Asymptotic Learnability of Reinforcement Problems with Arbitrary Dependence*

Daniil Ryabko and Marcus Hutter

IDSIA, Galleria 2, CH-6928 Manno-Lugano, Switzerland
{daniil, marcus}@idsia.ch
http://www.idsia.ch/~{daniil, marcus}

Abstract. We address the problem of reinforcement learning in which observations may exhibit an arbitrary form of stochastic dependence on past observations and actions, i.e. environments more general than (PO) MDPs. The task for an agent is to attain the best possible asymptotic reward where the true generating environment is unknown but belongs to a known countable family of environments. We find some sufficient conditions on the class of environments under which an agent exists which attains the best asymptotic reward for any environment in the class. We analyze how tight these conditions are and how they relate to different probabilistic assumptions known in reinforcement learning and related fields, such as Markov Decision Processes and mixing conditions.

1 Introduction

Many real-world "learning" problems (like learning to drive a car or playing a game) can be modelled as an agent π that interacts with an environment μ and is (occasionally) rewarded for its behavior. We are interested in agents which perform well in the sense of having high long-term reward, also called the value $V(\mu,\pi)$ of agent π in environment μ. If μ is known, it is a pure (non-learning) computational problem to determine the optimal agent $\pi^\mu := \text{argmax}_\pi V(\mu,\pi)$. It is far less clear what an "optimal" agent means, if μ is unknown. A reasonable objective is to have a single policy π with high value simultaneously in many environments. We will formalize and call this criterion *self-optimizing* later.

Learning approaches in reactive worlds. Reinforcement learning, sequential decision theory, adaptive control theory, and active expert advice, are theories dealing with this problem. They overlap but have different core focus: Reinforcement learning algorithms [SB98] are developed to learn μ or directly its value. Temporal difference learning is computationally very efficient, but has slow asymptotic guarantees (only) in (effectively) small observable MDPs. Others have faster guarantee in finite state MDPs [BT99]. There are algorithms [EDKM05] which are optimal for any finite connected POMDP, and this is apparently the largest class of environments considered. In sequential decision theory, a Bayes-optimal agent π^* that maximizes $V(\xi,\pi)$ is considered, where ξ is

* This work was supported by the Swiss NSF grant 200020-107616.

J.L. Balcázar, P.M. Long, and F. Stephan (Eds.): ALT 2006, LNAI 4264, pp. 334–347, 2006.

a mixture of environments $\nu \in \mathcal{C}$ and $\mathcal{C}$ is a class of environments that contains the true environment $\mu \in \mathcal{C}$ [Hut05]. Policy π^* is self-optimizing in an arbitrary (e.g. non-POMDP) class $\mathcal{C}$, provided $\mathcal{C}$ allows for self-optimizingness [Hut02]. Adaptive control theory [KV86] considers very simple (from an AI perspective) or special systems (e.g. linear with quadratic loss function), which sometimes allow computationally and data efficient solutions. Action with expert advice [dFM04, PH05, PH06, CBL06] constructs an agent (called master) that performs nearly as well as the best agent (best expert in hindsight) from some class of experts, in *any* environment ν. The important special case of passive sequence prediction in arbitrary unknown environments, where the actions=predictions do not affect the environment is comparably easy [Hut03, HP04].

The difficulty in active learning problems can be identified (at least, for countable classes) with *traps* in the environments. Initially the agent does not know μ, so has asymptotically to be forgiven in taking initial "wrong" actions. A well-studied such class are ergodic MDPs which guarantee that, from any action history, every state can be (re)visited [Hut02].

What's new. The aim of this paper is to characterize as general as possible classes $\mathcal{C}$ in which self-optimizing behaviour is possible, more general than POMDPs. To do this we need to characterize classes of environments that forgive. For instance, exact state recovery is unnecessarily strong; it is sufficient being able to recover high rewards, from whatever states. Further, in many real world problems there is no information available about the "states" of the environment (e.g. in POMDPs) or the environment may exhibit long history dependencies.

Rather than trying to model an environment (e.g. by MDP) we try to identify the conditions sufficient for learning. Towards this aim, we propose to consider only environments in which, after any arbitrary finite sequence of actions, the best value is still achievable. The performance criterion here is asymptotic average reward. Thus we consider such environments for which there exists a policy whose asymptotic average reward exists and upper-bounds asymptotic average reward of any other policy. Moreover, the same property should hold after any finite sequence of actions has been taken (no traps).

Yet this property in itself is not sufficient for identifying optimal behavior. We require further that, from any sequence of k actions, it is possible to return to the optimal level of reward in $o(k)$ steps. (The above conditions will be formulated in a probabilistic form.) Environments which possess this property are called *value-stable.*

We show that for any countable class of value-stable environments there exists a policy which achieves best possible value in any of the environments from the class (i.e. is *self-optimizing* for this class). We also show that strong value-stability is in a certain sense necessary.

We also consider examples of environments which possess strong value-stability. In particular, any ergodic MDP can be easily shown to have this property. A mixing-type condition which implies value-stability is also demonstrated. Finally, we provide a construction allowing to build examples of value-stable environments which are not isomorphic to a finite POMDP, thus demonstrating that the class of value-stable environments is quite general.

It is important in our argument that the class of environments for which we seek a self-optimizing policy is countable, although the class of all value-stable environments is uncountable. To find a set of conditions necessary and sufficient for learning which do not rely on countability of the class is yet an open problem. However, from a computational perspective countable classes are sufficiently large (e.g. the class of all computable probability measures is countable).

