On Reward Binarisation and Bayesian Agents

Elliot Catt ecatt@deepmind.com
Marcus Hutter uai@deepmind.com
Joel Veness aixi@deepmind.com
DeepMind, London, UK

Abstract

Reward binarisation is a common heuristically applied technique which can potentially simplify a given reinforcement learning problem. However this procedure done without care can modify the original problem, or throw away essential information. In this paper we study a number of natural forms of reward binarisation, and characterise their effects in terms of problem expressivity. We show positive results for MDPs, POMDPs, and k-order MDPs and a negative result for general history based reinforcement learning agents. Furthermore we show that binary Bayesian reinforcement learning agents enjoy convergence properties similar to their non-binarised counterparts.

Keywords: reward, binarisation, MDP, POMDP, Bayes, reinforcement learning, theory, agents, history, reduction

1. Introduction

Various kinds of reward shaping are common practice for many reinforcement learning practitioners. This paper considers some theoretical aspects of an extreme class of shaping methods, known as reward binarisation, whereby each reward received by the agent is mapped via some particular mechanism to either a 0 or a 1. Indeed, the origins of reinforcement learning date back to presence (one) or lack of (zero) reward with operant conditioning [Pav27, BB61]. Perhaps surprisingly a lot can be said in this case, especially in terms of representation power and ability of an agent to learn provided an appropriate binarisation scheme is chosen.

There are several reasons we may be interested in reward binarisation in practice. Specifically, some form of reward quantization features heavily in the general family of planning-as-inference techniques, whereby a reduction in the size of the reward/return space considerably eases the task of learning return-conditional models, such as in generative policy evaluation [VBH+15]. Another important current application is in upside-down RL, when applied to sequence model based RL (e.g. see [CLR+21, AASS22, JLL21]); here, restricting the reward space provides the benefit of significantly reducing the total number of tokens, which in turn simplifies the task of context-dependent probabilistic sequence modelling using modern methods such as Transformers.

While there are many possible binarisation schemes, there are a couple of natural candidates. In this work we consider two approaches, one which binarises the reward to a single bit, and another scheme which iteratively presents the reward to the agent one bit at a time over multiple time steps. We show that in the case of Markov Decision Processes (MDPs), binary rewards are a sufficient reward signal to exactly capture the value function; this extends to k-Markov Decision Processes and Partially Observable Markov Decision Processes (POMDPs) as well. We also provide a negative result, which shows that in a more general history based setup, binarisation cannot hope to preserve the original value function. Table 1 shows the resultant reward binarisations we consider.

Binarisation Type	Equivalent of aor	Notes
Single bit	$ao0 \vee ao1$	Possibly lossy
Multi-bit	$aor_1a'or_2 \ldots a'or_d$	a' is any action

Table 1: Reward Binarisation types

The second part of our paper deals with Bayesian learning in the presence of reward binarisation. Here we provide a number of technical results that justify the use of Bayesian learning in such situations. More precisely, we show that

a Bayesian mixture of the set of binarised environments dominates all semimeasures in the original class, as does the binarised form of the original Bayesian mixture. Furthermore, in the appendix, we show that Bayesian learning on top of a recently proposed action binarisation scheme [MH21] is also justified theoretically.

2. Background

In this work we use a general form of reinforcement learning notation which will allow our results to cover multiple reinforcement learning setups with a unified notation.

Setup. We will use the general reinforcement learning framework of [Hut05, Lei16, MH21]. Let $\mathcal{A}$ be the finite action set, $\mathcal{O}$ be the finite observation set, $\{0,1\} \subset \mathcal{R} \subset [0,1]$ be the finite (non-binary) reward set and $\mathcal{H} := (\mathcal{A} \times \mathcal{O} \times \mathcal{R})^* = \cup_{t=0}^{\infty} (\mathcal{A} \times \mathcal{O} \times \mathcal{R})^t$ be the set of finite length interaction histories (empty history in the $t = 0$ case). In this setup agents, denoted by $\pi : \mathcal{H} \to \Delta \mathcal{A}$, take actions $a \in \mathcal{A}$ and receive observations $o \in \mathcal{O}$ and rewards $r \in \mathcal{R}$ from the environment, denoted by $\mu : \mathcal{H} \times \mathcal{A} \to \Delta(\mathcal{O} \times \mathcal{R})$. The history of interactions up to time t, $a_1 o_1 r_1 \dots a_{t-1} o_{t-1} r_{t-1}$ is denoted by $h_{<t}$, and $h_{1:t} := h_{<t} a_t o_t r_t$. We denote the empty string by ϵ.

Value Functions. The goal of the agent in this setup is to maximise the expected future reward from the environment; this expected future reward is called the value function

$$V_\mu^{\pi,m}(h_{<t}) := \mathbb{E}_\mu^\pi \left[\sum_{i=t}^{m} \gamma(i) r_i | h_{<t} \right],$$

where γ is the (often geometric) discount function and m is the maximum horizon. The optimal value is defined as $V_\nu^{*,m}(h_{<t}) := \sup_\pi V_\nu^{\pi,m}(h_{<t})$, the optimal policy with respect to the value is defined as $\pi_\nu^{*,m}(h_{<t}) \in \arg\max_\pi V_\nu^{\pi,m}(h_{<t})$. In the case when $m = \infty$ we omit the m, that is, $V_\nu^\pi(h_{<t}) := V_\nu^{\pi,\infty}(h_{<t})$. We will also drop the history argument when it is an empty history, $V_\nu^\pi := V_\nu^\pi(\epsilon)$.

Unknown Environments. If the agent does not initially know which environment it is in, a model-based approach would be to consider a class of possible environments. Let $\mathcal{M}$ denote the class of possible environments; in this paper we allow for both finite $\mathcal{M}$ and countable $\mathcal{M}$, with special attention being given to the case when $\mathcal{M}$ is the set of enumerable semimeasures. An environment ν is a semimeasure if $\sum_{o_t r_t} \nu(o_t r_t | h_{<t} a_t) \leq 1$ for all $h_{<t}, a_t$ and a (proper) measure if equality holds instead.

Bayesian Reinforcement Learning. The Bayesian-optimal agent AIXI considers a Bayesian mixture of the class of possible environments and acts optimally with respect to the Bayesian mixture [Hut05, Hut06, Lei16]. We define the Bayesian mixture over the class of environments as follows:

$$\xi_\mathcal{M}(h) := \sum_{\nu \in \mathcal{M}} w(\nu) \nu(h)$$

where $w(\nu) > 0$ is the prior probability of the environment ν.

One of the key properties of the Bayesian mixture $\xi_\mathcal{M}$ is its dominance [RH11]. This means that for all $\nu \in \mathcal{M}$ and for all x we have that $\xi_\mathcal{M}(x) \geq c\nu(x)$. This property is important as it can be used to show that if μ is the true environment then ξ learns to correctly predict μ, specifically, $\xi_\mathcal{M} \to \mu$ with μ-probability 1 [Hut05].

MDPs/POMDPs. We will additionally consider the simplified Markov Decision Process setting of reinforcement learning wherein the agent and environment depend only on the most recent observation instead of the whole history. Lastly we will will consider the partially observable markov decision processes (POMDP) [KLM96, KLC98] class of environments. A POMDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mu_T, R, \mathcal{O}, \mu_O, \gamma \rangle$ where $\mathcal{S}$ is a finite set of hidden states, $\mathcal{A}$ is a set of actions, μ_T is the transition probability of the hidden states, R is the reward function, $\mathcal{O}$ is the set of possible observations, μ_O is the observation probability given a state, and γ is the discount factor. POMDPs can be represented in general reinforcement learning with the following definition from [Maj21]. An environment μ is a POMDP if there exists a finite state space $\mathcal{S}$, emission process μ_O, underlying MDP μ_T, and true occupancy probability χ such that

$$\mu(o'r'|ha) = \sum_{s' \in \mathcal{S}} \mu_O(o'r'|s') \sum_{s \in \mathcal{S}} \mu_T(s'|sa) \chi(s|h).$$

3. Binary Reward (Single-bit) Formulation

When considering the binarisation of the reward we will separately consider a specific case of when the rewards are binarised to a single bit, as opposed to the reward being binarised into several bits. Unless the original reward space $\mathcal{R}$ has a cardinality of 2, this will be a lossy binarisation in the sense that information can be lost during this binarisation. It turns out however, that despite this being a lossy binarisation, we are still able to show that in some cases it is sufficient.

In comparison to multi-bit binarisation, if we use single-bit binarisation we will not need to artificially add timesteps, this can be very important for methods which depend on the horizon, which becomes modified in the action binarisation and multi-bit reward binarisation cases.

Let $\overline{\mathcal{M}} := \{\mu \in \mathcal{M} | \mu \text{ only outputs reward 0 and 1}\}$ be the class of environments which output only binary rewards. Since we are considering the class of enumerable semimeasures this just means that the probability that the environment outputs a reward which isn't in $\{0, 1\}$ is 0.

