One Decade of Universal Artificial Intelligence

a

27. September 2015

Abstract

Artificial Intelligence is a widely investigated area with the grand goal to create an artificial system which exhibits general intelligence on or beyond human-level. Such a system should be able to learn fast and be independent of any parameters or environments.

The theory of Universal Artificial Intelligence (UAI), which emerged in the beginning of this century, is a top-down approach to the problem of general intelligence. In this work we will present a sound and complete mathematical model for a super intelligent agent (AIXI) which can be embedded in any unknown environment and possesses all aspects of rational intelligence, i.e. it is able to self-adapt to a diverse range of interactive environments [2].

Furthermore, several approximations of AIXI will be introduced which have been successfully implemented. They demonstrate that AIXI is able to *learn* from scratch to play games like TicTacToe, Pacman etc. without even providing any rules.

This seminar paper is mainly based on the work of Marcus Hutter [2–5].

1 Introduction

Plenty of research has been invested in studying the human mind and understanding the traits of intelligence. The field of Artificial Intelligence (AI) is responsible for studying and creating artificial systems which exhibit intelligent behavior. The dream of AI is to develop systems that possess general intelligence on a par with the human mind or even beyond. Many theories and approaches have been constructed in the area of Artificial General Intelligence (AGI).

Nowadays most results in the field of AI are limited in the area of application, e.g. logical/probabilistic systems which are utilized for pattern recognition [1], or neural networks being used to learn specific tasks. Our goal is to provide an intelligent system which has the ability to learn and adapt to general tasks and a wide range of environments. A further issue with AGI is the absence of a general definition or theory of "intelligence". Assume the availability of unlimited computational resources. Then solving NP-complete problems or e.g. playing chess optimally becomes trivial,

but managing practical tasks like driving a car or surviving in nature is still a hard problem. The reason is the complexity of well-defining the latter problems, not to mention presenting an algorithm. Thus a formal mathematical definition of intelligence could provide a basis for further work on AGI.
In the first decade of this century a new theory of *Universal Artificial Intelligence*, a modern information-theoretic inductive approach, has been developed which may provide a formal definition to the latter problem. Furthermore, the model provides a universally optimal rational agent AIXI, i.e. an parameterless, model-free agent which theoretically learns faster compared to other possible agents. It is able to self-adapt to unknown environments and learn from scratch by reinforcement. A major drawback of AIXI is that it is incomputable, thus requiring effective approximations to be implementable.
In the following chapter we will first introduce basic notions and definitions of an agent model, followed by an optimal model for known computable environments. In chapter 3, the universal agent AIXI is presented, which is provably optimal and fully rational. Afterwards, a few approximations of AIXI are listed, including the AIXI*tl* agent, which has a runtime of $O(t \cdot 2^l)$. Finally, the results of this work are summarized in section 4.

2 Preliminaries

This chapter presents the basic definitions of the Universal Agent Model [9]. First, we formalize the notion of a rational agent interacting with some environment. Afterwards, an optimal agent will be presented w.r.t maximizing his own reward given a known computable environment. We will then modify this agent and introduce several ingredients to derive a model which acts optimally even if the environment is unknown.

2.1 Rational Agent Model

Let $\mathcal{X} := \mathcal{O} \times \mathcal{R}, \mathcal{R} \subset \mathbb{R}$ denote the input alphabet and $\mathcal{Y}$ be the output alphabet, where $\mathcal{O}$ and $\mathcal{Y}$ can be arbitrary sets of symbols. We then define the Agent Model, consisting of an *agent* and *environment* interacting with each other in (possibly infinitely many) *cycles*, as follows:

Definition 1. *Agent Model*
In each cycle k the agent performs an action $y_k \in \mathcal{Y}$ depending on all previous observations: $y_k = p(y_1 x_1 ... y_{k-1} x_{k-1})$. Afterwards the environment performs a reaction $o_k r_k =: x_k \in \mathcal{X}$, where $x_k = q(y_1 x_1 ... y_{k-1} x_{k-1} y_k)$.

In the following we will use the term “output” synonymously for the agent’s action (resp. “input” for the environment’s reaction). The output x_k consists of an actual output o_k and a real-valued *reward* r_k, which serves as a feedback mechanism for the agent’s decisions (higher reward means “better” decisions). Furthermore, the input function q can either be a *deterministic function* $q(y_1x_1...y_{k-1}x_{k-1}y_k)$ or more generally a *probability distribution* $\mu(y_1x_1...y_{k-1}x_{k-1}y_k)$. In particular, if we define $\mathcal{X}$ and $\mathcal{Y}$ as strings over an alphabet, then p and q can be interpreted as Turing machines with unidirectional input/output tapes (Fig. 1).
In order to keep the definitions clear, we define the following notation: A string s shall be expressed as the concatenation of other strings $s = x_1x_2...x_n$. Furthermore, we have $x_{n:m} := x_nx_{n+1}...x_{m-1}x_m$, $x_{<n} := x_1...x_{n-1}$, $yx_n := y_nx_n$ and $yx_{n:m} := y_nx_n...y_mx_m$.

