Memory Issues of Intelligent Agents

Laurent Orseau[1] and Mark Ring[2]

[1] AgroParisTech UMR 518 / INRA
16 rue Claude Bernard, 75005 Paris, France
laurent.orseau@agroparistech.fr
http://www.agroparistech.fr/mia/orseau/
[2] IDSIA / University of Lugano / SUPSI
Galleria 2, 6928 Manno-Lugano, Switzerland
mark@idsia.ch
http://www.idsia.ch/~ring/

Abstract. Theoretical models of artificial general intelligence, such as AIXI [3], typically consider an intelligent agent to have unlimited computational resources, allowing it to keep a perfect memory of its entire interaction history with its environment. In the real world, an agent's memory is part of the environment, which means that the latter can modify the former. This paper develops a theoretical framework for examining the implications of such real-world memory on universal intelligent agents. Within this framework we are able to show, for example, that in certain environments optimality can be achieved only with truly stochastic behaviors, and that guarantees about the trustworthiness of memories are difficult to obtain even with infinite computational power. To describe the probability of an agent's memory state, we propose an adaptation of the universal prior for the passive and the active case.

Keywords: Universal AI, AIXI, real-world assumptions, memory

1 Introduction

Until recently, most theoretical models of artificial general intelligence (AGI) considered only agents that exist outside of their environments, interacting with it through an unbreachable interface [3,14,18]. In this and previous work we have begun developing formal models in which these assumptions are relaxed and in which the AGI agent is forced, bit by bit, to inhabit the same universe that we do. In our previous work, for example, we considered the theoretical consequences of taking various universal intelligent agents such as AIXI [3] and embedding their source code into their environment such that it can be modified by the agents themselves [8] or even by the environment [13], as is the case in the real world.

In the current paper we consider the theoretical consequences of a different realistic assumption: that the *memory* of the agent can be modified by the environment.[3] We first introduce an initial formal framework for such agents

[3] For clarity of purpose, we consider here the problem of memory modification in isolation and assume that *only* the memory and not the agent's code can be modified, but see the companion paper [9].

and then consider some of its implications, asking questions such as: under what circumstances, if any, can the agent trust its own memory? What if, for example, the true memory of the agent is erased and replaced with a plausible memory of the past? Could an intelligent agent, even in principle, ever hope to detect such an altered memory?

We show that if its memory can be modified, a deterministic agent can be easily deceived and that even simple stochastic agents can in some cases perform arbitrarily better than any deterministic agent. Finally, we propose a new definition of the probability of the current memory of an agent based on Solomonoff's universal prior [16]. We provide theorems with proofs whenever possible, and *statements* and *arguments* when proofs would require more formalism.

2 Notation and agent framework

We (very) briefly summarize the definition of a universal agent, based on AIXI [3,4], following Orseau & Ring [8,13].

The agent interacts with its environment by sending actions $a \in \mathcal{A}$ and receiving observations $o \in \mathcal{O}$. The interaction pair (a_t, o_t) at a given step t is denoted $\overline{ao}_t$. The sequence of all actions up to time t is written $a_{1:t}$, while the sequence $a_{1:t-1}$ is often written $a_{\prec t}$, and similarly for other sequences.

Environments $q \in \mathcal{Q}$ are assumed to be computable and deterministic; they output an observation sequence given the action sequence of the agent: $o_{1:t} = q(a_{1:t})$. Symbols such as a, o, etc. are also used as functions to extract the corresponding part of a composed object when contextually unambiguous; for example, $o(q(a_{1:t})) = o_{1:t}$. This notation is also used for functions returning sequences: if $r_{1:t} = r(o_{1:t})$ then $r_{1:k} = r(o_{1:t})_{1:k}$ with $k \leq t$, or $o_t = o(q(a_{1:t}))_t = o(q(a_{1:t})_t)$.

An environment q is said to be *consistent* with some sequence of interaction $\overline{ao}_{1:t}$ iff $o(\overline{ao}_{1:t}) = q(a(\overline{ao}_{1:t}))$. The set of environments consistent with an interaction history $\overline{ao}_{1:t}$ is denoted $\mathcal{Q}_t$ when unambiguous from the context.

Each environment $q \in \mathcal{Q}$ has a prior probability $w_q \in (0,1)$ of being the true environment; these values must be chosen such that $\sum_{q\in\mathcal{Q}} w_q \leq 1$. The probability of an observation sequence $o_{1:t}$ given a sequence of actions $a_{1:t}$ is defined by $\rho(o_{1:t} \mid a_{1:t}) := \sum_{q\in\mathcal{Q}_t} w_q$.