Contents. The paper is organized as follows. Section 2 introduces necessary notation of the agent framework. In Section 3 we define and explain the notion of value-stability, which is central in the paper. Section 4 presents the theorem about self-optimizing policies for classes of value-stable environments, and illustrates the applicability of the theorem by providing examples of strongly value-stable environments. In Section 5 we discuss necessity of the conditions of the main theorem. Section 6 provides some discussion of the results and an outlook to future research. The formal proof of the main theorem is given in the appendix, while Section 4 contains only intuitive explanations.

2 Notation and Definitions

We essentially follow the notation of [Hut02, Hut05].

Strings and probabilities. We use letters $i,k,l,m,n\in I\!N$ for natural numbers, and denote the cardinality of sets $\mathcal{S}$ by $\#\mathcal{S}$. We write $\mathcal{X}^*$ for the set of finite strings over some alphabet $\mathcal{X}$, and $\mathcal{X}^\infty$ for the set of infinite sequences. For a string $x\in\mathcal{X}^*$ of length $\ell(x)=n$ we write $x_1x_2...x_n$ with $x_t\in\mathcal{X}$ and further abbreviate $x_{k:n}:=x_kx_{k+1}...x_{n-1}x_n$ and $x_{<n}:=x_1...x_{n-1}$. Finally, we define $x_{k..n}:=x_k+...+x_n$, provided elements of $\mathcal{X}$ can be added.

We assume that sequence $\omega=\omega_{1:\infty}\in\mathcal{X}^\infty$ is sampled from the "true" probability measure μ, i.e. $\mathbf{P}[\omega_{1:n}=x_{1:n}]=\mu(x_{1:n})$. We denote expectations w.r.t. μ by $\mathbf{E}$, i.e. for a function $f:\mathcal{X}^n\to I\!R$, $\mathbf{E}[f]=\mathbf{E}[f(\omega_{1:n})]=\sum_{x_{1:n}}\mu(x_{1:n})f(x_{1:n})$. When we use probabilities and expectations with respect to other measures we make the notation explicit, e.g. $\mathbf{E}_\nu$ is the expectation with respect to ν. Measures ν_1 and ν_2 are called *singular* if there exists a set A such that $\nu_1(A)=0$ and $\nu_2(A)=1$.

The agent framework is general enough to allow modelling nearly any kind of (intelligent) system [RN95]. In cycle k, an agent performs *action* $y_k\in\mathcal{Y}$ (output) which results in *observation* $o_k\in\mathcal{O}$ and *reward* $r_k\in\mathcal{R}$, followed by cycle $k+1$ and so on. We assume that the action space $\mathcal{Y}$, the observation space $\mathcal{O}$, and the reward space $\mathcal{R}\subset I\!R$ are finite, w.l.g. $\mathcal{R}=\{0,...,r_{max}\}$. We abbreviate $z_k:=y_kr_ko_k\in\mathcal{Z}:=\mathcal{Y}\times\mathcal{R}\times\mathcal{O}$ and $x_k=r_ko_k\in\mathcal{X}:=\mathcal{R}\times\mathcal{O}$. An agent is identified with a (probabilistic) *policy* π. Given *history* $z_{<k}$, the probability that agent π acts y_k in cycle k is (by definition) $\pi(y_k|z_{<k})$. Thereafter, *environment* μ provides (probabilistic) reward r_k and observation o_k, i.e. the probability that the agent perceives x_k is (by definition) $\mu(x_k|z_{<k}y_k)$. Note that policy and environment are allowed to depend on the complete history. We do not make any MDP

or POMDP assumption here, and we don't talk about states of the environment, only about observations. Each (policy,environment) pair (π,μ) generates an I/O sequence $z_1^{\pi\mu}z_2^{\pi\mu}....$ Mathematically, history $z_{1:k}^{\pi\mu}$ is a random variable with probability

$$\mathbf{P}[z_{1:k}^{\pi\mu}=z_{1:k}] \;=\; \pi(y_1)\cdot\mu(x_1|y_1)\cdot...\cdot\pi(y_k|z_{<k})\cdot\mu(x_k|z_{<k}y_k)$$

Since value maximizing policies can always be chosen deterministic, there is no real need to consider probabilistic policies, and henceforth we consider deterministic policies p. We assume that $\mu\in\mathcal{C}$ is the true, but unknown, environment, and $\nu\in\mathcal{C}$ a generic environment.

3 Setup

For an environment ν and a policy p define random variables (lower and upper average value)

$$\overline{V}(\nu,p) \;:=\; \limsup_m \left\{\tfrac{1}{m}r_{1..m}^{p\nu}\right\} \quad \text{and} \quad \underline{V}(\nu,p) \;:=\; \liminf_m \left\{\tfrac{1}{m}r_{1..m}^{p\nu}\right\}$$

where $r_{1..m}:=r_1+...+r_m$. If there exists a constant V such that

$$\overline{V}(\nu,p) \;=\; \underline{V}(\nu,p) \;=\; V \text{ a.s.}$$

then we say that the limiting average value exists and denote it by $V(\nu,p)=:V$.

An environment ν is *explorable* if there exists a policy p_ν such that $V(\nu,p_\nu)$ exists and $\overline{V}(\nu,p)\leq V(\nu,p_\nu)$ with probability 1 for every policy p. In this case define $V_\nu^*:=V(\nu,p_\nu)$.

