On Discriminative Environments, Randomness, Two-part Compression and MML

José Hernández-Orallo

DSIC, Univ. Politècnica de València, Camí de Vera s/n, 46020 Valencia, Spain.
jorallo@dsic.upv.es

Abstract. In this paper we analyse whether there is a subclass of environments that are more discriminative for intelligence measurement. We try to characterise this class as a kind of selection of those which do not have noise or randomness. We explore such a possibility and whether it can be formalised and put into practice. In order to do this, we first introduce a simple formalisation of 'projectible' complexity which is valid for infinite strings and, consequently, for environments. From this result, we suggest an approach which both reduces the dependence on the reference machine and on the possible start-up garbage generated by an environment. More precisely, in order to avoid 'noisy' environments, especially those where 'noise' appears initially, we propose to let the environment play with a random agent for a certain number of interactions before letting the agent we want to be evaluated interact with the environment.

Keywords: Intelligence measurement, artificial intelligence, Kolgomorov complexity, two-part compression, MML, effective complexity, intensional complexity.

1 Introduction

Effective testing and evaluation of an individual's ability require an accurate choice of items in such a way that they are discriminative, in the way that they help to discern or quantify the capability which is being measured. Measuring (machine) intelligence is not different. In the late 1990s, a series of works using Kolmogorov complexity, compression, Solomonoff's prediction theory and MML inductive inference, etc., have developed or extended previous tests and definitions of intelligence (see, e.g, [3], [2] [8], [13], [11], [12] [14]). Following this line of research, [7] presents the first general and feasible test of intelligence, which should be valid for both artificial intelligent systems and biological systems, of any intelligence degree and of any speed. The test is not anthropomorphic, is gradual, is anytime and it is exclusively based on computational notions, such as Kolmogorov complexity. And it is also meaningful, since it averages the capability of succeeding in different environments. The key idea is to order all the possible action-reward-observation environments by their Kolmogorov complexity and use this ordering to make samples and construct adaptive tests that can be used to evaluate the intelligence of any kind of agent. The test configures a

new paradigm for intelligence measurement which dramatically differs from the current task-oriented and ad-hoc measurement used both in artificial intelligence and psychometrics.

One of the key issues in the previous test is the use of discriminative environments only. That means that environments which may lead to dead-ends, are too slow or that do only allow a few interactions with the agent should be ruled out. Additionally, a selection of the remaining environments must be done according to a probability distribution, since it is obviously impossible to evaluate an individual with all the (infinitely many) possible environments. The choice of the distribution is then crucial, since any biased choice would invalidate any intelligence test that claims to be universal (i.e., fair for any kind of individual).

In [7], a time-weighted version of the so-called "universal distribution" based on Kolmogorov Complexity [15] is used for the sampling of environments. This means that simple environments (in terms of Kolmogorov complexity) have higher probability of appearing in the test than more complex environments (in terms of Kolmogorov complexity).

But, what is the relation between being simple or complex (in terms of Kolmogorov complexity) and having or not a pattern such that an intelligent agent can take advantage from it? This is the very core and meaning of Kolmogorov complexity and the modern theory of randomness based upon it. Can a random environment have high probability of appearing in the test? In order to give an answer we must first realise that environments are infinite strings following the chronological enumeration of observations, actions and rewards (see [7] for details). This means that, since they are forced to iterate indefinitely, there must necessarily be a pattern in them.

The question arises when we take into account that many environments can have no pattern at all during some finite sequences of interactions. Imagine an environment which behaves randomly (as if it were noise) for the first n interactions and then starts to behave with a very simple pattern afterwards. If n is large, its Kolmogorov complexity is high, but the pattern is easy. As we discuss in [7], a low complexity implies simple patterns, and a complex pattern implies high complexity, but high complexity can be created with simple patterns + "noise"', and simple patterns can be included in high complexity environments. Is this a problem for measuring intelligence? It depends. If we do not care about knowing the real difficulty of each environment, we can accept having some environments with some degree of "noise". But if we want to adapt our test more quickly to the agent's intelligence by knowing, in principle, the difficulty of each item, we need to know the *actual* difficulty of each item. This is precisely what item-response theory does in computerised adaptive testing [16][4]. Not knowing the difficulty of an item still makes testing possible but less efficient.