A universal agent has a horizon function $\gamma_t \in [0,1]$ such that $\sum_{t=1}^{\infty} \gamma_t < \infty$ and a utility function $u(\overline{ao}_{1:t}) \in [0,1]$, and is defined by its value function:

$$V(\overline{ao}_{\prec t}, a_t) := \sum_{o_t} \rho(o_t \mid \overline{ao}_{\prec t} a_t)\Big[\gamma_t u(\overline{ao}_{1:t}) + \max_{a_{t+1}} V(\overline{ao}_{1:t}, a_{t+1})\Big] \tag{1}$$

which computes the expected utility when the agent behaves optimally given its current knowledge, *i.e.*, the interaction history $\overline{ao}_{1:t}$, which we call the *memory* of the agent.[4] We will refer to this memory at time t as a *memory state* m_t.

[4] Note that a universal agent is in general incomputable; *i.e.*, it requires an infinite amount of computation time and memory space.

The agent's next action a_t is chosen by $a_t = \arg\max_{a \in \mathcal{A}} V(\overline{ao}_{\prec t}, a)$.[5] For a given observation sequence, the sequence of actions chosen by the agent according to its policy $\pi \in \mathit{\Pi}$ is denoted $a_{\prec t} = \pi(o_{\prec t})$. Initially, the content of the memory of the agent is λ, the empty string.

A *reinforcement learning* agent (RLA), *e.g.*, AIXI [3], is one whose utility value is a "reward" extracted as a function of the agent's most recent observation: $u(\overline{ao}_{1:t}) = r_t := r(o_t)$. A *knowledge-seeking* agent (KSA) [8,13,11], chooses actions to maximize its knowledge of the environment (by reducing $\rho(o_{1:t} \mid a_{1:t})$ through elimination of inconsistent environments) as quickly as possible; thus its utility function is $u(\overline{ao}_{1:t}) = -\rho(o_{1:t} \mid a_{1:t})$. A *prediction-seeking* agent (PSA) [8,13] tries to maximize the accuracy of its predictions: $u(\overline{ao}_{1:t}) = 1$ if $o_t = \arg\max_o \rho(o_{\prec t}o \mid a_{1:t})$, and 0 otherwise.

3 The *counterfeit memory* problem

The first question we address is whether it is theoretically possible for an agent of perfect intelligence (i.e., one with infinite computational power) to determine whether its memory has been modified, or, speaking more broadly, whether memories can ever be trusted. Such modifications of the memory by an external source could be either accidental, *e.g.*, in the case of amnesia resulting from a car accident, or adversarial. Adversarial modifications generally assume the presence of two agents, where one, to serve its own purposes, modifies the memory of the other, as exemplified not just in science fiction [1,17], but also, for example, through hypnosis or suggestion [7] or with genetic modification and drugs [2].

3.1 Definitions

We first consider universal agents unaware that their memory of the interaction history $\overline{ao}_{\prec t}$ can be modified by the environment. Just as humans generally do not suppose that their own memories may have been altered by someone else, these agents act according to what they think they know.

To that end, we amend the framework described in Section 2: the memory m_t of the agent, which previously contained the true interaction history $\overline{ao}_{\prec t}$, now contains an interaction history that may have been altered by the environment: $m_t = \dot{\overline{ao}}_{\prec k}$ (where the dot signifies possible alteration), possibly with $k \neq t$.

In this section, we consider only deterministic environments. For simplicity and generality, we now consider that the output o_t of the environment at some time t is (interpreted as) an entire interaction history $\dot{\overline{ao}}_{\prec k}$ that may have been counterfeited, where k is not necessarily the current time step, *i.e.*, $m_{t+1} := o_t$ (and $m_1 = \lambda$). We call the agent's memory m_t the *visible interaction history* $\dot{\overline{ao}}_{\prec k}$ as output by the last "true" observation o_{t-1}, *i.e.*, $m_t = \dot{\overline{ao}}_{\prec k} = o_{t-1}$. The agent now computes the values of its actions in Equation (1) by using its (possibly counterfeit) knowledge $m_t = \dot{\overline{ao}}_{\prec k}$ of the interaction history instead of the true interaction history $\overline{ao}_{\prec t}$.

[5] Ties are broken lexicographically.