In lieu of a specific binarisation scheme, we will use $\langle r \rangle$ to represent a binarisation of the reward to a single bit and $\langle h_{<t} \rangle = a_1 o_1 \langle r_1 \rangle \dots a_{t-1} o_{t-1} \langle r_{t-1} \rangle$ to represent a binarisation of the reward parts of the history. This is an intended feature of this setup as the results hold for any choice of reward binarisation.

Similarly to action binarisation [MH21], we will define a reward binarisation of an environment μ to one which uses only a binary reward. This setup is shown with Figure 1.

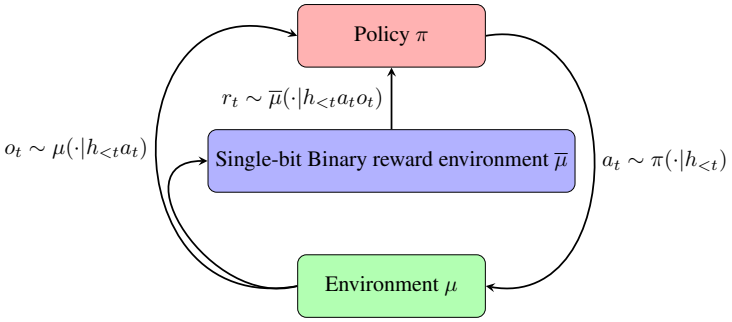

Figure 1: Agent-environment loop with single-bit reward.

Definition 1 (General Single-bit reward binarisation) *Given μ, we define the environment which outputs only binary reward $\overline{\mu} : \mathcal{H} \times \mathcal{A} \to \Delta(\mathcal{O} \times \{0, 1\})$ in two parts:*

$$\overline{\mu}(o_t 1 | h_{<t} a_t) := \sum_{r_t \in \mathcal{R}} \mu(o_t r_t | h_{<t} a_t) r_t$$

$$\overline{\mu}(o_t 0 | h_{<t} a_t) := \sum_{r_t \in \mathcal{R}} \mu(o_t r_t | h_{<t} a_t)(1 - r_t)$$

and $\overline{\mu}(o_t r | h_{<t} a_t) = 0$ if $r \notin \{0, 1\}$.

The reason we do $\sum_{r_t \in \mathcal{R}} \mu(o_t r_t | h_{<t} a_t)(1 - r_t)$ instead of $1 - \overline{\mu}(o_t 1 | h_{<t} a_t)$ is that there may be a semimeasure gap in ρ; that is, $\left(\sum_{r_t \in \mathcal{R}} \mu(o_t r_t | h_{<t} a_t)\right)$ may not equal 1.

We will now show that the transformation in Definition 1 is a linear transformation over environments. This will be important for later results.

Lemma 2 *Given an enumerable semimeasure ρ, if there exists $x, y \in \mathbb{R}$ and enumerable semimeasures μ_1, μ_2 such that $\rho = x\mu_1 + y\mu_2$, then $\overline{\rho} = x\overline{\mu}_1 + y\overline{\mu}_2$. That is, the operator $\bar{\cdot}$ is linear.*

Proof For reward 1,

$$\begin{aligned}
\overline{\rho}(o1|ha) &= \sum_{r\in\mathcal{R}} \rho(or|ha)\cdot r \\
&= \sum_{r\in\mathcal{R}} (x\mu_1(or|ha) + y\mu_2(or|ha))\cdot r \\
&= \sum_{r\in\mathcal{R}} (x\mu_1(or|ha)r + y\mu_2(or|ha)r) \\
&= x\sum_{r\in\mathcal{R}} \mu_1(or|ha)r + y\sum_{r\in\mathcal{R}} \mu_2(or|ha)r \\
&= x\overline{\mu}_1(o1|ha) + y\overline{\mu}_2(o1|ha).
\end{aligned}$$

For reward 0,

$$\begin{aligned}
\overline{\rho}(o0|ha) &= \sum_{r\in\mathcal{R}} \rho(or|ha)\cdot (1-r) \\
&= \sum_{r\in\mathcal{R}} (x\mu_1(or|ha) + y\mu_2(or|ha))\cdot (1-r) \\
&= \sum_{r\in\mathcal{R}} (x\mu_1(or|ha)(1-r) + y\mu_2(or|ha)(1-r)) \\
&= x\sum_{r\in\mathcal{R}} \mu_1(or|ha)(1-r) + y\sum_{r\in\mathcal{R}} \mu_2(or|ha)(1-r) \\
&= x\overline{\mu}_1(o0|ha) + y\overline{\mu}_2(o0|ha).
\end{aligned}$$

Therefore we have $\overline{\rho}(or|ha) = x\overline{\mu}_1(or|ha) + y\overline{\mu}_2(or|ha)$. ■

4. Value Function Results

In this section we will present several results regarding the reward binarisation and the corresponding value functions.

We start by presenting a lemma which will be useful for a later proof. Effectively this lemma says that given a set of history-real number pairs, there exists an environment such that the value function in that environment on each of those histories is its corresponding real number.

Formally, let $\mathcal{Q} \subset \mathcal{H} \times (0,1]$ be such that for all $h \in \mathcal{H}$ there exists a unique $q \in (0,1]$ such that $(h,q) \in \mathcal{Q}$.

Lemma 3 *For a given $\mathcal{Q}$, there exists a binary reward environment μ' such that for all $(h,q) \in \mathcal{Q}$ and for all policies π we have $V_{\mu'}^{\pi,m}(h) = q$.*

4.1 Impossibility of exact representation in general setting

We will now show that for any choice of reward-binarising function $\langle\cdot\rangle$, the value function cannot be exactly represented with a binary reward in *arbitrary* environments.

Theorem 4 *Given a reward-binarising function $\langle\cdot\rangle : \mathcal{R} \to \{0,1\}$, there does not exist an environment reward binarising function $\bar{\cdot} : \mathcal{M} \to \overline{\mathcal{M}}$ such that for all $\mu \in \mathcal{M}$, $h_{<t} \in \mathcal{H}$ and π we have*

$$V_{\overline{\mu}}^{\pi}(\langle h_{<t}\rangle) = V_{\mu}^{\pi}(h_{<t}).$$

Proof Let $\mathcal{R} = \{0, \frac{1}{2}, 1\}$ and $\mathcal{O} = \{o\}$. Let μ be such that for $t > 1$

$$\text{If } r_1 = 0 \text{ then } \mu(o0|h_{<t}a_t) = 0,\ \mu(o\frac{1}{2}|h_{<t}a_t) = 0,\ \mu(o1|h_{<t}a_t) = 1.$$
$$\text{If } r_1 = \frac{1}{2} \text{ then } \mu(o0|h_{<t}a_t) = \frac{1}{3},\ \mu(o\frac{1}{2}|h_{<t}a_t) = 0,\ \mu(o1|h_{<t}a_t) = \frac{2}{3}.$$
$$\text{If } r_1 = 1 \text{ then } \mu(o0|h_{<t}a_t) = \frac{2}{3},\ \mu(o\frac{1}{2}|h_{<t}a_t) = 0,\ \mu(o1|h_{<t}a_t) = \frac{1}{3}.$$

This means that

$$V^\pi_\mu(ao0h) = 1 + \gamma + \gamma^2 \ldots = \frac{1}{1-\gamma}$$
$$V^\pi_\mu(ao\frac{1}{2}h) = \frac{2}{3}(1 + \gamma + \gamma^2 \ldots) = \frac{2}{3(1-\gamma)}$$
$$V^\pi_\mu(ao1h) = \frac{1}{3}(1 + \gamma + \gamma^2 \ldots) = \frac{1}{3(1-\gamma)}$$

Since $\langle\cdot\rangle$ binarises the reward component of the history, (at least) two of $ao0h, ao\frac{1}{2}h, ao1h$ will map to the same binarised history, then by a pigeon hole argument we will have two of the above value functions needing to be equal, however that cannot occur, so the equality $V^\pi_{\overline{\mu}}(\langle h_{<t}\rangle) = V^\pi_\mu(h_{<t})$ cannot be satisfied. ■

While this result holds for any choice of binarisation of the environment (not limited to just Definition 1), it is a result about exact representation of the value function. It may still be possible that with binary rewards we can get within ε of the value function, like Theorem 23.

4.2 Possibility of exact representation

We have just shown that it is impossible to achieve exact representation of the value function for a general environment. We will now move on to showing that for a large interesting class of environments, it is possible to achieve exact representation of the value function. We will start by defining this class of environment, and then showing that it contains several environment classes of note, and lastly proving that any environment in this class can be represented by a corresponding binary environment.