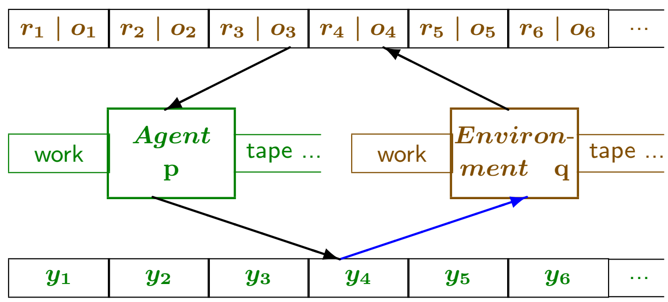

Figure 1: Illustration of the Agent Model, consisting of agent and environment interacting with each other

For the rest of this subsection we will assume that the environment is known to the agent, i.e. the agent can calculate the probability distribution $\mu(y_1x_1...y_{k-1}x_{k-1}y_k)$ for arbitrary values $y_1x_1...y_{k-1}x_{k-1}y_k$ (this also includes the case where the enviroment is calculated by a deterministic function, since deterministic functions can be expressed as probability distributions with the only probabilities 0 and 1). Our goal is now to find an agent which maximizes the *expected* future reward. We define the *value* of an agent p in environment μ as follows:

$$V_\mu^p := \sum_{x_{1:m}} (r_1 + ... + r_m)\mu(yx_{1:m})$$

where $y_{1:m} = p(x_{<m})$ denotes the output of p. The variable m is called the “horizon” and determines the lookahead of the agent. V_μ^p describes the weighted sum of the rewards over all inputs $x_{1:m}$ depending on the outputs $y_{1:m}$ of p.
We can now define the AIμ agent:

Definition 2. *AIμ agent*
The AIμ agent is, given μ, the agent with the policy p^μ which maximizes the μ-expected total reward:

$$p^\mu := \arg\max_p V_\mu^p$$

From this definition we can derive the optimal action of the AIμ agent for each cycle k:

$$y_k^{\mu} := \arg\max_{y_k} \sum_{x_k} \max_{y_{k+1}} \sum_{x_{k+1}} ... \max_{y_m} \sum_{x_m} (r_k + ... + r_m)\mu(yx_{k:m}|yx_{<k})$$

Hence, the optimal agent performs an expectimax algorithm for the next $m-k$ cycles and chooses the output y_k with the highest expected conditional reward, given the history $yx_{<k}$ (Fig. 2).

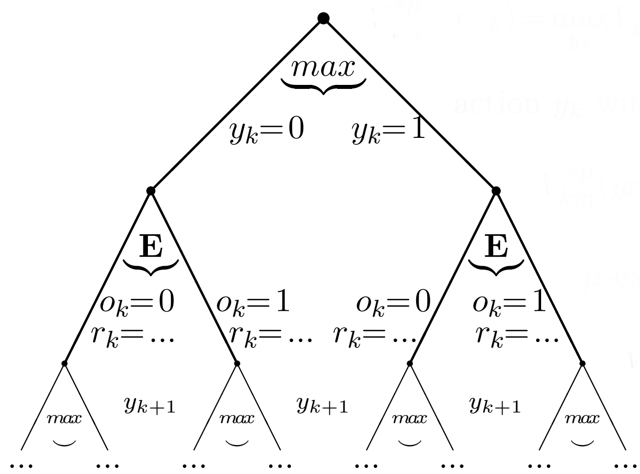

Figure 2: Illustration of the expectimax algorithm for $\mathcal{X} = \mathcal{O} = \{0, 1\}$.

2.2 Universal Induction

In the last section it was shown that we can explicitly construct an agent which performs the optimal action with respect to the horizon m, if the environment is *known*. What about *unknown* environments, i.e. if μ is not given?
This section will discuss a solution for unknown environments by presenting a global approximation for the probability distribution μ. The underlying concept is based on the theory of universal induction by Solomonoff [10,11].

Definition 3. *Inductive Inference*
Given data points x_t s.th. $t < n$, the task is to predict the next data point x_n from the sequence $x_1, ..., x_{n-1}$.

