A (hopefully) Unbiased Universal Environment Class for Measuring Intelligence of Biological and Artificial Systems

José Hernández-Orallo
DSIC, Univ. Politècnica de València,
Camí de Vera s/n, 46020 Valencia, Spain. joralloc@dsic.upv.es

Abstract

The measurement of intelligence is usually associated with the performance over a selection of tasks or environments. The most general approach in this line is called Universal Intelligence, which assigns a probability to each possible environment according to several constructs derived from Kolmogorov complexity. In this context, new testing paradigms are being defined in order to devise intelligence tests which are anytime and universal: valid for both artificial intelligent systems and biological systems, of any intelligence degree and of any speed. In this paper, we address one of the pieces in this puzzle: the definition of a general, unbiased, universal class of environments such that they are appropriate for intelligence tests. By appropriate we mean that the environments are discriminative and that they can be feasibly built, in such a way that the environments can be automatically generated and their complexity can be computed.

Introduction

This paper presents a feasible environment class which can be used to test intelligence of humans, non-human animals and machines. The environment class is developed under the theory presented in (HOD09), which is, in turn, based on (LH07)(HO00)(DH97). This theory presents the first general and feasible intelligence test framework, which should be valid for both artificial intelligent systems and biological systems, of any intelligence degree and speed. The test is not anthropomorphic, is gradual, is anytime and 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 to use this ordering to make a sample. In order to make this feasible (in contrast to (LH07)), in (HOD09) several constraints are imposed on the environments: (1) time is considered, (2) a time-bounded and computable version of Kolmogorov complexity is used, (3) rewards must be balanced, and (4) environments must be sensitive to the agent actions. The environments can then be used to construct adaptive (anytime) tests to evaluate the intelligence of any kind of agent. The test configures a new paradigm for intelligence measurement which dramatically differs from the current specific-task-oriented and ad-hoc measurement used both in artificial intelligence and psychometrics.

The previous theory, however, does not make the choice for *an* environment class, but just sets some constraints on the kind of environments that can be used. Consequently, one major open problem is to make this choice, i.e., to find a proper (unbiased and feasibly implementable) environment class which follows the constraints. Once this environment class is identified, we can use it to generate environments to run any of the tests variants introduced in (HOD09).

One recurrent problem is that the reference machine for environments is necessarily an arbitrary choice even though Kolmogorov Complexity only differs in a constant when using two different reference machines. But the constant (especially for short tests) is important, since using a specific universal machine could, in the end, constitute a strong bias for some subjects.

Another problem of using an arbitrary universal machine is that this machine can generate environments which are not discriminative. By discriminative we mean that there are different policies which can get very different rewards and, additionally, these good results are obtained by competent agents and not randomly. Note that if we generate environments at random without any constraint, we have that an overwhelming majority of environments will be completely useless to discriminate between capable and incapable agents, since the actions can be disconnected with the reward patterns, with reward being good (or bad) independently of what the agent does.

In (HOD09) a set of properties which are required for making environments discriminative are formally defined, namely that observations and rewards must be sensitive to agent's actions and that environments are balanced, i.e. that a random agent scores 0 in these environments (when rewards range from -1 to 1). This is crucial if we take time into account in the tests because if we leave a finite time to interact with each environment and rewards go between 0 and 1, a very proactive but little intelligent agent could score well (for a thorough discussion on this see (HO09b)). Given these constraints, if we decide to generate environments without any constraint and then try to make a post-processing sieve to select which of them comply with all the constraints, we will have a computationally very expensive (or even incomputable) problem. So, the approach taken in this paper is to generate an environment class that ensures that these properties hold. But, we have to be very careful, because we would not like to restrict the reference machine to comply with these properties at the cost of losing their universality (i.e. their ability to emulate or include any computable function).

And finally, we would like the environment class to be user-friendly to the kind of systems we want to be evaluated (humans, non-human animals and machines), but without any bias in favour or against some of them.