Definition 5 *Let $\mathcal{M}^*$ be the set of environments for which the transitions do not depend on the reward components of the history. Formally:*

$$\mathcal{M}^* := \{\mu \in \mathcal{M} \mid \forall h_{<t}, a_t, o_t, r_t.\ \mu(o_t r_t|h_{<t}a_t) = \mu(o_t r_t|a_1 o_1 \ldots a_{t-1}o_{t-1}a_t)\}.$$

We can now define the environment reward binarisation we will be using to convert environment in $\mathcal{M}^*$ to binary reward environments.

Definition 6 (Single-bit reward binarisation for $\mathcal{M}^*$) *Given $\mu \in \mathcal{M}^*$, we define the environment which outputs only binary reward $\overline{\mu} : \mathcal{H} \times \mathcal{A} \to \Delta(\mathcal{O} \times \{0,1\})$ in two parts:*

$$\overline{\mu}(o_t 1|h_{<t}a_t) := \left(\sum_{r_t \in \mathcal{R}} \mu(o_t r_t|h_{<t}a_t)\left(r_t + \gamma V^{\pi,m}_\mu(h_{<t}a_t o_t r_t)\right)\right) - \gamma V^{\pi,m}_\mu(h_{<t}a_t o_t)$$
$$\overline{\mu}(o_t 0|h_{<t}a_t) := 1 - \overline{\mu}(o_t 1|h_{<t}a_t).$$

Furthermore, we say that a policy π is reward history independent if $\pi(a_t|h_{<t}) = \pi(a_t|a_1 o_1 \ldots a_{t-1}o_{t-1})$ for all $h_{<t}$ and a_t. We now state our main result regarding the possibility of representation for $\mu \in \mathcal{M}^*$.

Theorem 7 *If π is reward history independent then for all $\mu \in \mathcal{M}^*$, all $h_{<t}$ and all choices of history binarising function $\langle\cdot\rangle$ we have*

$$V_{\overline{\mu}}^{\pi,m}(\langle h_{<t}\rangle) = V_{\mu}^{\pi,m}(h_{<t}).$$

Proof sketch: expand the definitions of the value function and $\overline{\mu}$ and apply Lemma 17.

An immediate consequence of this is that the optimal agent for $\overline{\mu}$ is optimal for the environment μ.

Corollary 8 *If $\mu \in \mathcal{M}^*$ then $\pi^*_{\overline{\mu}} \in \arg\max_\pi V_\mu^\pi$.*

We will now show that some well studied environment classes in reinforcement learning are subclasses of $\mathcal{M}^*$.

Proposition 9 $\mathcal{M}_{MDP}, \mathcal{M}_{kMDP}, \mathcal{M}_{POMDP} \subseteq \mathcal{M}^*$.

These follow trivially from the definition of $\mathcal{M}^*$, as in MDPs, k-MDPs, and POMDPs the transitions probabilities only depend on the previous observations and actions, and not the rewards.

5. The case of Bayesian RL

As discussed earlier, the Bayesian solution to the general reinforcement learning problem involves considering a Bayesian mixture over the class of environments in the case where μ is unknown. In this case it is well known that predictive distribution of the mixture will converge to the true environment if the true environment is contained within the mixture class. In this section we show that reward binarisation preserves this property. This result covers the case of finite mixtures, but also generalises to idealised Bayesian agents such as AIXI [Hut05] whose mixture class contains countably many environments.

To show this, it is sufficient to show that the mixture over binarised environments $\xi_{\overline{\mathcal{M}}}$ and the binarised form of the mixture $\overline{\xi}$ dominates all semimeasures in $\mathcal{M}$.

Theorem 10 *For all $\mu \in \mathcal{M}$, there exists a $c > 0$ such that for all $h_{<t} \in \mathcal{H}$ we have*

$$\xi_{\overline{\mathcal{M}}}(\langle h_{<t}\rangle) > c\mu(h_{<t})$$

That is, $\xi_{\overline{\mathcal{M}}}$ dominates all semimeasures in $\mathcal{M}$.

Proof idea: We show that for every possible reward string there exists a enumerable binary environment such that that binary environment with that reward string corresponds to the chosen μ.

Next we consider $\overline{\xi}$, the binarised version of $\xi_{\mathcal{M}}$, using Definition 1. To show the dominance of $\overline{\xi}$ we have to show that there exists a valid set of priors such that $\xi_{\overline{\mathcal{M}}} = \overline{\xi}$. To this end, we prove the following lemmas about $\mathcal{M}$ and $\overline{\mathcal{M}}$.

Lemma 11 *For all $\rho \in \mathcal{M}$, $\forall o, r, h, a.\ r \notin \{0,1\} \Rightarrow \rho(or|ha) = 0$ if and only if $\overline{\rho} = \rho$.*

Proof ($\Rightarrow$) If $\forall o, r, h, a.\ r \notin \{0,1\} \Rightarrow \rho(or|ha) = 0$, then from the definition of $\overline{\rho}$, we have for reward 1

$$\begin{aligned}
\overline{\rho}(o_t 1|h_{<t}a_t) &= \sum_{r_t \in \mathcal{R}} \rho(o_t r_t|h_{<t}a_t) r_t \\
&= \sum_{r_t \in \{0,1\}} \rho(o_t r_t|h_{<t}a_t) r_t \\
&= \rho(o_t 0|h_{<t}a_t) \cdot 0 + \rho(o_t 1|h_{<t}a_t) \cdot 1 \\
&= \rho(o_t 1|h_{<t}a_t),
\end{aligned}$$

and for reward 0

$$\begin{aligned}\overline{\rho}(o_t 0|h_{<t}a_t) &= \left(\sum_{r_t\in\mathcal{R}} \rho(o_t r_t|h_{<t}a_t)\right) - \overline{\rho}(o_t 1|h_{<t}a_t)\\ &= \left(\sum_{r_t\in\{0,1\}} \rho(o_t r_t|h_{<t}a_t)\right) - \overline{\rho}(o_t 1|h_{<t}a_t)\\ &= \left(\sum_{r_t\in\{0,1\}} \rho(o_t r_t|h_{<t}a_t)\right) - \rho(o_t 1|h_{<t}a_t)\\ &= \rho(o_t 0|h_{<t}a_t).\end{aligned}$$

Therefore $\overline{\rho} = \rho$.

($\Leftarrow$) If $\overline{\rho} = \rho$, then on all $r \notin \{0,1\}$ we have $\rho(or|ha) = \overline{\rho}(or|ha) = 0$. ■

Lemma 12 $\overline{\mathcal{M}} = \{\overline{\mu} \mid \mu \in \mathcal{M}\}$.

Proof We will prove this by showing that $\{\overline{\mu}|\mu \in \mathcal{M}\} \subseteq \overline{\mathcal{M}}$ and $\overline{\mathcal{M}} \subseteq \{\overline{\mu}|\mu \in \mathcal{M}\}$. For all $\mu \in \mathcal{M}$, we have that $\overline{\mu}$ only produces binary rewards. This means that all $\overline{\mu} \in \overline{\mathcal{M}}$, and therefore $\{\overline{\mu}|\mu \in \mathcal{M}\} \subseteq \overline{\mathcal{M}}$.

We have $\overline{\mathcal{M}} = \{\mu \in \mathcal{M}|\mu \text{ only outputs reward 0 and 1}\} = \{\mu \in \mathcal{M}|\forall o, r, h, a.\ r \notin \{0,1\} \Rightarrow \mu(or|ha) = 0\}$. Therefore, for an arbitrary $\rho \in \overline{\mathcal{M}} = \{\mu \in \mathcal{M}|\forall o, r, h, a.\ r \notin \{0,1\} \Rightarrow \mu(or|ha) = 0\}$ we also have that $\rho \in \mathcal{M}$. Since ρ only outputs rewards of 0 and 1, we also have by Lemma 11 that $\overline{\rho} = \rho$, and $\overline{\rho} \in \{\overline{\mu}|\mu \in \mathcal{M}\}$, therefore $\rho \in \{\overline{\mu}|\mu \in \mathcal{M}\}$, hence $\overline{\mathcal{M}} \subseteq \{\overline{\mu}|\mu \in \mathcal{M}\}$. Since we also have that $\{\overline{\mu}|\mu \in \mathcal{M}\} \subseteq \overline{\mathcal{M}}$, it must be the case that $\{\overline{\mu}|\mu \in \mathcal{M}\} = \overline{\mathcal{M}}$. ■

Lemma 13 *For all sets of lower semicomputable priors $\{w_\mu\}_{\mu\in\mathcal{M}}$, for the Bayes Mixture $\xi_\mathcal{M}$, there exists a set of lower semicomputable priors $\{w'_\rho\}_{\rho\in\overline{\mathcal{M}}}$ for the Bayes Mixture $\xi_{\overline{\mathcal{M}}}$ such that for all $h_{1:t} \in \mathcal{H}$ we have*

$$\xi_{\overline{\mathcal{M}}}(\langle h_{1:t}\rangle) = \overline{\xi}_\mathcal{M}(\langle h_{1:t}\rangle).$$

Proof idea: We define a binary environment prior that is the sum of the corresponding non-binary environment priors.