First, we recall Occam's Razor, a well-known principle of problem solving:

Proposition 1. *Occam's Razor*
Take the simplest hypothesis consistent with the data.

Thus, given arbitrary data points, Occam's Razor suggests to assume the simplest possible model for these data and make the next prediction based on that model. In order to accomplish this task, a measure of *complexity* is necessary. We use universal Turing machines to measure the complexity of arbitrary strings [7]:

Definition 4. *Kolmogorov complexity*
Let U be a universal prefix Turing machine, i.e. a TM with unidirectional input/output tapes. The Kolmogorov complexity of a string s is defined as

$$K(s) = \min_p \{l(p) : U(p) = s\}$$

The Kolmogorov complexity of a string s is length of the smallest Turing machine description which will terminate with s on the output tape. We note that $K(s)$ is not computable.
Now, by combining both Occam's Razor and the Kolmogorov complexity, the first attempt to define a probability distribution could be as follows:

$$P(s) := 2^{-K(s)},$$

where s is an arbitrary string. Thus, the simplest string with respect to the Kolmogorov complexity will be predicted with the highest probability. There are two issues with this approach: First, $P()$ is not a normalized probability distribution, and second, $P(s)$ is not computable, since neither $K(s)$ is.
Furthermore, another problem-solving principle, would contradict to this approach:

Proposition 2. *Epicurus' Principle of Indifference*
If more than one hypothesis is consisten with the observed data then keep all hypotheses.

Another approach is based on Solomonoff and is known as the *universal prior*:

Definition 5. *Solomonoff's universal prior*
Let U be a universal prefix TM. Solomonoff's universal prior is defined as

$$M(s) := \sum_{p:U(p)=s*} 2^{-l(p)}$$

Thus, given a string s, we sum up the weighted lengths of each TM description whose output starts with s (hence we respect both Occam's Razor and Epicurus' Principle). Alternatively, $M(s)$ can be interpreted as the probability that we obtain a (binary) description of a TM 1which outputs a string starting with s by flipping fair coins.
One can show that the following inequality holds [3]:

$$\sum_{t=1}^{\infty}(1 - M(x_t|x_{<t}))^2 \leq \frac{1}{2} \ln 2 \cdot K(x_{1:\infty})$$

It follows that if the environment μ is computable, i.e. $K(x_{1:\infty}) < \infty$ (there is a TM which outputs the perceptions $x_{1:\infty}$), then $M(x_t|x_{<t})$ converges to 1, thus $M()$ correctly predicts the next data point x_t for large t. Furthermore, Solomonoff proved that $M()$ will converge to μ, i.e. $M(x_t|x_{<t}) \stackrel{t \longrightarrow \infty}{\longrightarrow} \mu(x_t|x_{<t})$. In the following we can use a slight modification of Solomonoff's universal prior as an approximation of the unknown μ.

3 Universal Agent AIXI

In this section we present the main result of this work, the universal agent AIXI. The difference between AIXI and the AIμ agent (see Definition 2) will be that AIXI is independent of the environment μ and thus can be deployed in any unknown environment. It can also be shown that AIXI is optimal in the sense that no other program can learn or solve any task faster (in less cycles) [2]. A drawback of this model is that it is incomputable. Thus we will illustrate a few approximations of AIXI.

3.1 The AIXI Model

Chapter 2.2 presented an approximation M for any unknown environment μ for inductive inference systems. We are looking for a general agent model which can solve multiple problem classes. Since there are problem classes which cannot be represented as sequence prediction problems (e.g. optimization problems, strategic games) a few modifications to the universal prior have to be made. First, M will be generalized to ξ s.th. we can apply ξ to many problem classes. Afterwards, we can use ξ as an approximation for our environment, thus plugging it into the AIμ model from Definition 2 will result in the general AIξ model.
We define $\xi(y_1x_1...y_kx_k)$ as the weighted sum over all (computable) environments q whose output is equal to the data points $x_{1:k}$ given $y_{1:k}$ as inputs:

$$\xi(yx_{1:k}) = \sum_{q:U(q,y_{1:k})=x_{1:k}} 2^{-l(q)}$$

One can show that ξ is still a valid probability distribution and also converges to μ, i.e. $\xi(yx_t|yx_{<t}) \overset{t\to\infty}{\longrightarrow} \mu(yx_t|yx_{<t})$.
As the next step we recall the optimal action of the AIμ agent for cycle k:

$$y_k^{\boldsymbol{\mu}} := \arg\max_{y_k} \sum_{x_k} \max_{y_{k+1}} \sum_{x_{k+1}} ... \max_{y_m} \sum_{x_m} (r_k + ... + r_m) \cdot \boldsymbol{\mu}(yx_{k:m}|yx_{<k})$$