So, it seems that this can easily be solved by eliminating all the environments which have noise. There are some problems here, though:

– Eliminating random behaviours (i.e. noise) from environments might imply a very important bias.

– Is it easy to tell which environments have noise and which enviroments do not?

Let us address the first item. Just think about our own world. Would you consider a world where everything is pattern a 'natural' environment? Is there any random behaviour in our world or is everything mechanical (determined from simple or complex exception-free laws)? At all levels of space and time? Even though the official scientific view is that there must be (or we must seek) underlying laws for everything, much of what is around us has a high entropy and (apparently) random behaviours, such as weather, movements, mass distributions, gases, etc. And even with a fully mechanistic assumption, time and space complexity makes it more reasonable to foresee tomorrow's weather from some abstract but imperfect meteorological principles than by computing the movements of all the atoms in the Earth's atmosphere during 24 hours. However, it is important to distinguish between a real scenario and a testing scenario. A testing scenario should present a high concentration of patterns and decisions where an intelligent agent can show its abilities. The risk here is that we can favour agents which tend to overfit, i.e. that try to see patterns where there are not (such as seeing clouds as sheep), because these agents would benefit if they expect no noise to be present. It is relevant to note that human beings are usually overfitters, because we usually see patterns everywhere.

And now let us consider the issue of whether it is possible to separate pattern from noise. In computer science, there is a common view of inductive inference as a two-part compression, where we calculate the bits which are taken to code the theory followed by the description of the evidence using the theory. Typically, the first part is the explanation or pattern, and the second part includes the exceptions or stop-rules. For instance, given the string "1010001010101010", we can describe it by "repeat 01 seven times, and then modify the fifth bit to 0". In this case the main pattern is "repeat 01" (forever), and the exceptions are (repeat only) "seven times, and then modify the fifth bit to 0". Is this separation possible in general?

In the following section we concentrate on this.

2 Splitting the Two Parts of Two-Part Compression

The Minimum Message Length (MML) principle [17] is the first theory which formalises this idea and sets out what today is known as two-part compression[1]. MML is concerned about one hypothesis which explains the data and typically separates theory from exceptions or, in other cases, theory from parameters + exceptions. But a clean distinction between main theory and exception is not a requirement for MML to work.

One of the first approaches to effecively separate pattern from noise and derive a new complexity measure was endeavoured by Gell-Mann and Lloyd [5][6],

[1] We can find other related ideas such as the Minimum Description Length principle, which appeared several years after.

with the name of "effective complexity" (as a variant of Kolmogorov complexity). Effective complexity only measures the information content of the regularities of an object. "The main idea of effective complexity is to split the algorithmic information content of some string x into two parts, its random features and its regularities. Then, the effective complexity of x is defined as the algorithmic information content of the regularities alone" [1].

Independently, the notion of intensional complexity (also as a variant of Kolmogorov complexity) was developed as a tool for making intelligent test series less dependent on the universal machine chosen, and also because "intelligence was defined as the ability to comprehend" [8][11][14], and only the regular parts can be comprehended. The idea here is based on the notion of projectibility of strings, such that given a sequence, only the regular part will project for the following elements of the sequence. In other words, the irregular part (noise) is useless for prediction.

A quite different approach (not based on Kolmogorov complexity) was taken in [12]. Here, each part of the theory gets as much as reinforcement as the times it is used for the evidence. A part of a theory must be necessary for the evidence many times and other parts of a theory are only occasionally useful. So there are different degrees of "pattern". For instance, the sequence: "print 1 when i is even and 0 otherwise, except when it is a power of 2, when 0 must be printed", shows different degrees of pattern. Since the number of powers of 2 decays very quickly, most of the bits in the sequence can be explained by the first part of the theory, and only a few account for the second part of the theory. In fact, our wording of the theory has been "[main theory], except [secondary theory]". So the notion of pattern and noise, and two-part compression seems to be more a gradual thing than a clean cut.