Finally, using Lemma 13 we can show that binarised reward transformation of the Bayesian mixture $\xi_\mathcal{M}$ is, like $\xi_{\overline{\mathcal{M}}}$, dominates all semimeasures in $\mathcal{M}$.

Theorem 14 *$\overline{\xi}$ dominates all semimeasures in $\mathcal{M}$*

Proof This is a direct consequence of Theorem 10 and Lemma 13. ■

We have now shown that both $\xi_{\overline{\mathcal{M}}}$ and $\overline{\xi}$ dominate all semimeasures in $\mathcal{M}$. This means that both of these measures universal semimeasures and will converge to the true semimeasure μ on histories generated by μ [RH11].

6. Binary Reward (Multiple bits)

In the previous sections we have discussed how we can turn the reward into a single bit and the effect of this on the environment and the agent. In this section we will discuss various methods for transforming the reward into several bits, but only giving the agent a single bit at a time.

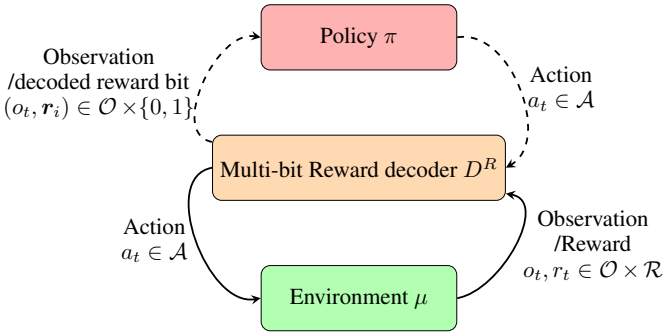

Figure 2: Agent-environment loop with multi-bit reward.

We will start with the overall formulation, and then later go on to various binarising methods and their effects. Let $\widehat{\mathcal{H}}$ denote the set of histories with only binary rewards, that is $\widehat{\mathcal{H}} := \bigcup_{t=0}^{\infty} (\mathcal{A} \times \mathcal{O} \times \{0,1\})^t$.

We will define an environment transformation $\widehat{\mu}$ in a similar fashion to Definition 20, starting with a function which maps multiple-bit binarised histories to original histories. Figure 2 shows the setup.

We are going to split up the multiple-bit binary reward environment into two parts: the first part, $\widehat{\mu}_1$, at the point when the true environment outputs the true reward and true observation, and the other part, $\widehat{\mu}_2$, when reward is being decoded. We will use $\boldsymbol{r}$ for the binary form of the true reward from the environment $\mathcal{R}$. We will use d for the length of the binarised rewards; $\boldsymbol{r}$ will always have length d. We will use o for the true observation from the environment, while decoding the reward $\widehat{\mu}_2$ will produce the current observation. We will use the function $\psi : \widehat{\mathcal{H}} \to \mathcal{H}$ to represent translations from the binarised reward histories to the original histories.

$$\psi(h_{1:td}) := \psi(h_{1:(t-1)d}) a_{(t-1)d} o_{(t-1)d} C^R(r_{(t-1)d:td}) \text{ and } \psi(\epsilon) = \epsilon$$

and $\psi(h) = \perp$ (undefined) if $\ell(h) \neq td$ for some t, or if $o_{(j-1)d+i} \neq o_{(j-1)d}$ for any i such that $0 \leq i \leq d-1$ and any j such that $1 \leq j \leq t$. The function $C^R : \{0,1\}^d \to \mathcal{R}$ turns the binary form of the rewards to the real-numbered reward in $\mathcal{R}$. This is fine, since we will only be using ψ in cases when the histories have length td for some t.

The inverse of ψ, ψ^{-1}, is not well defined since the choices of actions in the expansion of the history may not match the choices of actions of the agent.

Definition 15 (Environment with Binarisation of Rewards into Several Bits) *Given history $h_{1:td}(aor)_{<k}$, if for all $1 \leq j \leq t$ we have $\psi(h_{1:jd}) \neq \perp$, then*

$$\widehat{\mu}(h_{1:td}(aor)_{<k}) := \prod_{j=1}^{t-1} \mu(o_{jd} C^R(r_{jd:(j+1)d}) | \psi(h_{1:jd}) a_{jd+1}) \prod_{i=1}^{k-1} \sum_{\boldsymbol{r} \in \mathcal{R}, o \in \mathcal{O}} \mu(o\boldsymbol{r} | \psi(h_{1:td}) a_{td+1}) [\![r_{td+i} = \boldsymbol{r}_i]\!] [\![o_{td+i} = o]\!]$$

else $\widehat{\mu}(h_{1:td}(aor)_{<k}) = 0$.

In this setup we are using a fixed number of maximum bits for the binary encodings of the rewards. This lets us include any possible choices for rewards in $[0,1]$ with the caveat that we may not be able to represent all the rewards exactly.

6.1 The problem of $\gamma(t)$

In the multiple-bit binary reward setting, the agent receives these multiple bits over multiple timesteps in the form of single bits. At a given timestep the agent receives a single bit corresponding to the part of the multiple bits representing

the reward; the agent also considers the future reward bits it will receive. Specifically, it does so with a discount factor $\gamma(t)$. This means that the agent does not consider the multiple-bit reward of, for example, 10010 as 0.10010, but instead as $1 + \gamma(t) \cdot 0 + \gamma(t+1) \cdot 0 + \gamma(t+2) \cdot 1 + \gamma(t+3) \cdot 0$.

In binary we could have the multiple bit rewards 100 and 011, and if $\gamma(t) = 1$ then $1+1\cdot0+1\cdot0 < 0+1\cdot1+1\cdot1$, even though $100 > 011$. One simple solution would be to require $\gamma(t) \geq 2\gamma(t+1)$ for all t. This is a strong assumption, and will mean that the agent does not plan very far ahead. However, it is possible to do better.

If we start with some discount function $\gamma(t)$, we can use an updated discount function $\gamma'(t) := \gamma(n) \cdot 2^{-m}$ where $t = d \cdot n + m$ and $m < d$ (d is the number of bits we use). This will give us exact representation if the multiple-bit reward is the binary form of the reward.

6.2 Multiple-bit value function

We can show that if the policy "does not care" about actions it took during the reward decoding then we can get equality in the value functions, since the multiple-bit binary environment automatically does not care about the actions taken during the reward decoding.

Let $\widehat{V}$ be the value function V, with γ replaced by γ'.

Theorem 16 *If π is such that $\pi(a|h) = \pi(a|\psi(h))$ for all a, and all histories $h \in \widehat{\mathcal{H}}$ of length td for some t, then we have that for all environments μ and histories $h_{1:td} \in \widehat{\mathcal{H}}$,*

$$\widehat{V}^{\pi,m}_{\widehat{\mu}}(h_{1:td}) = V^{\pi,m}_{\mu}(\psi(h_{1:td})). \tag{1}$$

Proof Comes straight from the Definitions of ψ and $\widehat{\mu}$. ■

It could be argued that the assumption that $\pi(a|h) = \pi(a|\psi(h))$ for all a, and all histories $h \in \widehat{\mathcal{H}}$ of length td for some t, is quite strong. However, we found this assumption difficult to relax, which suggests that one requires some structural assumption on the policy to state anything meaningful.

7. Discussion and Future work

An immediate consequence of Theorem 7 is that all non-trivial choices of reward set are equivalent in this value function sense, as they can all be reduced to the binary reward set. This could be interpreted as showing that all reward sets are valid, however, as we previously mentioned there are situations such as generative policy evaluation and upside-down RL where binary reward sets would be preferred.

One possible extension of this work would be to try to achieve a result similar to Theorem 23 for single-bit or multiple-bit binary rewards. Specifically we may be able to show that an agent which is close to optimal in the binarised case is also close to optimal in the original environment, as the negative result given by Theorem 4 only precludes the possibility of exact representation.

It would be useful to, on top of the binary action and reward space, have a binary observation space in one complete framework. This can be done easily for the observation space, however, the combination of observation and reward space may lead to similar problems which we faced in the binarised reward setting.

8. Conclusion

In this work we have analysed a number of natural reward binarisation schemes. While in general we show that exact representation of the value function is impossible, we found that reward binarisation can be applied in many important special cases without a loss of representation power. Additionally, we considered the interaction of binarisation with Bayesian reinforcement learning, providing new technical results that justify the Bayesian approach in the presence of specific types of binarisation.

Acknowledgments

We thank Csaba Szepesvari, Qinghua Liu, Amy Zhang and the anonymous reviewers for their comments and feedback.