A policy p is *self-optimizing* for a set of environments $\mathcal{C}$ if $V(\nu,p)=V_\nu^*$ for every $\nu\in\mathcal{C}$.

Definition 1 (value-stable environments). *An explorable environment ν is* (strongly) value-stable *if there exist a sequence of numbers $r_i^\nu\in[0,r_{max}]$ and two functions $d_\nu(k,\varepsilon)$ and $\varphi_\nu(n,\varepsilon)$ such that $\frac{1}{n}r_{1..n}^\nu\to V_\nu^*$, $d_\nu(k,\varepsilon)=o(k)$, $\sum_{n=1}^\infty\varphi_\nu(n,\varepsilon)<\infty$ for every fixed ε, and for every k and every history $z_{<k}$ there exists a policy $p=p_\nu^{z_{<k}}$ such that*

$$\mathbf{P}\left(r_{k..k+n}^\nu - r_{k..k+n}^{p\nu} > d_\nu(k,\varepsilon)+n\varepsilon \mid z_{<k}\right)\leq\varphi_\nu(n,\varepsilon). \tag{1}$$

First of all, this condition means that the strong law of large numbers for rewards holds uniformly over histories $z_{<k}$; the numbers r_i^ν here can be thought of as expected rewards of an optimal policy. Furthermore, the environment is "forgiving" in the following sense: from any (bad) sequence of k actions it is possible (knowing the environment) to recover up to $o(k)$ reward loss; to recover means to reach the level of reward obtained by the optimal policy which from the beginning was taking only optimal actions. That is, suppose that a person A has made k possibly suboptimal actions and after that "realized" what the true environment was and how to act optimally in it. Suppose that a person B was

from the beginning taking only optimal actions. We want to compare the performance of A and B on first n steps after the step k. An environment is strongly value stable if A can catch up with B except for $o(k)$ gain. The numbers r_i^ν can be thought of as expected rewards of B; A can catch up with B up to the reward loss $d_\nu(k,\varepsilon)$ with probability $\varphi_\nu(n,\varepsilon)$, where the latter does not depend on past actions and observations (the law of large numbers holds uniformly).

In the next section after presenting the main theorem we consider examples of families of strongly-values stable environments.

4 Main Results

In this section we present the main self-optimizingness result along with an informal explanation of its proof, and illustrate the applicability of this result with examples of classes of value-stable environments.

Theorem 2 (value-stable⇒self-optimizing). *For any countable class $\mathcal{C}$ of strongly value-stable environments, there exists a policy which is self-optimizing for $\mathcal{C}$.*

A formal proof is given in the appendix; here we give some intuitive justification. Suppose that all environments in $\mathcal{C}$ are deterministic. We will construct a self-optimizing policy p as follows: Let ν^t be the first environment in $\mathcal{C}$. The algorithm assumes that the true environment is ν^t and tries to get ε-close to its optimal value for some (small) ε. This is called an exploitation part. If it succeeds, it does some exploration as follows. It picks the first environment ν^e which has higher average asymptotic value than ν^t ($V^*_{\nu^e} > V^*_{\nu^t}$) and tries to get ε-close to this value acting optimally under ν^e. If it can not get close to the ν^e-optimal value then ν^e is not the true environment, and the next environment can be picked for exploration (here we call "exploration" successive attempts to exploit an environment which differs from the current hypothesis about the true environment and has a higher average reward). If it can, then it switches to exploitation of ν^t, exploits it until it is ε'-close to $V^*_{\nu^t}$, $\varepsilon' < \varepsilon$ and switches to ν^e again this time trying to get ε'-close to V_{ν^e}; and so on. This can happen only a finite number of times if the true environment is ν^t, since $V^*_{\nu^t} < V^*_{\nu^e}$. Thus after exploration either ν^t or ν^e is found to be inconsistent with the current history. If it is ν^e then just the next environment ν^e such that $V^*_{\nu^e} > V^*_{\nu^t}$ is picked for exploration. If it is ν^t then the first consistent environment is picked for exploitation (and denoted ν^t). This in turn can happen only a finite number of times before the true environment ν is picked as ν^t. After this, the algorithm still continues its exploration attempts, but can always keep within $\varepsilon_k \to 0$ of the optimal value. This is ensured by $d(k) = o(k)$.

The probabilistic case is somewhat more complicated since we can not say whether an environment is "consistent" with the current history. Instead we test each environment for consistency as follows. Let ξ be a mixture of all environments in $\mathcal{C}$. Observe that together with some fixed policy each environment μ can be considered as a measure on $\mathcal{Z}^\infty$. Moreover, it can be shown that (for any fixed policy) the ratio $\frac{\nu(z_{<n})}{\xi(z_{<n})}$ is bounded away from zero if ν is the true environment μ and tends to zero if ν is singular with μ (in fact, here singularity is a probabilistic analogue of inconsistency). The exploration part of the algorithm ensures that at least one of the environments ν^t and ν^e is singular with ν on the current history, and a succession of tests $\frac{\nu(z_{<n})}{\xi(z_{<n})} \geq \alpha_s$ with $\alpha_s \to 0$ is used to exclude such environments from consideration.

The next proposition provides some conditions on mixing rates which are sufficient for value-stability; we do not intend to provide sharp conditions on mixing rates but rather to illustrate the relation of value-stability with mixing conditions.

