twentieth century developed a new approach to psychology called behaviorism. It is far on the nurture end of the nature-nurture spectrum, and environmental factors, and in particular reinforcement by rewards and punishment, play a huge role [22].

Turing speculated that a learning machine could also be taught by using rewards and punishments and as such reflected the behaviorist point of view. Nonetheless, he himself indicated that other learning mechanisms would be necessary as well.

In 1957, Skinner published his behaviorist account on language processing [23] which prompted a famous critique by Chomsky [2]. Whereas in the behaviorist point of view, language is thought to be required solely through experience, the criticism by Chomsky was that children are not exposed to enough linguistic data to learn the features of the language, an observation he later called *Poverty of the stimulus*. Rather, key linguistic features should be innate, should be preprogrammed in the genetic structure. In particular, infants should have internalized a certain universal generative grammar.

There is currently no mathematical theory to support either side of the debate, but in this context I always think that it is astounding that the (naïve?) information content of a person's DNA is about 750 MB: it fits on a DVD. (This estimate is based on a total length of the human genome of approximately 3 billion base pairs, with each basepair accounting for two bits.) Many contemporary software packages will take significantly more space on the hard disk on your computer than that. It gives some indication of the limits on the complexity of the 'start-up' software of children.

Another strong hint at universality is that humans can learn language even when they are deaf or deaf-blind. This means that language can be learned independently of the signal carrying the linguistic information. Moreover, some humans have an ability to learn complicated structures, such as how to play chess, by pure observation. Many humans have an ability to learn mathematics. Since these are rather new, it is unimaginable that they are somehow encoded in the DNA.

There are also some claims and indicators of non-universality in human signal processing. The initial layers responsible for processing visual and auditory signals seem adapted to the type of signals they are processing. For instance, scientists such as Petitot claim that the connectivity and the geometrical arrangements in the first layers of the human visual system are crucial in processing the visual signals [17]. Moreover, the cochlea in the inner ear performs a type of Fourier transform of the incoming sound. Here, preprocessing of the signal happens even before the signal enters the neuronal system.

In contrast to this observation of non-universality, there is the observation that the brain shows a remarkable ability to adapt itself to new signals, an ability often referred to as 'brain plasticity'. For instance, there have been experiments with ferrets where the scientists connected the visual input to the auditory cortex. These ferrets could still 'see'. Moreover, the connectivities in the auditory cortex changed and formed a structure characteristic for the first layer(s) of the visual cortex [15]. This could be a reflection of ergo learning.

Ultimate aim

To me, the ultimate outcome of the ergo project are good mathematical definitions of *structure*, *meaning*, *interpretation* and *understanding*, and a 1 GB USB stick with an actual universal learning algorithm `Prog`. We should be able to install the universal learning algorithm in the 'brain' of an arbitrary robot, where the robot-brain is just a general-purpose computer with human-scale computation and memory capacity.

One robot may have a microphone, another may have a camera, a third may just have some pressure sensors. Some may produce sound, others images, some can move around and maybe others are completely stationary. We then let the robots interact with the environment, in particular, we expose them to sound recordings of text, let them flip through books. By running the algorithm `Prog`, the robots will start to find structure in the incoming signals, start to interact with it through its outgoing signals, and, within a few years, will converge to a form of understanding of language, for instance.

The design of a universal learning program is highly challenging, and some will believe that it is impossible. It is, in fact, even hard to envision at this stage what such a program may look like.

A robot learning to read

In Gromov's setup, 'understanding' is a combinatorial structure in itself, by which

Figure 1

we mean something like (but different from) a graph, a simplicial complex or an n-category. When the universal learning program is listening to the incoming flow of signals, the ergo-learning program will try to incorporate the structure that is diluted in the flow, in the combinatorial structure of the 'understanding'.

Suppose that the ergo-learning program `Prog` is connected to a large body of text, such as Wikipedia. The program starts by identifying textual units, such as words or word combinations that are persistent in the text. Importantly, we don't tell the program what a word is, or even a 'space'. It should perform the division into units purely based on statistical properties of the text.

Next, it would attach names or tags to some of these words and fragments. One can think of such tags being written on a line above the original string. One can iterate the annotation process and write $(l+1)$st order tags on top of the line with lth order tags. The number of tags should decrease with l.