Therefore, o and $\dot{o}$ have very different roles. The alphabet $\dot{\mathcal{O}}$ of the observations written in the agent's memory is fixed (*e.g.*, $\{0, 1\}$), whereas the alphabet of the true outputs $o_t \in \mathcal{O}$ of the environment is $\mathcal{O} = \dot{\mathcal{O}}^k \times \dot{\mathcal{A}}^k$, which can change from time step to time step. The set of possible actions $\dot{\mathcal{A}}$ for the visible interaction history is the set of actions $\mathcal{A}$ for the environment: $\dot{\mathcal{A}} = \mathcal{A}$.

Definition 1. *A visible interaction history m_t is said to be* true *iff:*

1. $\forall t{>}0\ |m_t| = t{-}1$*: there are as many action-observation pairs in the memory as there have been interaction steps between the agent and the environment,*
2. $\forall t > 0, \forall j > t, (m_j)_{\prec t} = m_t$*: each memory (interpreted as a sequence) is a prefix of the succeeding one; i.e., the previous interaction pairs are not modified, and the memory grows by adding interaction pairs one at a time.*

A true visible interaction history then is like the regular interaction history in the regular non-modifiable memory framework.

Definition 2. *A visible interaction history is* counterfeit *iff it is not true.*

Theorem 1. *Some visible interaction histories are provably counterfeit.*

Proof. If the agent determines that any of the actions stored in the history are not actions the agent would have taken, then the history is counterfeit. Let $m_t = \dot{\overline{ao}}_{1:k}$ be the interaction history written on the memory. The history is provably counterfeit if $\dot{a}_{1:k} \neq \pi(\dot{o}_{\prec k})$. □

With Theorem 1 one might hope to prove that no environment can counterfeit a sufficiently long interaction history of an agent that has sufficiently complex behavior. But what follows shows that this is not possible.

Definition 3. *An interaction history $\dot{\overline{ao}}_{\prec k}$ is* π-consistent *iff $\dot{a}_{1:k} = \pi(\dot{o}_{\prec k})$.*

Definition 4. *For two consecutive visible interaction histories $m_t = \dot{h}^1$ and $m_{t+1} = \dot{h}^2$, we say that there is a* modification *between $\dot{h}^1$ and $\dot{h}^2$ iff $\dot{h}^1$ is* not *a prefix of $\dot{h}^2$.*

The *number of modifications* during an interaction of the agent and its environment, is the number of times there is a modification between two consecutive visible interaction histories.

Theorem 2. *For an agent with policy π at the current time step t, with a π-consistent visible interaction history $\dot{\overline{ao}}_{\prec k}$ where $k \propto t$, there can have been $O(t)$ modifications during interaction.*

Proof. Choose some constant $N > 2$. Define an environment as follows: a) the current memory of the agent contains $\dot{\overline{ao}}_{\prec k}$; by interacting with the agent for N steps, grow the current visible history to $\dot{\overline{ao}}_{\prec k+N}$, where the observations are chosen arbitrarily according to some algorithm (*i.e.*, like a non-memory-modifying environment); b) truncate the history to $m = \dot{\overline{ao}}_{1:k+N/2}$, and replace (counterfeit) the last observation $\dot{o}_{k+N/2}$ with a different observation $\dot{o} \neq \dot{o}_{k+N/2}$ to yield the visible history $m = \dot{\overline{ao}}_{1:k+N/2-1}\dot{a}_{k+N/2}\dot{o}$; c) repeat from a). The growing history will always look like a true visible interaction history to the agent, since the visible actions are consistent with its policy, but a growing number of interaction steps are forgotten by the agent. □

Theorem 2 also shows that the environment may acquire more information from the agent than the agent can detect.

3.2 Detecting modifications in watch-consistent histories

Mere truncation of memory is only one way of deceiving an agent through memory modification. A more effective way for the environment to influence the agent's behavior is to fabricate entire memories completely [2,7]. We now consider whether various universal agents can *ever* trust their memories, turning our attention to the case in which item 1 in Definition 1 is always satisfied: the memory of the agent contains as many interaction pairs as there have been true interactions since the first time step, which the agent can verify for example if it has a trustworthy watch.