We say that a stochastic process h_k, $k \in I\!N$ satisfies strong α-mixing conditions with coefficients $\alpha(k)$ if (see e.g. [Bos96])

$$\sup_{n\in I\!N} \sup_{B\in\sigma(h_1,\dots,h_n), C\in\sigma(h_{n+k},\dots)} |\,\mathbf{P}(B\cap C) - \mathbf{P}(B)\,\mathbf{P}(C)| \leq \alpha(k),$$

where $\sigma()$ stands for the sigma-algebra generated by the random variables in brackets. Loosely speaking, mixing coefficients α reflect the speed with which the process "forgets" about its past.

Proposition 3 (mixing conditions). *Suppose that an explorable environment ν is such that there exist a sequence of numbers r_i^ν and a function $d(k)$ such that $\frac{1}{n} r_{1..n}^\nu \to V_\nu^*$, $d(k) = o(k)$, and for each $z_{<k}$ there exists a policy p such that the sequence $r_i^{p\nu}$ satisfies strong α-mixing conditions with coefficients $\alpha(k) = \frac{1}{k^{1+\varepsilon}}$ for some $\varepsilon > 0$ and*

$$r_{k..k+n}^\nu - \mathbf{E}\left(r_{k..k+n}^{p\nu} \mid z_{<k}\right) \leq d(k)$$

for any n. Then ν is value-stable.

Proof. Using the union bound we obtain

$$\begin{aligned}&\mathbf{P}\left(r_{k..k+n}^\nu - r_{k..k+n}^{p\nu} > d(k) + n\varepsilon\right)\\ &\leq I\left(r_{k..k+n}^\nu - \mathbf{E}\, r_{k..k+n}^{p\nu} > d(k)\right) + \mathbf{P}\left(\left|r_{k..k+n}^{p\nu} - \mathbf{E}\, r_{k..k+n}^{p\nu}\right| > n\varepsilon\right).\end{aligned}$$

The first term equals 0 by assumption and the second term for each ε can be shown to be summable using [Bos96, Thm.1.3]: For a sequence of uniformly bounded zero-mean random variables r_i satisfying strong α-mixing conditions the following bound holds true for any integer $q \in [1, n/2]$:

$$\mathbf{P}\left(|r_{1..n}| > n\varepsilon\right) \leq c e^{-\varepsilon^2 q/c} + cq\alpha\left(\frac{n}{2q}\right)$$

for some constant c; in our case we just set $q = n^{\frac{\varepsilon}{2+\varepsilon}}$. □

(PO)MDPs. Applicability of Theorem 2 and Proposition 3 can be illustrated on (PO)MDPs. We note that self-optimizing policies for (uncountable) classes of finite ergodic MDPs and POMDPs are known [BT99, EDKM05]; the aim of the present section is to show that value-stability is a weaker requirement than the requirements of these models, and also to illustrate applicability of our results.

We call μ a (stationary) *Markov decision process* (MDP) if the probability of perceiving $x_k \in \mathcal{X}$, given history $z_{<k}y_k$ only depends on $y_k \in \mathcal{Y}$ and x_{k-1}. In this case $x_k \in \mathcal{X}$ is called a *state*, $\mathcal{X}$ the *state space*. An MDP μ is called *ergodic* if there exists a policy under which every state is visited infinitely often with probability 1. An MDP with a stationary policy forms a Markov chain.

An environment is called a (finite) *partially observable MDP* (POMDP) if there is a sequence of random variables s_k taking values in a finite space $\mathcal{S}$ called the state space, such that x_k depends only on s_k and y_k, and s_{k+1} is independent of $s_{<k}$ given s_k. Abusing notation the sequence $s_{1:k}$ is called the underlying Markov chain. A POMDP is called *ergodic* if there exists a policy such that the underlying Markov chain visits each state infinitely often with probability 1.

In particular, any ergodic POMDP ν satisfies strong α-mixing conditions with coefficients decaying exponentially fast in case there is a set $H \subset \mathcal{R}$ such that $\nu(r_i \in H) = 1$ and $\nu(r_i = r|s_i = s, y_i = y) \neq 0$ for each $y \in \mathcal{Y}, s \in \mathcal{S}, r \in H, i \in I\!N$. Thus for any such POMDP ν we can use Proposition 3 with $d(k,\varepsilon)$ a constant function to show that ν is strongly value-stable:

Corollary 4 (POMDP⇒value-stable). *Suppose that a POMDP ν is ergodic and there exists a set $H \subset \mathcal{R}$ such that $\nu(r_i \in H) = 1$ and $\nu(r_i = r|s_i = s, y_i = y) \neq 0$ for each $y \in \mathcal{Y}, h \in \mathcal{S}, r \in H$, where $\mathcal{S}$ is the finite state space of the underlying Markov chain. Then ν is strongly value-stable.*

However, it is illustrative to obtain this result for MDPs directly, and in a slightly stronger form.