In fact, the notion of compression ratio is a related concept which is useful for *finite* strings. For instance, if a theory has two parts of equal size (n bits each) and one part accounts for $k \cdot n$ bits of evidence and the other for only n bits of evidence, we say that the first part has compression ratio k:1, while the second part has only a ratio of 1:1. When the compression ratios are very dissimilar, we can talk about a part being the general rule and the other part being the exception. However, there are ways to express theories such that it is very difficult to separate one part from another, so the concept of part or local compression ratio is difficult to apply. See, e.g. the notion of subpart and subprogram [9][10].

Given the previous approaches, is there a way to re-define a universal distribution so that environments with noise are ruled out?

3 Projectible Complexity

Let us try to summarise the common idea of effective complexity and intensional complexity as simply as possible. It goes as follows:

Definition 1. *Projectible Complexity. The Projectible Complexity of an infinite string x is defined as follows:*

$$\check{K}_U(x) = min\{l(p) : U(p) = x' \ such \ that \ \exists n \ : x'_{n\cdots\infty} = x_{n\cdots\infty}\}.$$

This means that the projectible complexity of a string is the shortest description which produces any right-hand cut of the string. For instance, if we have the string 10110100100011011111111111111111111111111111111111111111111..., its complexity is equal to that of the string 11111111111111111111111111..., since the initial part (the first 15 bits of the first string) is considered noise (and with n=16 we set the equality). In other words, since the projectible complexity is finite, any randomness which is included in the description must be used sooner or later, and then the string must be stable in the limit. By stable, we mean that its pattern is clean (no noise). In order to see more examples, consider the string AB where A is a part full of noise and random bits and B is an infinite part without noise and with a complex pattern (e.g. the digits of π). Then, the projectible complexity of AB completely disregards A and only accounts for B, the regular part. Now, consider a string CAB where C is a regular but finite pattern, A is a random/noisy part and C is a regular part. Then the projectible complexity of CAB equals the projectible complexity of B. Now, consider the string AB where A is a sequence of one million 1s and B is an infinite sequence of pairs 01. Then the projectible complexity completely ignores the first part, even though it is highly compressible. Note that in relative terms, its compression ratio is much lower than the compression ratio of the rest of the string (infinite ratio).

There are some interesting properties about the previous definition. The first issue is that the previous definition only applies for infinite strings, because for finite strings this n would be chosen as the length of the string and the definition would not work. This is the reason why all the previous approaches [5][6][1][11][12] are so complex, while this previous definition is so simple.

The good news is that environments are infinite. That means that we can apply the previous definition to environments and assign probabilities according to the formula $p(x) = 2^{-\check{K}_U(x)}$. In fact, the notion of projectible complexity makes much more sense for environments if we consider that some of the initial interactions might be some kind of start-up process (or boot-up garbage) and only after several interactions the environment starts to behave regularly.

And finally, in the case of environments, we avoid a so-called problem of "aging environments", where we may have environments which are rich and complex initially (either by random content or because of a pattern) and then gradually or abruptly evolve into very simple environments. The use of Kolmogorov complexity does not avoid this.

4 Using Projectible Complexity to Select Environments

According to the previous discussion, we may propose the following methodology to make a sample of environments.

1. Sample environments according to the probability $p(x) = 2^{-\check{K}_U(x)}$ or something similar.

2. Use a random agent to play with the environment during a number of interactions greater or equal to $\breve{K}_U(x)$.

The first item has been justified in the introduction. Only the pattern, i.e., the discriminative part, should be used to order the environments by their difficulty. The second item is motivated by the use of the previous sample. Since environments with long random start-up content have low projectible complexity, we must start interacting with the environment once this garbage content has been surpassed.

The problem of the previous procedure is that the string AB where A is a billion-bit long random string and B is just an infinite sequence of 1s has a very low projectible complexity and we would need a long number of interactions before reaching the stable part of the environment.

A possibility would be to try to rule out these strings and only get the noise-free ones, but we are still in the same problem of making a selection and, due to the halting problem, this is not computable.