References

[AASS22] Kai Arulkumaran, Dylan R Ashley, Jürgen Schmidhuber, and Rupesh K Srivastava. All you need is supervised learning: From imitation learning to meta-rl with upside down rl. *arXiv preprint arXiv:2202.11960*, 2022.

[BB61] Keller Breland and Marian Breland. The misbehavior of organisms. *American psychologist*, 16(11):681, 1961.

[CLR+21] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34, 2021.

[Hut05] Marcus Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2005. 300 pages, http://www.hutter1.net/ai/uaibook.htm.

[Hut06] Marcus Hutter. On the foundations of universal sequence prediction. In *International Conference on Theory and Applications of Models of Computation*, pages 408–420. Springer, 2006.

[JLL21] Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34, 2021.

[KLC98] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[KLM96] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[Lei16] Jan Leike. *Nonparametric general reinforcement learning*. PhD thesis, The Australian National University (Australia), 2016.

[Maj21] Sultan J Majeed. Abstractions of general reinforcement learning. *arXiv preprint arXiv:2112.13404*, 2021.

[MH21] Sultan J Majeed and Marcus Hutter. Exact reduction of huge action spaces in general reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8874–8883, 2021.

[Pav27] Ivan P Pavlov. Conditioned reflexes (translated by gv anrep). *London: Oxford*, 1927.

[RH11] Samuel Rathmanner and Marcus Hutter. A philosophical treatise of universal induction. *Entropy*, 13(6):1076–1136, 2011.

[VBH+15] Joel Veness, Marc G Bellemare, Marcus Hutter, Alvin Chua, and Guillaume Desjardins. Compress and control. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Appendix A.

A.1 Proof of Lemma 3

Proof We will prove a more general result and show that there are infinitely many choices of μ that satisfy $V_{\mu'}^{\pi,m}(h) = q$ for an arbitrary history h. We do this by showing it is true at the maximum horizon m, then work backwards.

Let $q(h)$ denote the q such that $(h, q) \in \mathcal{Q}$.

$$\mu'(o_t 1|h_{<t}a_t) := \begin{cases} \frac{q(h_{<m})}{|\mathcal{O}|} & \text{if } t = m \\ \frac{\frac{q(h_{<t})}{|\mathcal{O}|} - \gamma q(h_{<t}a_t o_t 0)}{1+\gamma q(h_{<t}a_t o_t 1) - \gamma q(h_{<t}a_t o_t 0)} & \text{if } t < m \end{cases}$$
$$\mu'(o_t 0|h_{<t}a_t) := 1 - \mu'(o_t 1|h_{<t}a_t).$$

In the case that $t = m$,

$$\begin{aligned} V_{\mu'}^{\pi,m}(h_{<m}) &= \sum_{a_m \in \mathcal{A}} \pi(a_m|h_{<m}) \sum_{o_m r_m} \mu'(o_m r_m|h_{<m}a_m)(r_m) \\ &= \sum_{a_m \in \mathcal{A}} \pi(a_m|h_{<m}) q(h_{<m}) \\ &= q(h_{<m}) \sum_{a_m \in \mathcal{A}} \pi(a_m|h_{<m}) \\ &= q(h_{<m}). \end{aligned}$$

In the second case that $t < m$, we can work backwards from $t = m$.

$$\begin{aligned} V_{\mu'}^{\pi,m}(h_{<t}) &= \sum_{a_t \in \mathcal{A}} \pi(a_t|h_{<t}) \sum_{o_t r_t} \mu'(o_t r_t|h_{<t}a_t)(r_t + \gamma V_{\mu'}^{\pi,m}(h_{<t}a_t o_t r_t)) \\ &= \sum_{a_t \in \mathcal{A}} \pi(a_t|h_{<t}) \sum_{o_t r_t} \mu'(o_t r_t|h_{<t}a_t)(r_t + \gamma q(h_{<t}a_t o_t r_t)) \\ &= \sum_{a_t \in \mathcal{A}} \pi(a_t|h_{<t}) \sum_{o_t} ((1 - \mu'(o_t 1|h_{<t}a_t))(\gamma q(h_{<t}a_t o_t 0)) + \mu'(o_t 1|h_{<t}a_t)(1 + \gamma q(h_{<t}a_t o_t 1))) \\ &= \sum_{a_t \in \mathcal{A}} \pi(a_t|h_{<t}) \sum_{o_t} (\gamma q(h_{<t}a_t o_t 0) + \mu'(o_t 1|h_{<t}a_t)(1 + \gamma q(h_{<t}a_t o_t 1) - \gamma q(h_{<t}a_t o_t 0))) \\ &= \sum_{a_t \in \mathcal{A}} \pi(a_t|h_{<t}) \sum_{o_t} \left(\gamma q(h_{<t}a_t o_t 0) + \frac{q(h_{<t})}{|\mathcal{O}|} - \gamma q(h_{<t}a_t o_t 0)) \right) \\ &= \sum_{a_t \in \mathcal{A}} \pi(a_t|h_{<t}) q(h_{<t}) \\ &= q(h_{<t}). \end{aligned}$$

■

A.2 Proof of Theorem 7

Before we show that environments in $\mathcal{M}^*$ can be represented in by binary reward environments we will need a small lemma about $\mathcal{M}^*$.

Lemma 17 *If π is reward history independent, that is, $\pi(a|h) = \pi(a|a_1 o_1 \dots a_{t-1} o_{t-1})$ then for all $\mu \in \mathcal{M}^*$ and $h_{<t}$ we have that*

$$V_\mu^\pi(\langle h_{<t}\rangle) = V_\mu^\pi(h_{<t})$$

and

$$V_\mu^\pi(h_{<t}) = V_\mu^\pi(h_{<t-1} a_{t-1} o_{t-1})$$

Proof This comes from the fact that the distributions in the expectation do not depend on the reward components of the history. ■

Proof of Theorem 7.

Proof In the $t = m$ case:

$$\begin{aligned}
V_{\overline{\mu}}^{\pi,m}(\langle h_{<m}\rangle) &= \sum_{a_m \in \mathcal{A}} \pi(a_m|\langle h_{<m}\rangle) \sum_{o_m r_m} \overline{\mu}(o_m r_m|\langle h_{<m}\rangle a_m)(r_m) \\
&= \sum_{a_m \in \mathcal{A}} \pi(a_m|\langle h_{<m}\rangle) \sum_{o_m} \overline{\mu}(o_m 1|\langle h_{<m}\rangle a_m) \\
&= \sum_{a_m \in \mathcal{A}} \pi(a_m|\langle h_{<m}\rangle) \sum_{o_m} \sum_{r_m \in \mathcal{R}} \mu(o_m r_m|\langle h_{<m}\rangle a_m) r_m \\
&= V_\mu^{\pi,m}(\langle h_{<m}\rangle) \\
&= V_\mu^{\pi,m}(h_{<m})
\end{aligned}$$

In the second case that $t < m$, we can work backwards from $t = m$.

$$\begin{aligned}
V_{\overline{\mu}}^{\pi,m}(\langle h_{<t}\rangle) &\overset{(a)}{=} \sum_{a_t \in \mathcal{A}} \pi(a_t|\langle h_{<t}\rangle) \sum_{o_t r_t} \overline{\mu}(o_t r_t|\langle h_{<t}\rangle a_t)(r_t + \gamma V_{\overline{\mu}}^{\pi,m}(\langle h_{<t}\rangle a_t o_t r_t)) \\
&\overset{(b)}{=} \sum_{a_t \in \mathcal{A}} \pi(a_t|\langle h_{<t}\rangle) \sum_{o_t r_t} \overline{\mu}(o_t r_t|\langle h_{<t}\rangle a_t)(r_t + \gamma V_{\overline{\mu}}^{\pi,m}(\langle h_{<t}\rangle a_t o_t)) \\
&\overset{(c)}{=} \sum_{a_t \in \mathcal{A}} \pi(a_t|\langle h_{<t}\rangle) \sum_{o_t} \Big(\overline{\mu}(o_t 0|\langle h_{<t}\rangle a_t)(\gamma V_{\overline{\mu}}^{\pi,m}(\langle h_{<t}\rangle a_t o_t)) + \overline{\mu}(o_t 1|\langle h_{<t}\rangle a_t)(1 + \gamma V_{\overline{\mu}}^{\pi,m}(\langle h_{<t}\rangle a_t o_t)) \Big) \\
&\overset{(d)}{=} \sum_{a_t \in \mathcal{A}} \pi(a_t|\langle h_{<t}\rangle) \sum_{o_t} \Big((1 - \overline{\mu}(o_t 1|\langle h_{<t}\rangle a_t))(\gamma V_{\overline{\mu}}^{\pi,m}(\langle h_{<t}\rangle a_t o_t)) + \overline{\mu}(o_t 1|\langle h_{<t}\rangle a_t)(1 + \gamma V_{\overline{\mu}}^{\pi,m}(\langle h_{<t}\rangle a_t o_t)) \Big) \\
&\overset{(e)}{=} \sum_{a_t \in \mathcal{A}} \pi(a_t|\langle h_{<t}\rangle) \sum_{o_t} \Big(\overline{\mu}(o_t 1|\langle h_{<t}\rangle a_t) + \gamma V_{\overline{\mu}}^{\pi,m}(\langle h_{<t}\rangle a_t o_t) \Big) \\
&\overset{(f)}{=} \sum_{a_t \in \mathcal{A}} \pi(a_t|\langle h_{<t}\rangle) \sum_{o_t} \big(\overline{\mu}(o_t 1|\langle h_{<t}\rangle a_t) + \gamma V_{\mu}^{\pi,m}(\langle h_{<t}\rangle a_t o_t) \big) \\
&\overset{(g)}{=} \sum_{a_t \in \mathcal{A}} \pi(a_t|\langle h_{<t}\rangle) \sum_{o_t} \left(\sum_{r_t \in \mathcal{R}} \mu(o_t r_t|\langle h_{<t}\rangle a_t) \left(r_t + \gamma V_{\mu}^{\pi,m}(\langle h_{<t}\rangle a_t o_t r_t) \right) \right) \\
&\overset{(h)}{=} V_\mu^{\pi,m}(\langle h_{<t}\rangle) \\
&\overset{(i)}{=} V_\mu^{\pi,m}(h_{<t})
\end{aligned}$$