Proposition 5 (MDP⇒value-stable). *Any finite-state ergodic MDP ν is a strongly value-stable environment.*

Proof. Let $d(k,\varepsilon) = 0$. Denote by μ the true environment, let $z_{<k}$ be the current history and let the current state (the observation x_k) of the environment be $a \in \mathcal{X}$, where $\mathcal{X}$ is the set of all possible states. Observe that for an MDP there is an optimal policy which depends only on the current state. Moreover, such a policy is optimal for any history. Let p_μ be such a policy. Let r_i^μ be the expected reward of p_μ on step i. Let $l(a,b) = \min\{n : x_{k+n} = b|x_k = a\}$. By ergodicity of μ there exists a policy p for which $\mathbf{E}l(b,a)$ is finite (and does not depend on k). A policy p needs to get from the state b to one of the states visited by an optimal policy, and then acts according to p_μ. Let $f(n) := \frac{n r_{\max}}{\log n}$. We have

$$
\begin{aligned}
&\mathbf{P}\left(\left|r^\mu_{k..k+n} - r^{p\mu}_{k..k+n}\right| > n\varepsilon\right) \le \sup_{a\in\mathcal{X}} \mathbf{P}\left(\left|\mathbf{E}\left(r^{p_\mu\mu}_{k..k+n}|x_k = a\right) - r^{p\mu}_{k..k+n}\right| > n\varepsilon)\right) \\
&\le \sup_{a,b\in\mathcal{X}} \mathbf{P}(l(a,b) > f(n)/r_{\max}) \\
&+ \sup_{a,b\in\mathcal{X}} \mathbf{P}\left(\left|\mathbf{E}\left(r^{p_\mu\mu}_{k..k+n}|x_k = a\right) - r^{p_\mu\mu}_{k+f(n)..k+n}\right| > n\varepsilon - f(n)\middle| x_{k+f(n)} = a\right) \\
&\le \sup_{a,b\in\mathcal{X}} \mathbf{P}(l(a,b) > f(n)/r_{\max}) \\
&+ \sup_{a\in\mathcal{X}} \mathbf{P}\left(\left|\mathbf{E}\left(r^{p_\mu\mu}_{k..k+n}|x_k = a\right) - r^{p_\mu\mu}_{k..k+n}\right| > n\varepsilon - 2f(n)\middle| x_k = a\right).
\end{aligned}
$$

In the last term we have the deviation of the reward attained by the optimal policy from its expectation. Clearly, both terms are bounded exponentially in n. □

In the examples above the function $d(k,\varepsilon)$ is a constant and $\varphi(n,\varepsilon)$ decays exponentially fast. This suggests that the class of value-stable environments stretches beyond finite (PO)MDPs. We illustrate this guess by the construction that follows.

An example of a value-stable environment: Infinitely armed bandit. Next we present a construction of environments which can not be modelled as finite POMDPs but are value-stable. Consider the following environment ν. There is a countable family $\mathcal{C}' = \{\zeta_i : i \in I\!N\}$ of *arms*, that is, sources generating i.i.d. rewards 0 and 1 (and, say, empty observations) with some probability δ_i of the reward being 1. The action space $\mathcal{Y}$ consists of three actions $\mathcal{Y}=\{g,u,d\}$. To get the next reward from the current arm ζ_i an agent can use the action g. At the beginning the current arm is ζ_0 and then the agent can move between arms as follows: it can move one arm "up" using the action u or move "down" to the first environment using the action d. The reward for actions u and d is 0.

Clearly, ν is a POMDP with countably infinite number of states in the underlying Markov chain, which (in general) is not isomorphic to a finite POMDP.

Claim. The environment ν just constructed is value-stable.

Proof. Let $\delta = \sup_{i \in I\!N} \delta_i$. Clearly, $\overline{V}(\nu,p') \leq \delta$ with probability 1 for any policy p' . A policy p which, knowing all the probabilities δ_i, achieves $\overline{V}(\nu,p)=\underline{V}(\nu,p)=\delta=:V_\nu^*$ a.s., can be easily constructed. Indeed, find a sequence ζ'_j, $j \in I\!N$, where for each j there is $i=:i_j$ such that $\zeta'_j=\zeta_i$, satisfying $\lim_{j\to\infty}\delta_{i_j}=\delta$. The policy p should carefully exploit one by one the arms ζ_j, staying with each arm long enough to ensure that the average reward is close to the expected reward with ε_j probability, where ε_j quickly tends to 0, and so that switching between arms has a negligible impact on the average reward. Thus ν can be shown to be explorable. Moreover, a policy p just sketched can be made independent on (observation and) rewards.

Furthermore, one can modify the policy p (possibly allowing it to exploit each arm longer) so that on each time step t (from some t on) we have $j(t) \leq \sqrt{t}$, where $j(t)$ is the number of the current arm on step t. Thus, after any actions-perceptions history $z_{<k}$ one needs about $\sqrt{k}$ actions (one action u and enough actions d) to catch up with p. So, (1) can be shown to hold with $d(k,\varepsilon)=\sqrt{k}$, r_i the expected reward of p on step i (since p is independent of rewards, $r_i^{p\nu}$ are independent), and the rates $\varphi(n,\varepsilon)$ exponential in n. □

In the above construction we can also allow the action d to bring the agent $d(i)<i$ steps down, where i is the number of the current environment ζ, according to some (possibly randomized) function $d(i)$, thus changing the function $d_\nu(k,\varepsilon)$ and possibly making it non-constant in ε and as close as desirable to linear.

5 Necessity of Value-Stability

Now we turn to the question of how tight the conditions of strong value-stability are. The following proposition shows that the requirement $d(k,\varepsilon)=o(k)$ in (1) can not be relaxed.

Proposition 6 (necessity of $\boldsymbol{d(k,\varepsilon)=o(k)}$). *There exists a countable family of deterministic explorable environments $\mathcal{C}$ such that*

- *for any $\nu\in\mathcal{C}$ for any sequence of actions $y_{<k}$ there exists a policy p such that*

$$r^{\nu}_{n..k+n}=r^{p\nu}_{k..k+n} \text{ for all } n\geq k,$$

where r^{ν}_i are the rewards attained by an optimal policy p_ν (which from the beginning was acting optimally), but

- *for any policy p there exists an environment $\nu\in\mathcal{C}$ such that $\underline{V}(\nu,p)<V^*_\nu$.*

Clearly, each environment from such a class $\mathcal{C}$ satisfies the value stability conditions with $\varphi(n,\varepsilon)\equiv 0$ except $d(k,\varepsilon)=k\neq o(k)$.