Consequently, we propose the alternative procedure below:

1. Sample environments according to the probability $p(x) = 2^{-K_U(x)}$ (or a resource-bounded approximation).
2. For each selected environment, use a random agent to play with the environment during a number of interactions greater or equal to $K_U(x)$ (start-up period).
3. Consider $\breve{K}_U(x)$ (and not $K_U(x)$) as the actual difficulty of the item.
4. Administer an environment of appropriate difficulty to the agent starting after the start-up period.

With this, we generally avoid start-up behaviours or abrupt changes of patterns, since the agent we want to evaluate starts playing with the environment after a sort of testing period so it typically starts a more stable part of the environment. How long this start-up period should be is a question, although our choice is that it should be much greater than (and proportional to) $K_U(x)$.

5 Conclusions

In this paper we have investigated whether it is possible to tell pattern and noise apart in the context of Kolmogorov complexity. The ultimate goal is its use for making samples of discriminative environments which are eventually used for measuring intelligence. We have seen that this problem is quite slippery, especially for finte strings. For infinite strings in general, and environments in particular, we have presented a very straightforward formalisation which allows us to define a procedure such that we can evaluate agent performance on the regular parts of the environments, avoiding noisy and random parts (especially start-up garbage), which are less discriminative. Additionally, this choice makes the test less dependent on the reference machine used.

6 Acknowledgements

I am grateful to David L. Dowe for reading a draft of this paper and spotting several mistakes and typos.

References

1. N. Ay, M. Mueller, and A. Szkola. Effective complexity and its relation to logical depth. *ArXiv e-prints*, October 2008.
2. D. L. Dowe and A. R. Hajek. A non-behavioural, computational extension to the Turing Test. In *Intl. Conf. on Computational Intelligence & multimedia applications (ICCIMA'98), Gippsland, Australia*, pages 101–106, 1998.
3. D.L. Dowe and A.R. Hajek. A computational extension to the Turing test. In *Proceedings of the 4th Conference of the Australasian Cognitive Science Society, Newcastle, NSW*, 1997.
4. S. Embretson and S. Reise. *Item response theory for psychologists*. Mahwah, NJ: Erlbaum, 2000.
5. Murray Gell-Mann and Seth Lloyd. Information measures, effective complexity, and total information. *Complexity*, 2(1):44–52, 1996.
6. Murray Gell-Mann and Seth Lloyd. Effective complexity. In *Santa Fe Institute, http://www.santafe.edu/research/publications/workingpapers/03-12-068.pdf*, pages 387–398, 2003.
7. J. Hernández-Orallo and D. L. Dowe. Measuring universal intelligence: Towards an anytime intelligence test. *Artificial Intelligence*, 174(18):1508 – 1539, 2010.
8. J. Hernández-Orallo and N. Minaya-Collado. A formal definition of intelligence based on an intensional variant of Kolmogorov complexity. In *Proceedings of the International Symposium of Engineering of Intelligent Systems (EIS'98)*, pages 146–163. ICSC Press, 1998.
9. José Hernández-Orallo. Universal and cognitive notions of part. In *Ferrer, L. et al. (eds.), Proceedings of 4th Systems Science European Congress*, pages 711–722, 1999.
10. José Hernández-Orallo. What is a subprogram? In *Technical Report, http://users.dsic.upv.es/∼jorallo/escrits/subprog.ps.gz*, 1999.
11. José Hernández-Orallo. Beyond the Turing test. *Journal of Logic, Language and Information*, 9(4):447–466, 2000.
12. José Hernández-Orallo. Constructive reinforcement learning. *International Journal of Intelligent Systems*, 15(3):241–264, 2000.
13. José Hernández-Orallo. On the computational measurement of intelligence factors. In *Performance metrics for intelligent systems workshop*, pages 1–8. Gaithersburg, MD, 2000.
14. José Hernández-Orallo. Thesis: Computational measures of information gain and reinforcement in inference processes. *AI Communications*, 13(1):49–50, 2000.
15. Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications (3rd ed.)*. Springer-Verlag New York, Inc., 2008.
16. F.M. Lord. *Applications of item response theory to practical testing problems*. Mahwah, NJ: Erlbaum, 1980.
17. Chris S. Wallace and David M. Boulton. A information measure for classification. *The Computer Journal*, 11(2):185–194, 1968.