Measuring universal intelligence: Towards an anytime intelligence test


José Hernández Orallo¹, David L. Dowe²

1. Departament de Sistemes Informàtics i Computació, Universitat Politècnica de València, Spain. jorallo@dsic.upv.es.

2. Computer Science & Software Engineering, Clayton School of I.T., Monash University, Clayton, Victoria, 3800, Australia. david.dowe@infotech.monash.edu.au.
Outline

- Towards a universal intelligence test
- Precedents
- Addressing the problems of universal intelligence
- An anytime test
- Instances and implementation
- Conclusions and future work
Towards a universal intelligence test

Evaluating intelligence. Some issues:

1. Harder the less we know about the examinee.

2. Harder if the examinee does not know it is a test.

3. Harder if evaluation is not interactive (static vs. dynamic).

4. Harder if examiner is not adaptive.
Towards a universal intelligence test

State of the art: different subjects, different tests.

- IQ tests:
  1. Human-specific tests. Natural language assumed.
  2. The examinees know it is a test.
  4. Generally non-adaptive (pre-designed set of exercises)
- Other tests exist (interviews, C.A.T.)

- Turing test:
  1. Held in a human natural language.
  2. The examinees ‘know’ it is a test.
  3. Interactive.
- Other task-specific tests exist.
  • Robotics, games, machine learning.

- Children’s intelligence evaluation:
  1. Perception and action abilities assumed.
  2. The examinees do not know it is a test. Rewards are used.
  3. Interactive.
  4. Frequently non-adaptive (pre-designed set of exercises).

- Animal intelligence evaluation:
  1. Perception and action abilities assumed.
  2. The examinees do not know it is a test. Rewards are used.
  3. Interactive.
  4. Generally non-adaptive (pre-designed set of exercises).
Towards a universal intelligence test

Can we construct a test for all of them?

- Without knowledge about the examinee,
- Derived from computational principles,
- Non-biased (species, culture, language, etc.)
- No human intervention,
- Producing a score,
- Meaningful,
- Practical, and
- **Anytime.**

Is this possible?

- No previous measurement or test of intelligence presented to date fulfils all of these requirements.
Towards a universal intelligence test

Project: **anYnt** *(Anytime Universal Intelligence)*
http://users.dsic.upv.es/proy/anynt/

- **Any** kind of system (biological, non-biological, human)
- **Any** system now or in the future.
- **Any** moment in its development (child, adult).
- **Any** degree of intelligence.
- **Any** speed.
- Evaluation can be stopped at **any** time.
Turing Test (Turing 1950): anytime and adaptive.

- It is a test of humanity, and needs human intervention.
- Not actually conceived to be a practical test to measure intelligence up to and beyond human intelligence.

- Exercises (series) are not arbitrarily chosen.
- They are drawn and constructed from a universal distribution:

  \[
  \begin{align*}
  k = 9 & : \quad a, \, d, \, g, \, j, \ldots \quad \text{Answer : m} \\
  k = 12 & : \quad a, \, a, \, z, \, c, \, y, \, e, \, x, \ldots \quad \text{Answer : g} \\
  k = 14 & : \quad c, \, a, \, b, \, d, \, b, \, c, \, c, \, e, \, c, \, d, \ldots \quad \text{Answer : d}
  \end{align*}
  \]

  **Fig. 2.** Examples of series of $Kt$ complexity 9, 12, and 14 used in the C-test [7].

- However, some relatively simple agents can cheat on them (Sanghi and Dowe 2003) and they are static (no planning abilities are required).
Captchas (von Ahn, Blum and Langford 2002): quick and practical, but strongly biased. They soon become obsolete.

A strong impact in real applications and in the scientific community.

But...

They are not conceived to evaluate intelligence, but to tell humans and machines apart at the current state of AI technology.

It is widely recognised that CAPTCHAs will not work in the future.
- **Universal Intelligence** (Legg and Hutter 2007): an interactive extension to C-tests from sequences to environments.

\[
\gamma(\pi, U) := \sum_{\mu=i}^{\infty} p_U(\mu) \cdot V_\mu = \sum_{\mu=i}^{\infty} p_U(\mu) \cdot E\left(\sum_{i=1}^{\infty} r_{i,\mu,\pi}\right)
\]

= performance over a universal distribution of environments.

- **Obvious Problems:**
  - \(U\) is a choice which defines the environment class.
  - The probability distribution is not computable.
  - There are two infinite sums (number of environments and interactions).
  - Time/speed is not considered for the environment or for the agent.