(a) comes from the Bellman equation of V. (b) comes from Lemma 17. (c) involves expanding the reward sum for the two rewards of 0 and 1. (d) and (e) are algebra. (f) comes from the $t + 1$ case working backwards. (g) comes from Definition 6. (h) is the Bellman equation of V. (i) is Lemma 17. This completes the proof. ■

A.3 Proof of Theorem 10

Proof We can consider a binary reward environment for each possible $r_{1:t} \in \mathcal{R}^{t-1}$.

Let $\mu_{r,r'}$ be defined on the observation space as

$$\mu_{r,r'}(o_k|\langle h_{<k}\rangle a_k) := \begin{cases} \mu(o_k|h'_{<k}a_k) & \text{if } \langle r'\rangle = \langle r_{<k}\rangle \\ \frac{1}{|\mathcal{O}|} & \text{otherwise} \end{cases}$$

and defined on the reward space as

$$\mu_{r,r'}(\langle r_k\rangle|\langle h_{<k}\rangle a_k o_k) = \begin{cases} \mu(r|h'_{<k}a_k o_k) & \text{if } \langle r'\rangle\langle r\rangle = \langle r_{<k}\rangle\langle r_k\rangle \\ 1-\mu(r|h'_{<k}a_k o_k) & \text{if } \langle r'\rangle = \langle r_{<k}\rangle \wedge \langle r\rangle \neq \langle r_k\rangle \\ \frac{1}{2} & \text{otherwise} \end{cases}$$

where $h'_{<k}$ is $h_{<k}$ with the reward sequence replaced by r'. Since μ is enumerable, $\mu_{r,r'}$ is also enumerable for each r, r'. Note that r' is a string of rewards and r is a single reward.

Therefore we have

$$\begin{aligned}
\xi_{\overline{\mathcal{M}}}(\langle h_{<t}\rangle) &= \sum_{\nu\in\overline{\mathcal{M}}} w(\nu)\nu(\langle h_{<t}\rangle) \\
&> \sum_{r\in\mathcal{R},r'\in\mathcal{R}^*} w(\mu_{r,r'})\mu_{r,r'}(\langle h_{<t}\rangle) \\
&= \sum_{r\in\mathcal{R},r'\in\mathcal{R}^*} w(\mu_{r,r'}) \prod_{k=1}^{t-1} \mu_{r,r'}(o_k\langle r_k\rangle|\langle h_{<k}\rangle a_k) \\
&= \sum_{r\in\mathcal{R},r'\in\mathcal{R}^*} w(\mu_{r,r'}) \\
&\cdot \prod_{k=1}^{t-1} \mu_{r,r'}(o_k|\langle h_{<k}\rangle a_k)\mu_{r,r'}(\langle r_k\rangle|\langle h_{<k}\rangle a_k o_k) \\
&> w(\mu_{r_{t-1},r_{1:t-2}}) \\
&\cdot \prod_{k=1}^{t-1} \mu_{r_{t-1},r_{1:t-2}}(o_k|\langle h_{<k}\rangle a_k)\mu_{r_{t-1},r_{1:t-2}}(\langle r_k\rangle|\langle h_{<k}\rangle a_k o_k) \\
&= w(\mu_{r_{t-1},r_{1:t-2}}) \prod_{k=1}^{t-1} \mu(o_k|h_{<k}a_k)\mu(r_k|h_{<k}a_k o_k) \\
&= w(\mu_{r_{t-1},r_{1:t-2}})\mu(h_{<t})
\end{aligned}$$

where $\mathcal{R}^* = \cup_{t=0}^{\infty} R^t$.

If we choose $c = \sup\{w(\mu_{r,r'}) \mid r \in \mathcal{R} \wedge r' \in \mathcal{R}^*\}$, then we have $\xi_{\overline{\mathcal{M}}}(\langle h_{<t}\rangle) > c\mu(h_{<t})$. ■

A.4 Proof of Lemma 13

Proof Since $\overline{\mathcal{M}} = \{\overline{\mu} | \mu \in \mathcal{M}\}$, for every element $\rho \in \overline{\mathcal{M}}$ there may be some (possiblly an infinite number of) $\mu \in \mathcal{M}$ such that $\overline{\mu} = \rho$.

If we set the prior $w'_\rho = \sum_{\mu \in \mathcal{M}: \overline{\mu} = \rho} w_\mu$, we will get equality between the mixtures $\xi_{\overline{\mathcal{M}}}(o\langle r\rangle | \langle h_{<t}\rangle a) = \overline{\xi}_{\mathcal{M}}(o\langle r\rangle | \langle h_{<t}\rangle a)$.

To show this equality we will first need to prove that

$$\sum_{\rho \in \overline{\mathcal{M}}} w'_\rho \prod_{i=1}^{t-1} \rho(o_i\langle r_i\rangle | \langle h_{<i}\rangle a_i) = \sum_{\mu \in \mathcal{M}} w_\mu \prod_{i=1}^{t-1} \overline{\mu}(o_i\langle r_i\rangle | \langle h_{<i}\rangle a_i). \tag{2}$$

To do this we just expand our definition of w'_ρ and rearrange.

$$\begin{aligned}
\sum_{\rho \in \overline{\mathcal{M}}} w'_\rho \prod_{i=1}^{t-1} \rho(o_i\langle r_i\rangle | \langle h_{<i}\rangle a_i) &= \sum_{\rho \in \overline{\mathcal{M}}} \left(\sum_{\mu \in \mathcal{M}: \overline{\mu} = \rho} w_\mu \right) \prod_{i=1}^{t-1} \rho(o_i\langle r_i\rangle | \langle h_{<i}\rangle a_i) \\
&= \sum_{\rho \in \overline{\mathcal{M}}} \left(\sum_{\mu \in \mathcal{M}: \overline{\mu} = \rho} w_\mu \prod_{i=1}^{t-1} \rho(o_i\langle r_i\rangle | \langle h_{<i}\rangle a_i) \right) \\
&= \sum_{\rho \in \overline{\mathcal{M}}} \left(\sum_{\mu \in \mathcal{M}: \overline{\mu} = \rho} w_\mu \prod_{i=1}^{t-1} \overline{\mu}(o_i\langle r_i\rangle | \langle h_{<i}\rangle a_i) \right) \\
&= \sum_{\mu \in \mathcal{M}} w_\mu \prod_{i=1}^{t-1} \overline{\mu}(o_i\langle r_i\rangle | \langle h_{<i}\rangle a_i).
\end{aligned}$$

Now that we have Equation 2 we can prove equality of $\xi_{\overline{\mathcal{M}}}$ and $\overline{\xi}$.

$$\begin{aligned}
\xi_{\overline{\mathcal{M}}}(o\langle r\rangle | \langle h_{<t}\rangle a) &= \frac{\sum_{\rho \in \overline{\mathcal{M}}} w'_\rho \rho(o\langle r\rangle | \langle h_{<t}\rangle a) \prod_{i=1}^{t-1} \rho(o_i\langle r_i\rangle | \langle h_{<i}\rangle a_i)}{\sum_{\nu \in \overline{\mathcal{M}}} w'_\nu \prod_{i=1}^{t-1} \nu(o_i\langle r_i\rangle | \langle h_{<i}\rangle a_i)} \\
&= \frac{\sum_{\mu \in \mathcal{M}} w_\mu \overline{\mu}(o\langle r\rangle | \langle h_{<t}\rangle a) \prod_{i=1}^{t-1} \overline{\mu}(o_i\langle r_i\rangle | \langle h_{<i}\rangle a_i)}{\sum_{\mu \in \mathcal{M}} w_\mu \prod_{i=1}^{t-1} \overline{\mu}(o_i\langle r_i\rangle | \langle h_{<i}\rangle a_i)} \\
&= \overline{\xi}(o\langle r\rangle | \langle h_{<t}\rangle a).
\end{aligned}$$

Therefore for all histories $h_{1:t} \in \mathcal{H}$

$$\xi_{\overline{\mathcal{M}}}(\langle h_{1:t}\rangle) = \prod_{k=1}^{t} \xi_{\overline{\mathcal{M}}}(o_k\langle r_k\rangle | \langle h_{<k}\rangle a_k) = \prod_{k=1}^{t} \overline{\xi}(o_k\langle r_k\rangle | \langle h_{<k}\rangle a_k) = \overline{\xi}(\langle h_{1:t}\rangle).$$

We now need to show that the chosen prior w'_ρ is lower semicomputable.