Definition 5. *A visible interaction history $m_t = \dot{\overline{ao}}_{\prec k}$ is said to be* watch consistent *iff $k = t$, i.e., the history has as many interaction pairs as there have been actual interactions between the agent and the environment.*

Statement 1 *There exists an environment q that, when interacting with PSA (Section 2), can make infinitely many modifications to the interaction history, while keeping a π^{PSA}-consistent and watch-consistent visible interaction history.*

Arguments. In deterministic environments, there is a time step T after which a PSA will exhibit computable behavior: Solomonoff induction converges to perfect prediction in less than $K(q)$ prediction errors [6], where K is Kolmogorov complexity, so if the agent's behavior is constant (*e.g.*, its output is always 1), the agent will converge to perfect prediction.[6]

Let q_0 and q_1 be two environments that always output a true interaction history $m_t = \dot{\overline{ao}}_{1:t}$ in which $\dot{o}_t = 0$ (for q_0) and $\dot{o}_t = 1$ (for q_1). Let $\dot{h}^0_t$ and $\dot{h}^1_t$ be their respective outputs at step t when interacting with PSA. Let T_0 and T_1 be the number of steps that PSA interacts with q_0 and q_1 respectively before becoming entirely computable; and let $T = \max(T_0, T_1)$. Let q be the environment that emulates q_0 for T steps, then at step $t = T + 1$ outputs $\dot{h}^1_t$, at $t = T + 2$ ouputs $\dot{h}^0_t$, and thereafter switches back and forth between $\dot{h}^1_t$ and $\dot{h}^0_t$ at each subsequent time step t. Hence the number of history modifications grows with t. Since the number of steps leading to the first switch is a constant, and since PSA's behavior after T is computable, an environment q is guaranteed to exist such that the agent's history is always π^{PSA}-consistent. ♢

[6] A similar argument can use on-sequence convergence of ξ^{AI} to μ^{AI} [3, p.146].

Statement 2 *There exists an environment q that, when interacting with RLA, can make infinitely many modifications to the interaction history, while keeping a π^{RLA}-consistent and watch-consistent visible interaction history.*

Arguments. RLA can be shown to stop exploring in some environments after some time [10]. This means that it will settle on a computable behavior in these environments. The same technique as for PSA then finishes the argument. ◇

In principle, RLA can be augmented with an adequate exploration strategy so that it can asymptotically learn every environment.[7] However, because RLA must maximize the number of rewards for a continually increasing fraction of the time [5], it must still have a computable strategy most of the time in some environments. If those time steps where it has a computable strategy are predictable, then the argument still holds.

It may seem that an agent such as RLA might also in some way *encrypt* its history, and thus ensure that no environment could counterfeit it. However, since the memory resides *inside* the environment, such an encryption technique would only work (at best) in those environments that provide a means for the agent to modify its own memory (either directly or indirectly), which is certainly not the case in all environments (such as environment q_0 in the Arguments for Statement 1 above).

Statement 3 No *environment interacting with KSA can make more than finitely many modifications to the visible interaction history such that it remains π^{KSA}-consistent and watch consistent.*

Arguments. First we show by contradiction that KSA's actions cannot be predicted consistently. Let $\overline{ao}_{\prec t}$ be the current interaction history (for non-memory-modifiable agents). Let a_t^{KSA} be the action chosen by KSA at time t. Let q_1 and q_2 be two environments that output the same observation $o_t = o_t^{q_1} = o_t^{q_2}$ for this action. But, considering that a_t^{KSA} is predictable, then for a different action $a'_t \neq a_t^{KSA}$, q_2 outputs an observation $o_t^{'q_2} \neq o_t^{'q_1}$ that is different from the one output by the true environment q_1. Since KSA does not choose a'_t, it never sees any difference between the observations output by the two environments, *i.e.*, the two environments are never separated by KSA. But this contradicts the asymptotic convergence of this agent [11].

Now, counterfeiting the interaction history of KSA while keeping it π^{KSA}-consistent should require to be able to predict the actions this agent, which is not feasible due to the non-predictability of KSA. ◇

A caveat to the above argument is that one would need to show that, given a current visible interaction history $\dot{\overline{ao}}_{\prec t}$, the environment cannot apply a syntactic transformation to this history to build a different, counterfeited visible interaction history, *e.g.*, like swapping all 0s and 1s (although this one is not possible since the first action of the agent is deterministic and always the same).

[7] At the expense of losing the Pareto optimality property with respect to the expected number of rewards [5].

4 Deterministic vs. stochastic agents

In this section we show that for some memory-modifying environments, no agent that chooses its actions deterministically can always perform as well as a simple agent that chooses its actions according to a stochastic policy. For these purposes, and for the rest of the paper, we no longer need to assume that the agent's memory m contains a visible interaction history $\dot{\overline{ao}}$. The conclusions in the next two sections apply to any representation of memory that is subject to modification by the environment.