The next step is the compression of the library, that is, the applications of several structure-preserving reductions, which results in a more elaborate combinatorial structure. This combinatorial structure reflects the grammar. It should be significantly smaller than the original body of text.

The resulting combinatorial structure will be a dynamically changing entity, which we will denote by $\mathcal{U}_t$. The program is iteratively applied to $\mathcal{U}_t$ to obtain the combinatorial structure at the next time point. Eventually, we expect this process to approximately converge to a fixed point. Then, we would say that the ergo brain 'understands' the language.

Generalizing from the above, the whole process of understanding would consist of three components:

- A combinatorial structure $\mathcal{U}$ in the mind, brain or program that 'understands';
- an ergo-system that implements $\mathcal{U}$, i.e. takes $\mathcal{U}$ as an input and uses it to process and interpret incoming (and internally generated) signals;
- the result of applying the implementation of $\mathcal{U}$ to incoming flows of signals.

Developing reasonable understanding of a language takes a long time, at least several years for humans. A similar timeframe holds for the understanding of a mathematical theory. That is why Gromov expects that the time-complexity of a learning process is log-linear in the size of the language or theory. On the other hand, the applications of the understanding to a flow of signals is extremely fast, and therefore expected to be logarithmic in the size of the signal.

Even though some will be skeptical about the feasibility of the ergo project, closely related research has been going on for decades, albeit under different names such as *artificial curiosity* [21], *developmental robotics* [3,12] and the *Bayesian brain hypothesis* [5,6]. (In fact, Jürgen Schmidhuber's survey article on artificial curiosity has the surprising title 'Formal theory of creativity, fun, and intrinsic motivation (1990–2010)'.)

Going back in time even further, Turing's original proposal of teaching a machine by rewards and punishments was made operational in a technique called reinforcement learning. Reinforcement learning in itself is not quite universal, but it can play an important role in more universal learning algorithms, so we review it first.

Reinforcement learning

Suppose we want to let the robot perform a certain task, such as play a game of soccer or chess. We could try to hardcode rules of play, program the robot to perform optimally, but this is a very non-universal approach. Instead, we let the robot figure out by itself how to play, except we, as external experts, do 'grade' the performance of the robot, and make our grading accessible to the robot, through a part of its input signal.

We can program the robot in such away that this part of the input signal gets the interpretation of a reward, which the robot tries to maximize (often on average). This approach goes by the name of *reinforcement learning*, and is a reflection of the behaviorist principle of learning through rewards and punishments.

Reinforcement learning has grown into an extremely powerful method to teach tasks and games to machines and robots [18].

Reinforcement learning has both universal and non-universal aspects. It is universal in that the same algorithm can be used on a large variety of tasks. However, reinforcement learning is predominantly used with outside-programmed reward signals that indicate performance on a certain task. In that sense it is very (external) goal-oriented, task-specific and therefore not really universal.

Artificial curiosity

Reinforcement learning is analogous to teaching an animal a trick by giving it food in return. In such a case, the reward signal is coupled to a primary (or secondary...) drive of the animal.

Around the fifties, psychologists started to realize that it is hard to explain the behavior of children, humans and animals by merely goal-oriented behavior. Experiments showed that animals had a craving for novel experiences and were behaving in exploratory ways even when their primary needs seemed to be satisfied, when exploratory behavior actually led to punishments, or when there really seemed to be no reward for an activity at all [1,25].

If children and animals are not solely motivated by external rewards, then how do they learn? And how could we mimic such behavior in robots? In particular, how can we instill exploratory behavior in robots, a craving for novel, surprising, funny experiences, and a certain focus on situations that are interesting because they are not too predictable nor too unpredictable? How can we artificially generate curiosity?

Interestingly, very similar suggested answers to these questions come from many different directions. The core of these suggestions is as follows: Endow the robot with a predictor of its own sensory input and choose its actions based on the accuracy of the predictions. The robot can measure this accuracy by itself, without the need for an external grader.

The above principle comes in many different flavors, as it depends on a particular system of prediction used and the particular quantities that are optimized and how the optimization procedure takes place.

A very interesting application was constructed by Oudeyer, Kaplan and Hafner [16]. Their robot seemed to go through various stages of development. If it would start to 'understand' a certain toy or situation, it would move on to the next, where there still was something new to learn.