Proof. There are two possible actions $y_i\in\{a,b\}$, three possible rewards $r_i\in\{0,1,2\}$ and no observations.

Construct the environment ν_0 as follows: if $y_i=a$ then $r_i=1$ and if $y_i=b$ then $r_i=0$ for any $i\in I\!N$.

For each i let n_i denote the number of actions a taken up to step i: $n_i:=\#\{j\leq i:y_j=a\}$. For each $s>0$ construct the environment ν_s as follows: $r_i(a)=1$ for any i, $r_i(b)=2$ if the longest consecutive sequence of action b taken has length greater than n_i and $n_i\geq s$; otherwise $r_i(b)=0$.

Suppose that there exists a policy p such that $\underline{V}(\nu_i,p)=V^*_{\nu_i}$ for each $i>0$ and let the true environment be ν_0. By assumption, for each s there exists such n that

$$\#\{i\leq n: y_i=b, r_i=0\}\geq s>\#\{i\leq n: y_i=a, r_i=1\}$$

which implies $\underline{V}(\nu_0,p)\leq 1/2<1=V^*_{\nu_0}$. □

It is also easy to show that the *uniformity of convergence in (1)* can not be dropped. That is, if in the definition of value-stability we allow the function $\varphi(n,\varepsilon)$ to depend additionally on the past history $z_{<k}$ then Theorem 2 does not hold. This can be shown with the same example as constructed in the proof of Proposition 6, letting $d(k,\varepsilon)\equiv 0$ but instead allowing $\varphi(n,\varepsilon,z_{<k})$ to take values 0 and 1 according to the number of actions a taken, achieving the same behaviour as in the example provided in the last proof.

Finally, we show that the requirement that the class $\mathcal{C}$ to be learnt is countable can not be easily withdrawn. Indeed, consider the following simple class of environments. An environment is called *passive* if the observations and rewards are independent of actions. Sequence prediction task is a well-studied (and perhaps the only reasonable) class of passive environments: in this task an agent gets the reward 1 if $y_i=o_{i+1}$ and the reward 0 otherwise. Clearly, any *deterministic* passive environment ν is strongly value-stable with $d_\nu(k,\varepsilon)\equiv 1$, $\varphi_\nu(n,\varepsilon)\equiv 0$ and $r_i^\nu = 1$ for all i. Obviously, the class of all deterministic passive environments is not countable. Since for every policy p there is an environment on which p errs exactly on each step,

Claim. The class of all deterministic passive environments can not be learned.

6 Discussion

We have proposed a set of conditions on environments, called value-stability, such that any countable class of value-stable environments admits a self-optimizing policy. It was also shown that these conditions are in a certain sense tight. The class of all value-stable environments includes ergodic MDPs, certain class of finite POMDPs, passive environments, and (provably) other and more environments. So the novel concept of value-stability allows to characterize self-optimizing environment classes, and proving value-stability is typically much easier than proving self-optimizingness directly.

We considered only countable environment classes $\mathcal{C}$. From a computational perspective such classes are sufficiently large (e.g. the class of all computable probability measures is countable). On the other hand, countability excludes continuously parameterized families (like all ergodic MDPs), common in statistical practice. So perhaps the main open problem is to find under which conditions the requirement of countability of the class can be lifted. Ideally, we would like to have some necessary and sufficient conditions such that the class of all environments that satisfy this condition admits a self-optimizing policy.

Another question concerns the uniformity of forgetfulness of the environment. Currently in the definition of value-stability (1) we have the function $\varphi(n,\varepsilon)$ which is the same for all histories $z_{<k}$, that is, both for all actions histories $y_{<k}$ and observations-rewards histories $x_{<k}$. Probably it is possible to differentiate between two types of forgetfulness, one for actions and one for perceptions. In particular, any countable class of passive environments (i.e. such that perceptions are independent of actions) is learnable, suggesting that uniform forgetfulness in perceptions may not be necessary.

References

[Bos96] D. Bosq. *Nonparametric Statistics for Stochastic Processes.* Springer, 1996.

[BT99] R. I. Brafman and M. Tennenholtz. A general polynomial time algorithm for near-optimal reinforcement learning. In *Proc. 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 734–739, 1999.

[CBL06] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006. in preparation.

[CS04] I. Csiszar and P.C. Shields. Notes on information theory and statistics. In *Foundations and Trends in Communications and Information Theory*, 2004.

[Doo53] J. L. Doob. *Stochastic Processes.* John Wiley & Sons, New York, 1953.

[EDKM05] E. Even-Dar, S. M. Kakade, and Y. Mansour. Reinforcement learning in POMDPs without resets. In *IJCAI*, pages 690–695, 2005.

[HP04] M. Hutter and J. Poland. Prediction with expert advice by following the perturbed leader for general weights. In *Proc. 15th International Conf. on Algorithmic Learning Theory (ALT'04)*, volume 3244 of *LNAI*, pages 279–293, Padova, 2004. Springer, Berlin.