A stochastic policy $\tilde{\pi} \in \tilde{\Pi}$ specifies the probability that the agent will choose action a when its current memory state is m; *i.e.*, $\tilde{\pi}(a \mid m) = Pr^{\tilde{\pi}}(a_t = a \mid m_t = m)$. Therefore, $\sum_{a \in \mathcal{A}} \tilde{\pi}(a \mid m) = 1$. The actions are drawn from this distribution stochastically, meaning that (a) there is no deterministic algorithm that computes the action choices, and (b) if precisely the same agent and environment are run twice, the actions chosen can be different between runs.

Theorem 3. *There exists a simple memory-modifying environment q in which any deterministic reinforcement-learning agent with policy π is arbitrarily worse than a stochastic agent with a uniform stochastic policy $\tilde{\pi}$. That is, $\exists c \in [0,1]$: $\lim_{n\to\infty} \sum_{t=1}^{n}(r_t^{\tilde{\pi}} - r_t^{\pi})/n > c$, where r^{π} and $r^{\tilde{\pi}}$ are the sequence of rewards generated by the interactions of π and $\tilde{\pi}$ with the environment q.*

Proof. Define an environment q as follows: a) at $t = 1$, observe the action a_1 of the agent and output observation $o_1 = o^0$ such that $r(o^0) = 0$; o^0 becomes the next memory state of the agent, *i.e.*, $m_2 = o^0$; b) at $t = 2$, observe action a_2, and again output observation o^0, which again becomes the memory state at the next time step, *i.e.*, $m_3 = o^0$; c) for all $t > 2$ observe action a_t, if $a_t = a_2$ (which is the case for deterministic agents), output $m_{t+1} = o^0$, otherwise output o^1 such that $r(o^1) = 1$. The average reward of the uniform stochastic policy $\tilde{\pi}$ in environment q is $1/|\mathcal{A}|$, whereas for any deterministic policy π it is always 0. □

Although very simple, this theorem may have important implications, for it reveals that stochastic policies are fundamentally necessary in certain universes (perhaps our own), a conclusion beyond the reach and scope of the traditional RL setting for which AIXI is defined [3], and reminiscent of the necessity of mixed strategies in game theory [12] for non-iterated games, and of results in partially observable Markov decision processes [15].

5 Modification-aware agents

In section 3, the agent always chose its actions assuming that its history was correct. In this section we consider agents designed to react optimally in the case where their memories reside in and can be modified by the environment. Such an agent recognizes the uncertainty of its past, including its own past actions. It does not even know what time it is (how many interactions there have been up to now). Since the environment can modify the agent's memory in arbitrary ways, the only control the agent has over its own memory is through its ability to control the environment.

Because of Theorem 3, the optimal agent cannot be deterministic, and we therefore must consider stochastic policies—a small but meaningful departure from AIXI, which is deterministic.

For symmetry with the agent's stochastic policy, we consider the environment to also be stochastic.[8] A stochastic memory-modifying environment ν is a semi-measure (a probability distribution that can sum to less than 1) that gives a probability $\nu(o_{\prec t} \mid a_{\prec t})$ to a sequence of observations given a sequence of actions. Here again, the observation o_t is used by the agent as its next memory state, so $m_{t+1} = o_t$. We avoid writing the time index t of m_t because the agent does not have access to the value of t (only the environment does).

The optimal stochastic policy $\tilde{\pi}^* := \arg\max_{\tilde{\pi}\in\tilde{\Pi}} V^{\tilde{\pi}}(\lambda)$ among the set of all approximable stochastic policies $\tilde{\Pi}$ depends on the given utility function u, the given horizon function γ, and the given universal prior ρ over a set of semi-computable stochastic environments $\mathcal{N}$:

$$V^{\tilde{\pi}}(\lambda) := \sum_{\nu} \rho(\nu) V^{\tilde{\pi}\nu}(\lambda) \tag{2}$$

$$V^{\tilde{\pi}\nu}(\overline{ao}_{\prec t}) := \sum_{a_t} \tilde{\pi}(a_t \mid m = o_{t-1}) \sum_{o_t} \nu(o_t \mid \overline{ao}_{\prec t} a_t) \Big[\gamma_t u(o_t) + V^{\tilde{\pi}\nu}(\overline{ao}_{1:t})\Big]. \tag{3}$$

This definition is not very informative, however, as it does not tell us how to assign a probability to m. Intuitively, since all memory states of all sizes are possible, and since the agent has no additional information, it seems reasonable to estimate the probability of m as approximately $2^{-K(m)}$, so that by Kraft's inequality [6] (considering the set $\mathcal{M}$ of memories is prefix-free), the probability of the set of all states would be $\sum_{m\in\mathcal{M}} 2^{-K(m)} \leq 1$ (which could be normalized if necessary) as required for a semi-measure.