Prediction and generation

The predictor is a key component of the above setup. It *generates* a signal that

Figure 2

can be compared to the true input. I first want to describe the simplest such predictor, which already contains the key idea of much more sophisticated versions.

We think of showing the robot a sequence of uniform grayscale images, according to the following procedure. To begin with, we choose a random 'original' picture according to a probability distribution that we call the *prior probability distribution*. There are just two choices for the original picture: it is either completely white or completely black. We call this picture Or and we keep it hidden from the robot. It stays fixed throughout the rest of the procedure.

We then show the robot a sequence of 'noisy' images Im_1, Im_2, …, in which the image is colored in a random gray scale in $\{0,1,...,10\}$, where 0 corresponds to black and 10 to white. This means for instance (just to give a concrete example) that if the original image is black, the probability that the grayscale of image j equals m is proportional to

$$\mathbb{P}[\mathsf{Im}_j = m \mid \mathsf{Or} = B] \sim \left(\frac{9}{10}\right)^m$$

and if the original image is white, the probability that the grayscale of image j equals m is proportional to

$$\mathbb{P}[\mathsf{Im}_j = m \mid \mathsf{Or} = W] \sim \left(\frac{9}{10}\right)^{10-m}.$$

The noise for image i is independent of the noise for image j if $i \neq j$.

The first ten images may look like shown in Figure 2. In inference, it is the task of the robot to infer what was the original picture, based on such a sequence of pictures. In prediction, it is the task of the robot to predict how the sequence continues.

The random picture Or is often called a latent variable. It 'explains' the sequence of images produced.

We first assume that the robot has full knowledge about the above procedure, including the various probabilities, so it can use this knowledge to do inference and prediction. Let us first see how inference works.

After the robot sees one image with gray scale m, it can calculate the probability that the original image was the black image B using Bayes' formula

$$\mathbb{P}[\mathsf{Or} = B \mid \mathsf{Im}_1 = m] = \frac{\mathbb{P}[\mathsf{Im}_1 = m \mid \mathsf{Or} = B]\,\mathbb{P}[\mathsf{Or} = B]}{\mathbb{P}[\mathsf{Im}_1 = m]}.$$

This conditional probability distribution on the original picture is called the *posterior probability distribution*.

After viewing more images, the robot can in the same way calculate the posterior probability that the original image was black, given observations of the images $\mathsf{Im}_1, \ldots, \mathsf{Im}_n$. The estimate the robot makes this way becomes more and more accurate (in fact exponentially fast). This is the essential idea of Bayesian inference.

Let us now look at Bayesian prediction. In this simple case, where the robot has full knowledge about the model generating the images, the robot uses that same generative model for its predictor: The robot's predictor generates a sequence of images exactly following the same procedure as above. After observing images, however, it will update its predictor: it uses the exact same generating process except for replacing the *prior* probability distribution for picking the original image by the *posterior* probability distribution. This way, after a few observations, the predicted signal of the robot will be much closer to the true signal.

This was a very simple example, but it contains the basis of Bayesian inference and prediction. Let us now make it a bit more complicated.

We are going to generate a sequence of images of arrangements of overlapping boxes. We first sample a random number N, which will be the total number of boxes in the picture, and for each of the boxes, we choose a random size. We keep both the number and the sizes fixed (and will sometimes refer to them as the fixed latent variables). To generate images, we vary the