[Hut02] M. Hutter. Self-optimizing and Pareto-optimal policies in general environments based on Bayes-mixtures. In *Proc. 15th Annual Conference on Computational Learning Theory (COLT 2002)*, Lecture Notes in Artificial Intelligence, pages 364–379, Sydney, Australia, July 2002. Springer.

[Hut03] M. Hutter. Optimality of universal Bayesian prediction for general loss and alphabet. *Journal of Machine Learning Research*, 4:971–1000, 2003.

[Hut05] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability.* Springer, Berlin, 2005. 300 pages, http://www.idsia.ch/~ marcus/ai/uaibook.htm.

[KV86] P. R. Kumar and P. P. Varaiya. *Stochastic Systems: Estimation, Identification, and Adaptive Control.* Prentice Hall, Englewood Cliffs, NJ, 1986.

[PH05] J. Poland and M. Hutter. Defensive universal learning with experts. In *Proc. 16th International Conf. on Algorithmic Learning Theory (ALT'05)*, volume 3734 of *LNAI*, pages 356–370, Singapore, 2005. Springer, Berlin.

[PH06] J. Poland and M. Hutter. Universal learning of repeated matrix games. In *Conference Benelearn'06 and GTDT workshop at AAMAS'06*, Ghent, 2006.

[dFM04] D. Pucci de Farias and N. Megiddo. How to combine expert (and novice) advice when actions impact the environment? In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16.* MIT Press, Cambridge, MA, 2004.

[RN95] S. J. Russell and P. Norvig. *Artificial Intelligence. A Modern Approach.* Prentice-Hall, Englewood Cliffs, 1995.

[SB98] R. Sutton and A. Barto. *Reinforcement learning: An introduction.* Cambridge, MA, MIT Press, 1998.

A Proof of Theorem 2

A self-optimizing policy p will be constructed as follows. On each step we will have two polices: p^t which exploits and p^e which explores; for each i the policy p either takes an action according to p^t $(p(z_{<i})=p^t(z_{<i}))$ or according to p^e $(p(z_{<i})=p^e(z_{<i}))$, as will be specified below. When the policy p has been defined up to a step k, each environment μ, endowed with this policy, can be considered as a measure on $\mathcal{Z}^k$. We assume this meaning when we use environments as measures on $\mathcal{Z}^k$ (e.g. $\mu(z_{<i})$).

In the algorithm below, i denotes the number of the current step in the sequence of actions-observations. Let $n=1$, $s=1$, and $j^t=j^e=0$. Let also $\alpha_s=2^{-s}$ for $s\in\mathbb{N}$. For each environment ν, find such a sequence of real numbers ε_n^ν that $\varepsilon_n^\nu\to 0$ and $\sum_{n=1}^\infty \varphi_\nu(n,\varepsilon_n^\nu)\leq\infty$.

Let $\imath: I\!N \to \mathcal{C}$ be such a numbering that each $\nu \in \mathcal{C}$ has infinitely many indices.For all $i>1$ define a measure ξ as follows

$$\xi(z_{<i}) = \sum_{\nu\in\mathcal{C}} w_\nu \nu(z_{<i}),$$

where $w_\nu \in \mathcal{R}$ are (any) such numbers that $\sum_\nu w_\nu = 1$ and $w_\nu > 0$ for all $\nu \in \mathcal{C}$.

Define $\boldsymbol{T}$. On each step i let

$$T \;\equiv\; T_i \;:=\; \left\{ \nu \in \mathcal{C} : \frac{\nu(z_{<i})}{\xi(z_{<i})} \geq \alpha_s \right\}$$

Define $\boldsymbol{\nu^t}$. Set ν^t to be the first environment in T with index greater than $\imath(j^t)$. In case this is impossible (that is, if T is empty), increment s, (re)define T and try again. Increment j^t.

Define $\boldsymbol{\nu^e}$. Set ν^e to be the first environment with index greater than $\imath(j^e)$ such that $V^*_{\nu^e} > V^*_{\nu^t}$ and $\nu^e(z_{<k}) > 0$, if such an environment exists. Otherwise proceed one step (according to p^t) and try again. Increment j^e.

Consistency. On each step i (re)define T. If $\nu^t \notin T$, define ν^t, increment s and iterate the infinite loop. (Thus s is incremented only if ν^t is not in T or if T is empty.)

Start the **infinite loop**. Increment n.

Let $\delta := (V^*_{\nu^e} - V^*_{\nu^t})/2$. Let $\varepsilon := \varepsilon^{\nu^t}_n$. If $\varepsilon < \delta$ set $\delta = \varepsilon$. Let $h = j^e$.

Prepare for exploration.

Increment h. The index h is incremented with each next attempt of exploring ν^e. Each attempt will be at least h steps in length.

Let $p^t = p^{y_{<i}}_{\nu^t}$ and set $p = p^t$.

Let i_h be the current step. Find k_1 such that

$$\frac{i_h}{k_1} V^*_{\nu^t} \leq \varepsilon/8 \tag{2}$$

Find $k_2 > 2i_h$ such that for all $m > k_2$

$$\left| \frac{1}{m - i_h} r^{\nu^t}_{i_h+1..m} - V^*_{\nu^t} \right| \leq \varepsilon/8. \tag{3}$$

Find k_3 such that

$$h r_{max}/k_3 < \varepsilon/8. \tag{4}$$

Find k_4 such that for all $m > k_4$

$$\frac{1}{m} d_{\nu^e}(m, \varepsilon/4) \leq \varepsilon/8, \quad \frac{1}{m} d_{\nu^t}(m, \varepsilon/8) \leq \varepsilon/8 \;\text{ and }\; \frac{1}{m} d_{\nu^t}(i_h, \varepsilon/8) \leq \varepsilon/8. \tag{5}$$

Moreover, it is always possible to find such $k > \max\{k_1, k_2, k_3, k_4\}$ that

$$\frac{1}{2k} r^{\nu^e}_{k..3k} \geq \frac{1}{2k} r^{\nu^t}_{k..3k} + \delta. \tag{6}$$

Iterate up to the step k.