To compute w'_ρ, we go through the enumeration of $\mu \in \mathcal{M}$ and do a dovetailed proof search on $\overline{\mu} = \rho$, then if we have a proof that $\overline{\mu} = \rho$ we add w_μ to the sum. If there exists a proof such that $\overline{\mu} = \rho$, we will find it in finite steps of the proof search. Since w_μ is lower semicomputable, the sum will also be lower semicomputable. ■

A.5 Proof of Theorem 16

We first need to prove the following lemma.

Additionally we have from the definition of ψ that $\psi(h_{1:td}\overline{\boldsymbol{aor}}) = \psi(h_{1:td})\boldsymbol{a}_0 o\boldsymbol{r}$.

Lemma 18 *We can write the binarised value function, $\widehat{V}^{\pi,m}_{\widehat{\mu}}$, for d steps with the recursive form of*

$$\widehat{V}^{\pi,m}_{\widehat{\mu}}(h_{1:td}) = \sum_{\boldsymbol{a}\in\mathcal{A}^d}\sum_{o\in\mathcal{O},\boldsymbol{r}\in\mathcal{R}}\prod_{k=0}^{d-1}\pi(\boldsymbol{a}_k|h_{1:(td+k)})\mu(o\boldsymbol{r}|\psi(h_{1:td})\boldsymbol{a}_0)\left(\gamma(t)\boldsymbol{r} + \widehat{V}^{\pi,m}_{\widehat{\mu}}(h_{1:td}\overline{\boldsymbol{aor}})\right) \tag{3}$$

where $\overline{\boldsymbol{aor}} = \boldsymbol{a}_0 o\boldsymbol{r}_0 \dots \boldsymbol{a}_{d-1} o\boldsymbol{r}_{d-1}$.

Proof

$$\begin{aligned}
\widehat{V}^{\pi,m}_{\widehat{\mu}}(h_{1:td}) &= \sum_{a_{td}\in\mathcal{A}}\pi(a_{td}|h_{1:td})\sum_{o_{td},r_{td}}\widehat{\mu}(o_{td}r_{td}|h_{1:td}a_{td},\boldsymbol{r},o)\left(\gamma'(td)r_{td} + \widehat{V}^{\pi,m}_{\widehat{\mu}}(h_{1:td}a_{td}o_{td}r_{td})\right)\\
&= \sum_{a_{td}\in\mathcal{A}}\sum_{o_{td},r_{td}}\cdots\sum_{a_{(t+1)d-1}\in\mathcal{A}}\sum_{o_{(t+1)d-1},r_{(t+1)d-1}}\\
&\prod_{k=0}^{d-1}\pi(a_{td+k}|h_{1:(td+k)})\prod_{k=0}^{d-1}\widehat{\mu}(o_{td+k}r_{td+k}|h_{1:(td+k)}a_{td+k},\boldsymbol{r},o)\\
&\cdot\left(\sum_{i=1}^{d}\gamma'(td+i)r_{td+i} + \widehat{V}^{\pi,m}_{\widehat{\mu}}(h_{<(t+1)d})\right)\\
&= \sum_{a_{td}\in\mathcal{A}}\cdots\sum_{a_{(t+1)d-1}\in\mathcal{A}}\sum_{o_{td},\boldsymbol{r}}\\
&\prod_{k=0}^{d-1}\pi(a_{td+k}|h_{1:(td+k)})\mu(o_{td}\boldsymbol{r}|\psi(h_{1:td})a_{td})\left(\gamma(t)\boldsymbol{r} + \widehat{V}^{\pi,m}_{\widehat{\mu}}(h_{1:(t+1)d})\right)\\
&= \sum_{\boldsymbol{a}\in\mathcal{A}^d}\sum_{o\in\mathcal{O},\boldsymbol{r}\in\mathcal{R}}\prod_{k=0}^{d-1}\pi(\boldsymbol{a}_k|h_{1:(td+k)})\mu(o\boldsymbol{r}|\psi(h_{1:td})\boldsymbol{a}_0)\left(\gamma(t)\boldsymbol{r} + \widehat{V}^{\pi,m}_{\widehat{\mu}}(h_{1:td}\overline{\boldsymbol{aor}})\right).
\end{aligned}$$

■

Proof of Theorem 16.

Proof Final case, when $t = m$

$$\begin{aligned}
\widehat{V}^{\pi,m}_{\widehat{\mu}}(h_{1:md}) &= \sum_{\boldsymbol{a}\in\mathcal{A}^d}\sum_{o\in\mathcal{O},\boldsymbol{r}\in\mathcal{R}}\prod_{k=0}^{d-1}\pi(\boldsymbol{a}_k|h_{1:(md+k)})\mu(o\boldsymbol{r}|\psi(h_{1:md})\boldsymbol{a}_0)\left(\gamma(m)\boldsymbol{r}\right)\\
&= \sum_{\boldsymbol{a}_0\in\mathcal{A}}\sum_{o\in\mathcal{O},\boldsymbol{r}\in\mathcal{R}}\pi(\boldsymbol{a}_0|h_{1:md})\mu(o\boldsymbol{r}|\psi(h_{1:md})\boldsymbol{a}_0)\left(\gamma(m)\boldsymbol{r}\right)\\
&= \sum_{\boldsymbol{a}_0\in\mathcal{A}}\pi(\boldsymbol{a}_0|h_{1:md})\sum_{o\in\mathcal{O},\boldsymbol{r}\in\mathcal{R}}\mu(o\boldsymbol{r}|\psi(h_{1:md})\boldsymbol{a}_0)\left(\gamma(m)\boldsymbol{r}\right)\\
&= \sum_{\boldsymbol{a}_0\in\mathcal{A}}\pi(\boldsymbol{a}_0|\psi(h_{1:md}))\sum_{o\in\mathcal{O},\boldsymbol{r}\in\mathcal{R}}\mu(o\boldsymbol{r}|\psi(h_{1:md})\boldsymbol{a}_0)\left(\gamma(m)\boldsymbol{r}\right)\\
&= V^{\pi,m}_{\mu}(\psi(h_{1:md})).
\end{aligned}$$