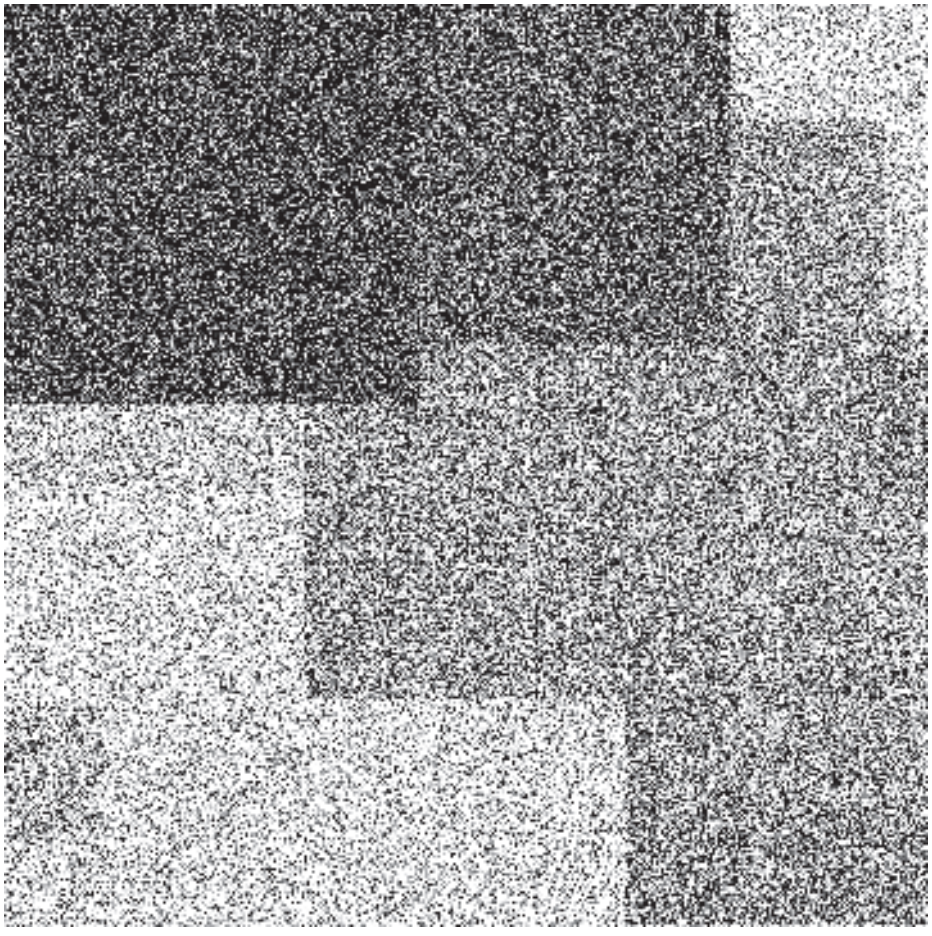

Figure 3

gray-scale, the xy-location, and an order parameter h of the boxes at random (and will refer to them as changing latent variables). A box is covering another box if the order parameter of the first box is larger than that of the second. Finally, we add Gaussian noise to every picture. A resulting picture may look like Figure 3, can you infer the number of boxes?

The various continuous variables present in the problem make it very hard to do Bayesian inference directly. For general distributions, it is impossible for the robot to find a good representation of the posterior distribution. There are several ways to deal with this problem, one of which is by sampling, and the other is by a technique called *variational inference*, where one approximates the true posterior distribution by an element of a simpler, parametrized family of distributions.

If the robot has full knowledge of the model, it can by observing the sequence of images obtain ever-more accurate estimates of the fixed latent variables: the number of boxes in the images and their sizes. For every single image, the robot can estimate the changing latent variables, namely the position of the boxes and their grayscale. In general, this procedure works quite well.

Analysis by synthesis

The situation changes drastically when the generating model is not known to the robot. Then we do not even know which variables to infer. Is there a way in which the robot may still find out the generative process?

My expectation is that this is possible if the robot is able to accurately approximate the signal (or rather its statistical properties), with a generative model that is as simple as possible. In that case, in the above example, the robot will in fact have encoded a certain number of boxes, a concept of size of the boxes, a concept of location and a concept of order. This way, the robot would have learned by constructing the signal itself, by applying analysis by synthesis.

The heuristic applied here is also known as Occam's razor, to select the simplest model explaining the observations. In the building where I work, there is restricted elevator access in the weekend. As my office is on the seventh floor, I soon discovered that I cannot go up to my office using the elevator without swiping my access card. Leaving the office, however, was never a problem. I stored this rule (and this was a golden truth to me) as "in the weekend you can go down, but you can't go up". Until, of course, a visitor who took the stairs got stuck on the third floor, and it turned out I cannot go down from my office to get him. I came up with a new rule (you can always go to floor 0 and 1, but not to any other floor) and am again utterly convinced of its truth, even though I never tested whether you can go from floor 4 to floor 6.

There are several layers to this story, but the main message is that humans have an uncanny ability to generalize from a very limited number of examples, favoring simpler explanations. The models that we build for ourselves are often wrong, but it doesn't matter much until we encounter new evidence, and we easily build a new simple model. It is a real challenge to get robots to do something similar.