- **Other less obvious problems.**
A definition of intelligence does not ensure an intelligence test.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Intelligence tests in passive and active environments (clarification).</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Universal agent</td>
</tr>
<tr>
<td>Passive environment</td>
<td>Solomonoff prediction</td>
</tr>
<tr>
<td>Active environment</td>
<td>AIXI</td>
</tr>
</tbody>
</table>

The C-test used Solomonoff’s theory of inductive inference (predictive learning) to define an inductive inference test.

Universal intelligence provides a definition which adds interaction and the notion of “planning” to the formula (so intelligence = learning + planning).

For “Universal Intelligence” we will have to “redefine” it, and then to think about how to use it to construct a feasible test.
Addressing the problems of universal intelligence

- **On the difficulty of environments:**
  - Very simple environments are given a very high probability
  
  \[
  K_U(x) := \min_{p \text{ such that } U(p)=x} l(p) \\
  p_U(x) := 2^{-K_U(x)}
  \]

- Most of the score will come from very simple environments.
  - E.g. The 256 environments with \( K \leq 8 \) accumulate a probability of 0.996 (and hence weight, i.e., score) in the definition.

  \[
  \mathcal{Y}(\pi, U) := \sum_{\mu=1}^{\infty} p_U(\mu) \cdot V_\mu^{\pi}
  \]

- Since we don’t have any information about the examinee, we cannot set any limit (or *soften* the distribution).
  - one solution is to make the test adaptive.
Selecting discriminative environments:

Many environments will be completely useless to evaluate intelligence, because:

- Rewards may be independent of agent actions.
- There must be sequences of actions that lead to unrecoverable “states”. We cannot assume environments to be ergodic.
- Some environments may be highly benevolent (high expected rewards) and some others can be very malevolent (low expected rewards).

We introduce two constraints on environments:

- Environments must be reward-sensitive: *an agent must be able to influence rewards at any point.*
- Environments must be balanced: *a random agent must have an expected reward of 0 (with rewards ranging between -1 and 1).*
On practical interactions:

- We have to consider that environments should react almost immediately. We modify the universal distribution as follows:

**Definition 9** \((K_t \text{ complexity weighting interaction steps})\).

\[
K_{t_{ij}}^{\text{max}}(\mu, n) := \min_{p \text{ such that } U(p) = \mu} \left\{ l(p) + \log \left( \max_{a_{1, i}, 1 \leq n} (\Delta \text{time}(U, p, a_{1, i})) \right) \right\}
\]

- The use of a parameter \(n\) makes the definition computable.

- From here, we re-define the distribution:

\[
p_{t_{ij}}^\mu(\mu) := 2^{-K_{t_{ij}}^{\text{max}}(\mu, n_i)}
\]

- And now:
  - We create a finite sample of environments.
  - We also use a limited number of interactions for each environment.
Time and intelligence:

- We must consider fast but unintelligent agents as well as slow and intelligent ones.
  - But we cannot make these two things independent.
  - Otherwise, intelligence would be computationally easier than it is.

- A way to do that is to set a finite period of time for each environment instead of a “number of interactions”.
  - Speed will be important because it will increase both exploration and exploitation possibilities.
  - In fact, agent’s speed will be very relevant.
  - But, it is crucial to consider balanced environments.
Reward aggregation:

- Can we use RL aggregation measures such as accumulated reward and general discounting?
  - We show they present important caveats when measuring agents:
    - with a finite (previously unknown) period of time,
  - Why?
    - Given an evaluation time $\zeta$, a fast agent could act randomly and get a good accumulated score and then rest on its laurels.
    - These are called “stopping” policies in games.
- We introduce [48] a new measure for aggregating rewards in a given time $\zeta$, where “discounting” is made to be robust to delaying and stopping policies.

**Definition 16** *(Average reward with diminishing history)*:

$$
\bar{\mathcal{r}}_{\mu, \pi}^n := \frac{1}{n^*} \sum_{k=1}^{n^*} r_{k, \pi}
\quad \text{where } n^* = \left\lceil n_\tau \left( \frac{t_{n_\tau}}{\tau} \right) \right\rceil
$$
Given all the previous constraints and modifications we can give a definition, which is useful for a test.

**Definition 17 (Universal intelligence considering time (finite set of reward-sensitive and balanced environments, finite number of interactions, \( K_U^{\text{max}} \) complexity) with adjusted score and using physical time to limit interactions).**

\[
\mathcal{T}^{iv}(\pi, U, m, n_i, \tau) := \frac{1}{m} \sum_{\mu \in S} \hat{W}_{\mu}^{\tau} \| \tau
\]

where \( S \) is a finite subset of \( m \) balanced environments that are also \( n_i \)-actions reward-sensitive, \( S \) is extracted with 
\[
p_U^i(\mu) := 2^{-K_U^{\text{max}}(\mu, n_i)}.
\]

The definition is parameterised by the **number of environments** \( m \) and the **time limit** for each of them \( \zeta \).