Exploration. Set $p^e = p_{\nu^e}^{y_{<n}}$. Iterate h steps according to $p = p^e$. Iterate further until either of the following conditions breaks

(i) $\left|r_{k..i}^{\nu^e} - r_{k..i}^{p\nu}\right| < (i-k)\varepsilon/4 + d_{\nu^e}(k,\varepsilon/4)$,
(ii) $i < 3k$.
(iii) $\nu^e \in T$.

Observe that either (i) or (ii) is necessarily broken.

If on some step ν^t is excluded from T then the infinite loop is iterated. If after exploration ν^e is not in T then redefine ν^e and **iterate the infinite loop**. If both ν^t and ν^e are still in T then **return** to "Prepare for exploration" (otherwise the loop is iterated with either ν^t or ν^e changed).
End of the infinite loop and the algorithm.

Let us show that with probability 1 the "Exploration" part is iterated only a finite number of times in a row with the same ν^t and ν^e.

Suppose the contrary, that is, suppose that (with some non-zero probability) the "Exploration" part is iterated infinitely often while $\nu^t, \nu^e \in T$. Observe that (1) implies that the ν^e-probability that (i) breaks is not greater than φ_{ν_e} $(i-k,\varepsilon/4)$; hence by Borel-Cantelli lemma the event that (i) breaks infinitely often has probability 0 under ν^e.

Suppose that (i) holds almost every time. Then (ii) should be broken except for a finite number of times. We can use (2), (3), (5) and (6) to show that with probability at least $1-\varphi_{\nu^t}(k-i_h,\varepsilon/4)$ under ν^t we have $\frac{1}{3k} r_{1..3k}^{p\nu^t} \geq V_{\nu^t}^* + \varepsilon/2$. Again using Borel-Cantelli lemma and $k > 2i_h$ we obtain that the event that (ii) breaks infinitely often has probability 0 under ν^t.

Thus (at least) one of the environments ν^t and ν^e is singular with respect to the true environment ν given the described policy and current history. Denote this environment by ν'. It is known (see e.g. [CS04, Thm.26]) that if measures μ and ν are mutually singular then $\frac{\mu(x_1,\dots,x_n)}{\nu(x_1,\dots,x_n)} \to \infty$ μ-a.s. Thus

$$\frac{\nu'(z_{<i})}{\nu(z_{<i})} \to 0 \ \nu\text{-a.s.} \tag{7}$$

Observe that (by definition of ξ) $\frac{\nu(z_{<i})}{\xi(z_{<i})}$ is bounded. Hence using (7) we can see that

$$\frac{\nu'(z_{<i})}{\xi(z_{<i})} \to 0 \ \nu\text{-a.s.}$$

Since s and α_s are not changed during the exploration phase this implies that on some step ν' will be excluded from T according to the "consistency" condition, which contradicts the assumption. Thus the "Exploration" part is iterated only a finite number of times in a row with the same ν^t and ν^e.

Observe that s is incremented only a finite number of times since $\frac{\nu'(z_{<i})}{\xi(z_{<i})}$ is bounded away from 0 where ν' is either the true environment ν or any environment from $\mathcal{C}$ which is equivalent to ν on the current history. The latter follows

from the fact that $\frac{\xi(z_{<i})}{\nu(z_{<i})}$ is a submartingale with bounded expectation, and hence, by the submartingale convergence theorem (see e.g. [Doo53]) converges with ν-probability 1.

Let us show that from some step on ν (or an environment equivalent to it) is always in T and selected as ν^t. Consider the environment ν^t on some step i. If $V^*_{\nu^t} > V^*_\nu$ then ν^t will be excluded from T since on any optimal for ν^t sequence of actions (policy) measures ν and ν^t are singular. If $V^*_{\nu^t} < V^*_\nu$ than ν^e will be equal to ν at some point, and, after this happens sufficient number of times, ν^t will be excluded from T by the "exploration" part of the algorithm, s will be decremented and ν will be included into T. Finally, if $V^*_{\nu^t} = V^*_\nu$ then either the optimal value V^*_ν is (asymptotically) attained by the policy p_t of the algorithm, or (if p_{ν^t} is suboptimal for ν) $\frac{1}{i} r^{p\nu^t}_{1..i} < V^*_{\nu^t} - \varepsilon$ infinitely often for some ε, which has probability 0 under ν^t and consequently ν^t is excluded from T.

Thus, the exploration part ensures that all environments not equivalent to ν with indices smaller than $\imath(\nu)$ are removed from T and so from some step on ν^t is equal to (an environment equivalent to) the true environment ν.

We have shown in the "Exploration" part that $n \to \infty$, and so $\varepsilon_n^{\nu^t} \to 0$. Finally, using the same argument as before (Borel-Cantelli lemma, (i) and the definition of k) we can show that in the "exploration" and "prepare for exploration" parts of the algorithm the average value is within $\varepsilon_n^{\nu^t}$ of $V^*_{\nu^t}$ provided the true environment is (equivalent to) ν^t. □