Then for $t < m$ we have we can work backwards from m,

$$
\begin{aligned}
\widehat{V}^{\pi,m}_{\widehat{\mu}}(h_{1:td}) &= \sum_{\boldsymbol{a}\in\mathcal{A}^d}\sum_{o\in\mathcal{O},\boldsymbol{r}\in\mathcal{R}}\prod_{k=0}^{d-1}\pi(\boldsymbol{a}_k|h_{1:(td+k)})\mu(o\boldsymbol{r}|\psi(h_{1:td})\boldsymbol{a}_0)\left(\gamma(t)\boldsymbol{r}+\widehat{V}^{\pi,m}_{\widehat{\mu}}(h_{1:td}\overline{\boldsymbol{a}o\boldsymbol{r}})\right)\\
&= \sum_{\boldsymbol{a}\in\mathcal{A}^d}\sum_{o\in\mathcal{O},\boldsymbol{r}\in\mathcal{R}}\prod_{k=0}^{d-1}\pi(\boldsymbol{a}_k|h_{1:(td+k)})\mu(o\boldsymbol{r}|\psi(h_{1:td})\boldsymbol{a}_0)\left(\gamma(t)\boldsymbol{r}+V^{\pi,m}_{\mu}(\psi(h_{1:td}\overline{\boldsymbol{a}o\boldsymbol{r}}))\right)\\
&= \sum_{\boldsymbol{a}\in\mathcal{A}^d}\sum_{o\in\mathcal{O},\boldsymbol{r}\in\mathcal{R}}\prod_{k=0}^{d-1}\pi(\boldsymbol{a}_k|h_{1:(td+k)})\mu(o\boldsymbol{r}|\psi(h_{1:td})\boldsymbol{a}_0)\left(\gamma(t)\boldsymbol{r}+V^{\pi,m}_{\mu}(\psi(h_{1:td})\boldsymbol{a}_0o\boldsymbol{r})\right)\\
&= \sum_{\boldsymbol{a}_0\in\mathcal{A}}\sum_{o\in\mathcal{O},\boldsymbol{r}\in\mathcal{R}}\pi(\boldsymbol{a}_0|h_{1:td})\mu(o\boldsymbol{r}|\psi(h_{1:td})\boldsymbol{a}_0)\left(\gamma(t)\boldsymbol{r}+V^{\pi,m}_{\mu}(\psi(h_{1:td})\boldsymbol{a}_0o\boldsymbol{r})\right)\\
&= \sum_{\boldsymbol{a}_0\in\mathcal{A}}\pi(\boldsymbol{a}_0|h_{1:td})\sum_{o\in\mathcal{O},\boldsymbol{r}\in\mathcal{R}}\mu(o\boldsymbol{r}|\psi(h_{1:td})\boldsymbol{a}_0)\left(\gamma(t)\boldsymbol{r}+V^{\pi,m}_{\mu}(\psi(h_{1:td})\boldsymbol{a}_0o\boldsymbol{r})\right)\\
&= \sum_{\boldsymbol{a}_0\in\mathcal{A}}\pi(\boldsymbol{a}_0|\psi(h_{1:td}))\sum_{o\in\mathcal{O},\boldsymbol{r}\in\mathcal{R}}\mu(o\boldsymbol{r}|\psi(h_{1:td})\boldsymbol{a}_0)\left(\gamma(t)\boldsymbol{r}+V^{\pi,m}_{\mu}(\psi(h_{1:td})\boldsymbol{a}_0o\boldsymbol{r})\right)\\
&= V^{\pi,m}_{\mu}(\psi(h_{1:td})).
\end{aligned}
$$

■

A.6 On action binarisation

An effective binarisation of action was proposed by [MH21]. In this section we build on the results by extending the action binarisation to the Bayesian reinforcement learning case.

To start with, [MH21] use a bijective decoder function $D : \{0,1\}^d \to \mathcal{A}$ and an encoder function $C : \mathcal{A} \to \{0,1\}^d$ to decode and encode binary sequences to actions. For the sake of notational simplicity, for this section we will assume $\mathcal{A} = \{0,1\}^d$ since we are not specifically concerned with what the original action space looks like, but splitting it up to be a binary action space over d timesteps.

In this binarised action setup, the binary action history space is defined as follows:

$$\breve{\mathcal{H}} := \bigcup_{t=1}^{\infty}(\mathcal{O}\times\mathcal{R}\times\{0,1\})^{(t-1)}\times\mathcal{O}\times\mathcal{R}\,.$$

Now we can consider a recursive history translation function which takes regular histories and translates them into binary action histories.

Definition 19 ([MH21]) *The history transformation function is expressed with $g : \mathcal{H} \to \breve{\mathcal{H}}$. The map is recursively defined for any history h, action a, next observation o' and next reward r' as*

$$g(hao'r') := g(h)\boldsymbol{a}_1o_\perp r_\perp\boldsymbol{a}_2o_\perp r_\perp\ldots\boldsymbol{a}_do'r' \text{ and } g(e) := e$$

where $\boldsymbol{a} \in \mathcal{A} = \{0,1\}^d$, $o_\perp$ is a "blank" observation of the history, e denotes the "initial" history, and $r_\perp$ is any fixed real-value. We assume $r_\perp = 0 \in \mathcal{R}$.

For shorthand we will use $\overline{\boldsymbol{a}o\boldsymbol{r}_\perp}_{<i}$ to represent $\boldsymbol{a}_1or_\perp\ldots\boldsymbol{a}_{i-1}or_\perp$. Now that we have a method to convert regular histories to binary action histories, we can construct the binary action equivalent of an arbitrary environment μ.

Definition 20 ([MH21]) *For any action $a \in \{0,1\}$, sequentialised history $\tau \in \breve{\mathcal{H}}$, and any partial extension $\overline{\boldsymbol{aor}_{\perp}}_{<i}$ for $i < d$, the probability of receiving o' and r' as the next observation and reward is as follows:*

$$\breve{\mu}(o'r'|\tau\overline{\boldsymbol{aor}_{\perp}}_{<i}a) := \begin{cases} \mu(o'r'|ha) & \text{if } \tau\overline{\boldsymbol{aor}_{\perp}}_{<i}ao'r' = g(hao'r') \\ 1 & \text{if } o'r' = o_{\perp}r_{\perp} \\ & \wedge\, g^{-1}(\tau\overline{\boldsymbol{aor}_{\perp}}_{<i}ao'r') = \perp \\ 0 & \text{otherwise} \end{cases}$$

where μ is the original environment.

Then to show that $\breve{\mu}$ is indeed the binary action equivalent of μ, the following equivalence result was proven.

Theorem 21 ([MH21]) *For any $o' \in \mathcal{O}$, $r' \in \mathcal{R}$, $h \in \mathcal{H}$, and $\boldsymbol{a} \in \mathcal{A}$, the following holds between $\breve{\mu}$ and μ*

$$\breve{\mu}(o'r'|g(h)\overline{\boldsymbol{aor}_{\perp}}_{<d}\boldsymbol{a}_d) = \mu(o'r'|h\boldsymbol{a}) \tag{4}$$

Extending on this result we have a corollary relating the binarised action environment and original environment, which we will use in a later proof.

Corollary 22

$$\breve{\mu}(g(h_{<t})) = \mu(h_{<t}).$$

Proof

$$\begin{aligned} \breve{\mu}(g(h_{<t})) &\overset{(a)}{=} \breve{\mu}(g(h_{<(t-1)})\boldsymbol{a}_1 o r_{\perp}\boldsymbol{a}_2 o r_{\perp} \ldots \boldsymbol{a}_d o_{t-1+d} r_{t-1+d}) \\ &= \prod_{i=1}^{t-1}\prod_{k=0}^{d} \breve{\mu}(o_{i+k}r_{i+k}|g(h_{<i})\overline{\boldsymbol{aor}_{\perp}}_{<k}\boldsymbol{a}_k) \\ &\overset{(b)}{=} \prod_{i=1}^{t-1} \breve{\mu}(o_i r_i|g(h_{<i})\overline{\boldsymbol{aor}_{\perp}}_{<d}\boldsymbol{a}_d) \\ &\overset{(c)}{=} \prod_{i=1}^{t-1} \mu(o_i r_i|h_{<i}a_i) \\ &= \mu(h_{<t}). \end{aligned}$$

(a) comes from the definition of g. (b) comes from Definition 20, where it will always be 1 on histories of that form. (c) comes from Theorem 21. ■

It was additionally shown that in the binarised action setting, a policy which performs well will still perform well in the original setting.

Theorem 23 ([MH21]) *Any $\gamma^{(d-1)/d}\cdot\varepsilon$-optimal policy of the binarised action environment is ε-optimal in the original environment.*

Now considering the Bayesian agent, we are interested in knowing if the learning component of the agent, ξ, will still learn as well in the binary action environment. To this end, let $\breve{\mathcal{M}} := \{\mu \in \mathcal{M} \,|\, \mathcal{A} = \{0,1\}\}$ be the class of environments with binary action space. We can show that $\xi_{\breve{\mathcal{M}}}$ indeed all semimeasures in $\mathcal{M}$, that is, it dominates all environments in $\mathcal{M}$.

Theorem 24 *$\xi_{\breve{\mathcal{M}}}$ dominates all semimeasures in $\mathcal{M}$.*

Proof Let $\rho \in \mathcal{M}$. We want to show there exists a constant c such that for all h we have $\xi_{\breve{\mathcal{M}}}(g(h)) > c\rho(h)$.

$$\begin{aligned}\xi_{\breve{\mathcal{M}}}(g(h)) &= \sum_{\mu \in \breve{\mathcal{M}}} w(\mu)\mu(g(h)) \\ &> w(\breve{\rho})\breve{\rho}(g(h)) \\ &= c\rho(h)\end{aligned}$$

Since $\breve{\rho}$ is an environment that only takes binary actions, this completes the proof. ■

We can alternatively consider when Definition 20 is applied to the original mixture $\xi_{\mathcal{M}}$. In this case we also see that $\breve{\xi}$ dominates all semimeasure in $\mathcal{M}$.

Theorem 25 *$\breve{\xi}$ dominates all semimeasures in $\mathcal{M}$.*

Proof This comes straight from Theorem 21 which gives us $\breve{\xi}(g(h)) = \xi_{\mathcal{M}}(h)$ ■