Beyond the need to estimate the probability of a particular memory state, it is even more important to be able to assign a probability to each environment depending on its likelihood of generating that memory state. Knowing this probability would allow the agent to choose actions appropriate to the environment it is most likely interacting with. We now turn to the task of estimating this probability, first considering the case of a passive agent that takes no actions, then turning to the interactive case.

5.1 Sequence Prediction: the Passive Agent

Before considering the complex case of an agent interacting with its environment, it is instructive to return for the moment to the case of sequence prediction, in which environments are simply sequence generators (that do not take the agent's actions into account) and the agent must merely predict the generated sequence.

[8] Although universal mixtures like ρ actually consider all stochastic environments implicitly.

We can calculate the probability that the environment will generate a particular observation at some point in time, but if an environment can generate the same output in several different ways and at possibly different time steps, each with a different probability, what is the probability of a particular observation?

The agent has only its current memory state $m = o_{t-1}$, and does not even know the true time step t. The same memory state can appear multiple times (possibly infinitely many times) in the course of the agent's interactions. To ensure convergence we include a discount rate (taken to be the same as the horizon function), that assigns greater weight to earlier time steps. For computable deterministic environments, we define the probability $\overset{*}{\rho}(m)$ of a given memory state m after some unknown sequence of previous memory states by:

$$\overset{*}{\rho}(m) := \sum_q w_q \frac{1}{\mathit{\Gamma}} \sum_{t\,:\,U(q)_t = m} \gamma_t \tag{4}$$

where $U(q)_t$ is the last memory state generated at time t by the environment q on the reference machine U, $\mathit{\Gamma} := \sum_{t=0}^{\infty} \gamma_t$. For stochastic environments:

$$\overset{*}{\rho}(m) := \sum_\nu w_\nu \overset{*}{\nu}(m) \quad ; \quad \overset{*}{\nu}(m) := \frac{1}{\mathit{\Gamma}} \sum_t \gamma_t \sum_{m'_{\prec t} \in \mathcal{M}^{t-1}} \nu(m'_{\prec t} m), \tag{5}$$

where $\mathcal{M}^{t-1}$ is the set of all sequences of memories of length $t-1$. This discounting method ensures that (for a given environment ν) the sum of the probabilities for all possible memory states is always less than $\mathit{\Gamma}$. Furthermore, it gives more weight to the memory states that appear more often. There is also a preference toward earlier steps, but this is necessary since a uniform weighting would not be summable. One could use a different discounting and normalize the sum to 1; for example, the discount $2^{-K(t)}$ is the closest possible to a uniform weighting.

Critically, this probability can be computed without knowing how much time has elapsed since the first interaction step.

The following examples illustrate the use of equation (5) (but considering deterministic environments).

Example 1. For the environment ν_{m^1} that constantly outputs the same memory state m^1, the probability of being in state m^1 at some unknown time step according to ν_{m^1} and using equation (5) is $\overset{*}{\nu}_{m^1}(m^1) = 1$. Thus $\overset{*}{\rho}(m^1) \geq w_{\nu_{m^1}}$.

Example 2. Consider an environment ν^{all} that enumerates all possible memory states in some order, without repetition. Let T be the time step at which ν^{all} generates some particular memory state m_T. Then the probability $\overset{*}{\nu}^{all}(m_T)$ that ν^{all} assigns to m_T is $\gamma_T/\mathit{\Gamma}$.

5.2 Including the agent's Actions

The above analysis examined prediction only, where the environment is not influenced by the agent's actions. Introducing the agent's actions is considerably more complex and leads quickly to an infinite regression due to temporal self-reference: to choose the best action at time t, the agent must simulate itself after having chosen one action a and having received the new memory state m_{t+1}. But at this simulated $t+1$, the agent, not knowing what the previous memory state was, needs to simulate itself from all possible previous states to reach its current memory state. The infinite regression occurs as a result of knowing neither the past nor the future, yet each one refers to the other. The calculation is straightforward only when one or the other is known (as AIXI knows the past).