By replacing μ by our approximation ξ we obtain the AIξ model in one line:

$$y_k^{\boldsymbol{\xi}} := \arg\max_{y_k} \sum_{x_k} \max_{y_{k+1}} \sum_{x_{k+1}} ... \max_{y_m} \sum_{x_m} (r_k + ... + r_m) \cdot \boldsymbol{\xi}(yx_{k:m}|yx_{<k})$$

$$= \arg\max_{y_k} \sum_{o_k r_k} \max_{y_{k+1}} \sum_{o_{k+1} r_{k+1}} ... \max_{y_m} \sum_{o_m r_m} (r_k + ... + r_m) \sum_{q:U(q,y_{1:m})=or_{1:m}} 2^{-l(q)}$$

Hence, the result is an expectimax description of the optimal output y_k for cycle k depending on the future outputs $y_{k+1}...y_m$, perceptions $o_k r_k ... o_m r_m$ and all possible environments q which are consistent with the history $yx_{1:m}$. This implies that the AIξ agent will automatically “learn” and predict the correct environment q after a sufficient amount of cycles k, thus the agent will be equivalently optimal as the AIμ agent for known environments.

The following properties of AIξ can be shown: AIξ is *universally optimal*, i.e. it is independent of μ and no other agent can learn faster (in fewer cycles). Also, this model is *complete*, since it is a full specification without missing any information/parameters. Furthermore, the agent acts *rational* due to maximizing his future long-term reward.
A few problem classes are already known to be solveable by AIXI, namely Sequence Prediction, Strategic Games, Function Optimization and Supervised Learning problems [3]. Also, the theory of AIXI allows to define a universal measure of intelligence, i.e. agents can be sorted w.r.t their intelligence using this measure [6].
One open question is the optimal choice of the *horizon* m, i.e. the number of cycles the agent has to "look ahead", since too small values may miss out a global optimum but too large values could result in bad adaptations to small local changes. Another open problem is the incomputability of the AIXI model: it is only limit-computable. In order to implement AIXI, approximations are necessary, which lead us to the next section.

3.2 Approximations of AIXI

The core component of the AIξ agent is the approximation ξ of the environment. This function is not computable, thus AIξ is also not computeable. This highlights the importance of approximation algorithms.
One first approach could be to asymptotically compute the next action. This does not scale well since the convergence of ξ is very slow [3]. Another approach is to limit the number of possible agents by setting a boundary to runtime t and length l of the TM description, thus resulting in the AIXItl model with runtime $O(t \cdot 2^l)$.
The idea is as follows: Given an agent p' which has a runtime t' in each cycle we first execute all programs p which have less or equal runtime and description size than p'. Afterwards, we pick the p with the output which maximizes the expected reward $V_\xi^p := \sum_{x_{1:m}} (r_1 + ... + r_m)\xi(yx_{1:m})$, where $y_{1:m} = p(x_{<m})$. This results in a total runtime of $O(2^{l(p')} \cdot t')$. This approach has another issue: V_ξ^p is not computable. But there is a solution by assuming that after each cycle every agent p writes a reward value w_k^p on the output tape and thus rate themselves. Furthermore, it can be shown that there is a logical predicate VA(p) which guarantees that $w_k^p \leq V_\xi^p$ and which the universal TM can easily verify [3].
Now we can turn to the actual algorithm:

Given: parameters l_P, l, t

1. Create all possible binary strings of length $\leq l_P$ and interpret them as a logical proof VA(p) for some program p.
2. Remove all p whose length exceeds l.
3. Modify each p s.th. for every cycle k it holds: If p exceeds t timesteps, then it writes $w_k = 0$ and an arbitrary output y_k on the output tape.
4. Set $k := 1$.
5. Simulate each p on $yx_{<k}$.
6. Select the program with highest (verified) rating: $p_k^* := \arg\max_p w_k^p$
7. Write $y_k^{p_k^*}$ on the output tape.
8. Receive a perception x_k from environment.
9. Set $k := k + 1$, repeat step 5.