One way is by letting the robot perform *inductive inference*. In a way, inductive inference is Bayesian inference taken to an extreme: because the robot does not know what is the true generative model, it adds the model itself as a (fixed) latent variable! In this case, the robot's generator first selects at random *any* sequence-producing model, and then samples from this model as before. The robot implements Occam's razor by taking the complexity of the models into account in the prior probability distribution: 'simpler' models are selected with higher probability.

There are, of course, a few problems with this approach. The space of all sequence-producing models is too large to handle effectively. In addition, one needs to have a workable concept of complexity of models, and although in this generality there are beautiful mathematical definitions such as Kolmogorov complexity, the fact that the Kolmogorov complexity cannot be computed is problematic.

Nonetheless, Hutter introduced a rational reinforcement learning agent, called AIXI [9], completely based on inductive inference. The AIXI algorithm is more like an abstract, mathematical object, since the model is uncomputable. Hutter also introduced a computable procedure, called AIXI(t,l), but it suffers from very large computation times. Schmidhuber developed the Gödel machine partly to deal with this issue of computation time [20]. The Gödel machine can rewrite its own software, after it has proven that such a rewrite is useful for rewards and speedup.

Although these learning algorithms are universal, even for the Gödel machine the computation times seem so large that its role in an ergo-learning system is questionable.

Variational autoencoders

I still believe that analysis by synthesis is possible by following the heuristic of Occam's razor: if the robot does not know the generative model producing the data, it tries to search over a whole class of generative models that best approximates the observed signal, but keeping in account the complexity.

However, instead of considering about every possible generative model as in inductive inference, I think it will be necessary to restrict to a class of generative models for which it is easier to define and compute measures of complexity, which are easier to train and easier to analyze mathematically, but still are capable of encoding efficiently and accurately a wide range of signals.

One such restricted class of generative models are generative models implemented by deep neural networks, such as the Variational Autoencoders introduced by Kingma and Welling [10].

In a Variational Autoencoder, one completely artificially adds a (changing) latent variable Z to the system, often with values in $\mathbb{R}^n$. One then optimizes over a whole class of generative models which generate signals by sending the latent variable, together with a noise variable, through a deep neural network; different choices of parameters of the neural network give rise to different generative models. One adapts the parameters of the neural network so that the true signal is approximated well, but it is not the only quantity that is optimized. At the same time, one uses a second deep neural network for variational inference: that is, one approximates the true posterior distribution by the family of distributions that one gets by varying the parameters of the second network. (This most basic setup of a Variational Autoencoder takes into account the complexity of Z — through the principle of minimum description length —, but not of the generative model itself. One could,

instead, incorporate the complexity of the generative model by also adding the parameters of the neural network as fixed latent variables in a Bayesian setup. This would be closer to inductive inference. One can even iterate the procedure, and get models similar to the hierarchical models Friston proposes in his variational-inference approach to the Bayesian brain hypothesis [5].)

Together with Vlado Menkovski and Luis Pérez, we are currently investigating the use of Variational Autoencoders for analysis by synthesis. For instance, if in the example with the pictures with boxes, we fix the number of boxes at 1, we expect that from the latent variable Z we can read off the x-coordinate, the y-coordinate and the grayscale of the box.

The x-coordinate will almost certainly not correspond to the first component of Z, but (in a slightly idealized case) we can find a (linear?) change of coordinates Φ from $\mathbb{R}^n$ to $\mathbb{R}^n$ so that the first, second and third component of $\Phi(Z)$ correspond to the x-coordinate, y-coordinate and the gray scale of the box, respectively.

This way, we are naturally led to questions about when two generative models are equivalent, about when there exists a translation from one generative model to the other and back. This opens the door to more mathematical definitions of interpretations of models and structures diluted in signals.

Indeed, together with Rostislav Matveev we looked at a definition of structure in signals for a certain type of translation between signals relevant for information processing [14]. If a signal consists of multiple components, the concept of structure we define naturally includes the Shannon mutual information between the components. However, it also covers finer dependency structure relevant for information processing.