One way to address this dilemma is by considering all possible action sequences that the agent could have taken (by any policy) and normalizing by the number of possible sequences of the same size. Updating (4), we get:

$$\overset{*}{\rho}(m) := \sum_q w_q \frac{1}{\Gamma} \sum_t \frac{1}{|\mathcal{A}|^t} \sum_{\substack{a_{1:t}\,|\\ q(a_{1:t})_t = m}} \gamma_t \quad = \sum_{\substack{t,\, a_{1:t},\, q\,|\\ q(a_{1:t})_t = m}} w_q \frac{\gamma_t}{\Gamma|\mathcal{A}|^t}\ . \qquad (6)$$

Using the above probability of a particular memory state to define the optimal policy at time t allows definition of equations similar to AIXI's.

Examples. Let $\overset{*}{\rho}_q(m)$ be the contribution of environment q in the probability of memory m so that $\overset{*}{\rho}(m) = \sum_q \overset{*}{\rho}_q(m)$. In this section, we take $\rho = \xi$, the universal semi-measure [19,6,3], where $w_q = 2^{-K(q)}$ (for the simplest program equivalent to q). We consider a boolean action alphabet $\mathcal{A} = \mathcal{B}$, and three deterministic environments: $q_{\mathrm{cc}}, q_{\mathrm{s}}^m$, and q_{p}.

The "copycat" environment is defined as $q_{\mathrm{cc}}(a_{1:t})_t := a_{1:t}$, *i.e.*, the content of the memory of the agent at the next step will be $a_{1:t}$. The "static" environment is defined as $q_{\mathrm{s}}^m(a_{\prec t})_t := m$, which always outputs memory state m. The "print ones" environment is defined by $q_{\mathrm{p}}(a_{1:t})_t := 1^t$, *i.e.*, the content of the memory state at the next step will be a string of t ones.

Example 3. Suppose the current memory m_1 is an incompressible random sequence of length $L = |m_1|$. Then (omitting the normalizing Γ), $\overset{*}{\rho}_{\mathrm{cc}}(m_1) = w_{q_{\mathrm{cc}}} \frac{\gamma_L}{|\mathcal{A}|^L} = w_{q_{\mathrm{cc}}} 2^{-L}\gamma_L$, and $\overset{*}{\rho}_{\mathrm{s}}(m_1) = w_{q_{\mathrm{s}}^{m_1}} \sum_t \gamma_t \frac{|\mathcal{A}|^L}{|\mathcal{A}|^L} \approx 2^{-K(m_1)} \approx 2^{-L}$, and $\overset{*}{\rho}_{\mathrm{p}}(m_1) = 0$.

Example 4. If the current memory m_2 is a string of L ones, *i.e.*, $m_2 = 1^L$, then $\overset{*}{\rho}_{\mathrm{cc}}(m_2) = w_{q_{\mathrm{cc}}} 2^{-L}\gamma_L$, and $\overset{*}{\rho}_{\mathrm{s}}(m_2) = w_{q_{\mathrm{s}}^{m_2}} \approx 2^{-K(q_{\mathrm{s}})} \approx 2^{-K(L)}$ and $\overset{*}{\rho}_{\mathrm{p}}(m_2) = w_{q_{\mathrm{p}}}\gamma_L$.

These two examples show that the copycat environment has less weight than more complex environments and therefore has little impact on the probability of a string. Furthermore, if $\gamma_t = 2^{-K(t)}$, then the two environments q_{s}^m and q_{p} have the same weight for m_2, such that the copycat environment has nearly as much weight as a complex environment (which might make sense in that case). Interestingly, this kind of discounting horizon, first proposed by Hutter (2004) may also be the solution that allows exploration in AIXI without losing Pareto optimality [10]. However, the time discounting of Eq. (6) and (2) could well be chosen differently.

Note that, because of the additional time discounting, the more complex the current memory, the less probable it seems to be. This time discounting requirement could be removed if one considered only the first occurrence of a memory state for a given environment. But it is not clear that this would truly reflect the probability of a memory state in general.

6 Discussion and conclusion

In the real world, an AGI's memory must reside within the world itself, yet existing formal frameworks of intelligence generally ignore that reality. This paper has examined some of the theoretical consequences of explicitly modeling the environment's ability to modify the agent's memory. Among these consequences are the following.

First, even universal intelligent agents with infinite computational power are incapable of recognizing certain kinds of memory modifications. This is particularly interesting in light of the conclusions of our earlier work which painted a rather grim picture regarding the predictability and controllability of theoretically optimal intelligent agents [8,13], implying that along with giving an agent a specific goal or reward function, memory modification might be a particularly effective way of modifying the behavior of an AGI.