According to all this, in this paper we present an operational way to define a universal environment class from which we can effectively generate valid environments, calculate their complexity and consequently derive their probability.

Definition of the Environment Class

The environment class is composed of a cell space and a set of objects that can move inside the space. In this short note, we only enumerate the most relevant traits of the class. For a more formal and complete definition of the class, we refer to (HO09a).

- **Space**: The space is defined as a directed labelled graph of nodes (or vertices), where each node represents a cell and arcs represent actions. The topology of the space can be quite varied. It can include a typical grid, but much more complex topologies too.
- **Objects**: Cells can contain objects. Objects can have any behaviour (deterministic or not), always under the space topology, can be reactive to other agents and can be defined to act with different actions according to their observations. Objects perform one and only one action at each interaction of the environment (except from the special objects Good and Evil, represented by $\oplus$ and $\ominus$ respectively, which can perform several actions in a row). Good and Evil must have the *same* behaviour.
- **Observations and Actions**: Actions allow the evaluated agent (denoted by π) to move in the space. Observations show the (adjacent) cell contents.
- **Rewards**: We will work with the notion of trace and the notion of "cell reward", that we denote by $r(C_i)$. Initially, $r(C_i) = 0$ for all i. Cell rewards are updated by the movements of $\oplus$ and $\ominus$. At each interaction, we set 0.5 to the cell reward where $\oplus$ is and -0.5 to the cell reward where $\ominus$. Each interaction, all the cell rewards are divided by 2. So, an intuitive way of seeing this is that $\oplus$ leaves a positive trace and $\ominus$ leaves a negative trace. The agent π *eats* the rewards it finds in the cells it occupies, updating the accumulated reward $\rho = \rho + r(C_i)$.

The previous environment class is sensitive to rewards and observations (the agent can perform actions in such a way that can affect the rewards and the observations), and it is also balanced (a random agent would have an expected accumulated reward equal to 0). For the formal definition of these properties, see (HOD09). For the proofs of the these properties see (HO09a). These properties make the environments suitable for an anytime test (HOD09).

Spaces and objects are coded with Markov algorithms (Turing-complete), their complexity computed and their probability derived. See (HO09a) for details.

Conclusions

Some choices made in this paper can obviously be improved, and better classes might be more elegantly defined. However, to our knowledge, this is the first attempt in the direction of setting a general environment class for intelligence measurement which can be effectively generated and coded.

The main idea for the definition of our environment class has been to separate the space from the objects, and two special symmetric objects are in charge of the rewards, in order to define a class which only includes observation and reward-sensitive environments which are balanced. The space sets some common rules on actions and the objects may include any universal behaviour. This opens the door to social environments.

Acknowledgments

The author thanks the funding from the Spanish Ministerio de Educación y Ciencia (MEC) for projects EXPLORA-INGENIO TIN2009-06078-E, CONSOLIDER-INGENIO 26706 and TIN 2007-68093-C02, and GVA project PROMETEO/2008/051.

References

D.L. Dowe and A.R. Hajek. A computational extension to the turing test. In *in Proc. of the 4th Conf. of the Australasian Cognitive Science Society*, 1997.

J. Hernández-Orallo. Beyond the turing test. *J. of Logic, Language and Information*, 9(4):447–466, 2000.

J. Hernández-Orallo. A (hopefully) unbiased universal environment class for measuring intelligence of biological and artificial systems. Extd. Version. *available at http://users.dsic.upv.es/proy/anynt/*, 2009.

J. Hernández-Orallo. On evaluating agent performance in a fixed period of time. Extd. Version. *available at http://users.dsic.upv.es/proy/anynt/*, 2009.

J. Hernández-Orallo and D.L. Dowe. Measuring universal intelligence: Towards an anytime intelligence test. *Under Review, http://users.dsic.upv.es/proy/anynt/*, 2009.

S. Legg and M. Hutter. Universal intelligence: A definition of machine intelligence. *Minds and Machines*, 17:391–444, 2007.