Conclusion

It is clear that the ergo project is in its infancy, but there seem to be a few directions in which we can try to make small steps forward, such as in the search for relevant definitions of structure in signals, and in the development of theory and understanding of the process of analysis by synthesis, for instance in the context of deep generative models.

My personal belief is that in some decades from now, the ergo project and related activities will have lead to a new branch of mathematics, in which words such as *structure*, *meaning*, *interpretation* and *understanding* are mathematical concepts, and the only reason that we won't be able to construct a 1 GB USB stick with a universal learning program `Prog` that can be used to let a robot find structure and eventually understand a wide range of flows, will be that 1 GB USB sticks are no longer to be found.

References

1 Daniel E. Berlyne, Novelty and curiosity as determinants of exploratory behaviour, *British Journal of Psychology* 41(1-2) (1950), 68–80.

2 Noam Chomsky, A review of B. F. Skinner's Verbal Behavior. *Language* 35(1) (1959), 26–58.

3 Angelo Cangelosi, Matthew Schlesinger, and Linda B. Smith, *Developmental Robotics: From Babies to Robots*, MIT Press, 2015.

4 Finale Doshi-Velez and Been Kim, A roadmap for a rigorous science of interpretability, arXiv:1702.08608, 2017.

5 Karl Friston, The free-energy principle: a unified brain theory?, *Nature Reviews Neuroscience* 11(2) (2010), 127–138.

6 Karl Friston, The history of the future of the bayesian brain, *NeuroImage* 62(2) (2012), 1230–1233.

7 Misha Gromov, *Structures, Learning and Ergosystems*, Chapters 1–4, 6, www.ihes.fr/~gromov/PDF/ergobrain.pdf, 2011.

8 Misha Gromov, *Memorandum Ergo*, www.ihes.fr/~gromov/PDF/ergo-cut-copyOct29.pdf, 2015.

9 Marcus Hutter, *Universal artificial intelligence: Sequential decisions based on algorithmic probability*, Springer, 2004.

10 Diederik P. Kingma and Max Welling, Auto-encoding Variational Bayes, arXiv:1312.6114, 2013.

11 Zachary C. Lipton, The mythos of model interpretability, arXiv:1606.03490, 2016.

12 Max Lungarella, Giorgio Metta, Rolf Pfeifer and Giulio Sandini, Developmental robotics: a survey, *Connection science* 15(4) (2003), 151–190.

13 Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum and Samuel J. Gershman, Building machines that learn and think like people, *Behavioral and Brain Sciences* 40, 2017.

14 Rostislav Matveev and Jacobus W. Portegies, Tropical limits of probability spaces, part I: The intrinsic Kolmogorov–Sinai distance and the asymptotic equipartition property for configurations, arXiv:1704.00297, 2017.

15 Laurie von Melchner, Sarah L. Pallas and Mriganka Sur, Visual behaviour mediated by retinal projections directed to the auditory pathway, *Nature* 404(6780) (2000), 871.

16 Pierre-Yves Oudeyer, Frédéric Kaplan and Verena V. Hafner, Intrinsic motivation systems for autonomous mental development, *IEEE transactions on evolutionary computation* 11(2) (2007), 265–286.

17 Jean Petitot, The neurogeometry of pinwheels as a sub-Riemannian contact structure, *Journal of Physiology-Paris* 97(2-3) (2003), 265–309.

18 Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, Volume 1, MIT Press, 1998.

19 Jürgen Schmidhuber, Towards solving the grand problem of AI, *Soft Computing and Complex Systems*, 2003, pp. 77–97.

20 Jürgen Schmidhuber, Gödel machines: Fully self-referential optimal universal self-improvers, in *Artificial General Intelligence*, Springer, 2007, pp. 199–226.

21 Jürgen Schmidhuber, Formal theory of creativity, fun, and intrinsic motivation (1990–2010), *IEEE Transactions on Autonomous Mental Development* 2(3) (2010), 230–247.

22 Burrhus F. Skinner, *Science and Human Behavior*, Simon and Schuster, 1953.

23 Burrhus F. Skinner, *Verbal Behavior*, Appleton-Century-Crofts, 1957.

24 A.M. Turing, Computing machinery and intelligence, *Mind* 59(236) (1950), 433.

25 Robert W. White, Motivation reconsidered: The concept of competence, *Psychological review* 66(5) (1959), 297.