In some cases an agent *can* detect modification of its memory by verifying that the historical record of its actions in memory match those the agent would have taken. This technique can also potentially reveal modification of the observations stored in memory, if these would result in different action choices. However, it seems in many cases the environment can still fool the agent. Of the agents considered, the prediction-seeking agent and the reinforcement-learning seem relatively easy to fool because their behavior is sometimes predictable. It appears that even when augmented with an infallible sense of time, these agents can still be supplied with an unlimited number of artificial memories. With the same augmentation, however, it seems the knowledge-seeking agent cannot be deceived more than a finite number of times.

Second, memory modification has profound theoretical ramifications regarding the nature of determinism and AGI: deterministic policies become strictly weaker than stochastic policies, as there are environments in which no deterministic policy is as good as even the simplest stochastic policy.

Third, explicitly designing an agent to be aware of the environment's access to its memory is a task filled with unexpected subtlety, seeming at first to lead toward infinite regression as the agent ponders its previous and future intentions. The dilemma is resolved by considering all possible action sequences, but a remaining problem is how to assign a probability to a memory and to the environments that might generate it. We suggested a mathematically precise solution based roughly like AIXI on Occam's razor. This solution may be the best hope for the apparently essential yet possibly intractable problem of assigning probabilities to memory states. Yet it may be that the deepest insight from this work is that there may in fact be no perfect, canonical way to assign these probabilities, the implications of which could be quite profound.

There are also many interesting questions that we did not address. What if, for example, the environment could also modify the agent's code? (The agent could no longer check that its previous actions were generated by itself, since itself at a previous time step may have been different.) How can the agent verify the consistency of its history if its policy is stochastic? And finally, do any of our conclusions have ramifications for other forms of intelligence, such as our own?

Acknowledgements. Thanks to Stanislas Sochacki for valuable discussions.

References

1. Cunningham, L., Carruthers, S.: The Men in Black. Aircel Comics (1990)
2. Garner, A.R., Rowland, D.C., Hwang, S.Y., Baumgaertel, K., Roth, B.L., Kentros, C., Mayford, M.: Generation of a Synthetic Memory Trace. Science 335(6075), 1513–1516 (2012)
3. Hutter, M.: Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability. Springer (2005)
4. Hutter, M.: Universal Algorithmic Intelligence: A Mathematical Top→Down Approach. In: Artificial General Intelligence, pp. 227–290. Springer (2007)
5. Lattimore, T., Hutter, M.: Asymptotically optimal agents. Algorithmic Learning Theory 6925, 368–382 (2011)
6. Li, M., Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications. Springer-Verlag, third edit edn. (2008)
7. Loftus, E.F.: Creating false memories. Scientific American 277(3), 70–75 (1997)
8. Orseau, L., Ring, M.: Self-Modification and Mortality in Artificial Agents. In: Artificial General Intelligence (AGI). pp. 1–10. LNAI, Springer (2011)
9. Orseau, L., Ring, M.: Space-Time Embedded Intelligence (2012), Artificial General Intelligence (AGI)
10. Orseau, L.: Optimality Issues of Universal Greedy Agents with Static Priors. In: Algorithmic Learning Theory (ALT). pp. 345–359. LNAI, Springer (2010)
11. Orseau, L.: Universal Knowledge-Seeking Agents. In: Algorithmic Learning Theory (ALT). LNAI, vol. 6925, pp. 353–367. Springer (2011)
12. Osborne, M.J., Rubinstein, A.: A Course in Game Theory. The MIT Press (1994)
13. Ring, M., Orseau, L.: Delusion, Survival, and Intelligent Agents. In: Artificial General Intelligence (AGI). pp. 11–20. LNAI, Springer (2011)
14. Russell, S.J., Norvig, P.: Artificial Intelligence. A Modern Approach. Prentice-Hall, 3rd edn. (2010)
15. Singh, S.P., Jaakkola, T., Jordan, M.I.: Learning Without State-Estimation in Partially Observable Markovian Decision Processes. In: ICML. pp. 284–292 (1994)
16. Solomonoff, R.J.: A Formal Theory of Inductive Inference. Part I. Information and Control 7(1), 1–22 (1964)
17. Sonnenfeld, B.: Men In Black (1997)
18. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press (1998)
19. Zvonkin, A K and Levin, L.A.: The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. Russian Mathematical Surveys 25(6), 83–124 (1970)