- The higher \( m \) and \( \zeta \) are, the better the assessment is expected to be.
- For a new (unknown) agent, it is difficult to tell the appropriate \( m \) and \( \zeta \).
An anytime test

Definition 18 (Anytime universal intelligence test taking time into account). We define $\gamma^\gamma(\pi, U, H, \Theta)$ as the result of the following algorithm, which can be stopped anytime:

1. **ALGORITHM**: Anytime Universal Intelligence Test
2. **INPUTS**: $\pi$ (an agent), $U$ (a universal machine), $H$ (a complexity function), $\Theta$ (test time, not as a parameter if the test is stopped anytime)
3. **OUTPUTS**: a real number (approximation of the agent’s intelligence)
4. **BEGIN**
5. $\gamma \leftarrow 0$ (initial intelligence)
6. $\tau \leftarrow 1$ microsecond (or any other small time value)
7. $\xi \leftarrow 1$ (initial complexity)
8. $S_{\text{used}} \leftarrow \emptyset$ (set of used environments, initially empty)
9. **WHILE** (TotalElapsedTime $< \Theta$) **DO**
10. **REPEAT**
11. $\mu \leftarrow \text{Choose}(U, \xi, H, S_{\text{used}})$ (get a balanced, reward-sensitive environment with $1 \leq H \leq \xi$ not already in $S_{\text{used}}$)
12. **IF** (NOT FOUND) **THEN**
13. $\xi \leftarrow \xi + 1$ (all of them have been used already)
14. **ELSE**
15. **BREAK REPEAT** (we can exit the loop and go on)
16. **END IF**
17. **END REPEAT**
18. $\text{Reward} \leftarrow V_{\mu}^\tau \parallel \tau$ (average reward until time-out $\tau$ stops)
19. $\gamma \leftarrow \gamma + \text{Reward}$ (adds the reward)
20. $\xi \leftarrow \xi + \xi \cdot \text{Reward}/2$ (updates the level according to reward)
21. $\tau \leftarrow \tau + \tau/2$ (increases time)
22. $S_{\text{used}} \leftarrow S_{\text{used}} \cup \{\mu\}$ (updates set of used environments)
23. **END WHILE**
24. $\gamma \leftarrow \gamma / |S_{\text{used}}|$ (averages accumulated rewards)
25. **RETURN** $\gamma$
26. **END ALGORITHM**
Implementation of the anytime test requires:

- To define an environment class $U$ (e.g., a Turing-complete machine), where all the environments are balanced and reward-sensitive (or define a computable, preferably efficient, sieve to select them).
- A complexity function (e.g., $K_{\text{max}}$)

Several environment classes may determine general or specific performance tests:

- In [53] we have presented a Turing-complete environment class $\Lambda$ which is balanced and reward-sensitive.

Other specific classes can be used to evaluate subfields of AI:

- If $U$ is chosen to only comprise static environments, we can define a test to evaluate performance on sequence prediction (for machine learning).
- If $U$ is chosen to be games (e.g. using the Game Description Language in the AAAI General Game Playing Competition), we have a test to evaluate performance on game playing.
- Similar things can be done with the reinforcement learning competition, maze learning, etc.
Since the late 1990s, we have derived several general intelligence tests and definitions with a precise mathematical formulation.

Algorithmic Information theory (a.k.a. Kolmogorov complexity) is the key for doing that.

The most important conclusions of this work are:

We have shaped the question of whether it is possible to construct an intelligence test which is universal, formal, meaningful and anytime.

We have identified the most important problems for such a test:

- the notion of environment complexity and an appropriate distribution,
- the issue that many environments may be useless for evaluation (not discriminative),
- a proper sample of environments and time slots for each environment,
- computability and efficiency,
- time and speed for both agent and environment,
- evaluation (reward aggregation) in a finite period of time,
- the choice of an unbiased environment.
This proposal can obviously be refined and improved:

- The use of balanced environments and the character of the anytime test suggest that for many (Turing-complete) environment classes, the measure is convergent, but this should be shown theoretically or experimentally.
- $K_t^{\text{max}}$ needs a parameter to be computable. Other variants might exist without parameters (e.g., using the speed prior).
- The probability of social environments (other intelligent agents inside) is almost 0. A complexity measure including other agents could be explored.

**Implementation:**

- Currently implementing an approximation to the test using the environment class $\Lambda$.
- Also considering implementing an approximation using the GDL (Game Description Language) as environment class.

**Experimentation:**

- On AI agents (e.g. RL Q learning, AIXI approximations, etc.), humans, non-human animals, children.