One can show that for every agent p with $l(p) \leq l$, $t(p) \leq t$ and a proof $VA(p)$ of size $\leq l_P$ the AIXI*tl* agent always has a higher or equal reward: Assume there is a program p' of reasonable size and runtime which achieves higher reward. Then its output would have been taken as the maximum and in step 6 and thus AIXI*tl* would have at least achieved the same reward, contradiction.
The total size of the agent is bounded by $O(log(l \cdot t \cdot l_p))$, with the setup-time of $O(l_P^2 \cdot 2^{l_P})$ (generating all proofs and verifying them). Also, in each cycle k AIXI*tl* needs $O(2^l \cdot t)$ time to enumerate all programs p of size $\leq l$ and execute them, each with runtime t.
In the recent past a few other approximations have been developed for AIXI, namely a direct brute-force approximation for 2×2 repeated matrix games (e.g. the Prisoner's Dilemma), a Monte-Carlo approximation [8] and the MC-AIXI-CTW algorithm which is based on Bayesian data compression and upper confidence bound expectimax tree search [12]. It was shown that the latter was able to learn simple games like TicTacToe, Pacman, Kuhn Poker from *scratch*, i.e. only by showing instances of the game without teaching the rules. An illustration of the learning curve is shown in Figure 3.

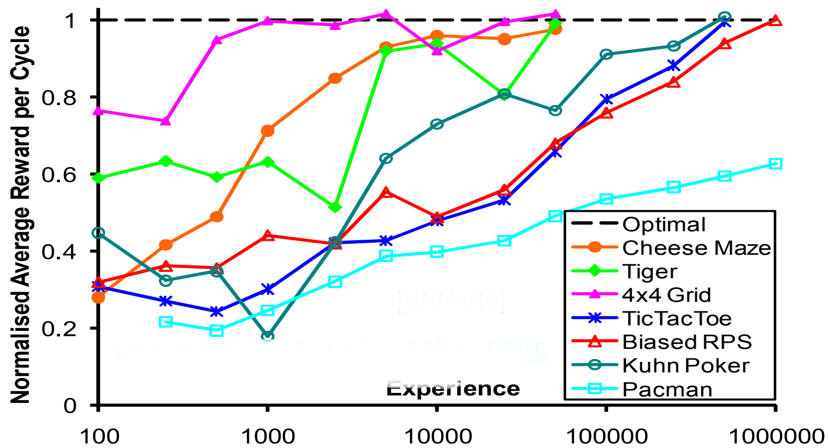

Figure 3: Illustration of the average reward as a function of the number of games played by the MC-AIXI-CTW algorithm

4 Conclusion

This seminar work presented an introduction to the theory of Universal AI, which is the first sound and complete mathematical model of general artificial intelligence. First, the foundations of UAI have been mentioned, namely Occam's Razor, the field of Algorithmic Information Theory (Kolmogorov complexity) and Epicurus' Principle of Indifference. Afterwards, a model for a super intelligent agent AIXI has been presented, which acts optimally in any unknown environment, given that it is computable. Since AIXI itself is not computable, we introduced several approximations, including the AIXI*tl* model. Recent implementations of these approximations have shown that AIXI is able to learn simple games from scratch, i.e. just by repeatedly playing the games without providing any rules.

Finally, there are still open issues: The question of finding better and more efficient approximations of AIXI and the generalization of these implementations to more problem classes than just simple games.

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[2] Marcus Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability.* Springer, Berlin, 2005. 300 pages, http://www.hutter1.net/ai/uaibook.htm.

[3] Marcus Hutter. Universal algorithmic intelligence: A mathematical top→down approach. In *Artificial General Intelligence*, pages 227–290. Springer, Berlin, 2007.

[4] Marcus Hutter. Open problems in universal induction & intelligence. *Algorithms*, 3(2):879–906, 2009.

[5] Marcus Hutter. One decade of universal artificial intelligence. In *Theoretical Foundations of Artificial General Intelligence*, pages 67–88. Atlantis Press, 2012.

[6] Shane Legg and Marcus Hutter. Universal intelligence: A definition of machine intelligence. *Minds & Machines*, 17(4):391–444, 2007.

[7] Ming Li and Paul M.B. Vitnyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer Publishing Company, Incorporated, 3 edition, 2008.

[8] Sergey Pankov. A computational approximation to the aixi model. In *Proceedings of the 2008 Conference on Artificial General Intelligence 2008: Proceedings of the First AGI Conference*, pages 256–267, Amsterdam, The Netherlands, The Netherlands, 2008. IOS Press.

[9] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

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

[11] R.J. Solomonoff. A formal theory of inductive inference: Part II. *Information and Control*, 7(2):224–254, June 1964.

[12] Joel Veness, Kee Siong Ng, Marcus Hutter, and David Silver. Reinforcement learning via AIXI approximation. In *Proc. 24th AAAI Conference on Artificial Intelligence*, pages 605–611, Atlanta, USA, 2010. AAAI Press.