Asymptotic Learnability of Reinforcement Problems with Arbitrary Dependence

Daniil Ryabko, Marcus Hutter

IDSIA, Switzerland

The 17$^{\text{th}}$ International Conference
on Algorithmic Learning Theory
Barcelona, 2006

Reinforcement learning problem

The agent framework is general enough to allow modelling nearly any kind of (intelligent) system.

In cycle k, an agent performs *action* $y_k \in \{0,1\}$ (output) which results in *observation* $x_k \in \mathcal{O}$ and *reward* $r_k \in \mathbb{R}$, followed by cycle $k+1$ and so on.

Cycle k
Environment: *observation* x_k
Agent: *action* y_k
Environment: *reward* r_k

The agent seeks to maximize the cumulative reward.
Observations, rewards and actions may depend on the whole previous history $x_0, y_0, r_0, \ldots, x_{k-1}, y_{k-1}, r_{k-1}$.
If x_k, y_k, r_k depends only on $x_{k-1}, y_{k-1}, r_{k-1}$ then we get a MDP.

The action space $\{0,1\}$, the observation space $\mathcal{O}$, and the reward space $\mathbb{R} \subset \boldsymbol{R}$ are finite.

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) $\mu(x_k|z_{<k}y_k)$.

Sequence prediction generalized

Reinforcement learning generalizes *sequence prediction*:
In cycle k, an agent outputs a *prediction* $y_k \in X$ and perceives the *observation* $x_k \in X$

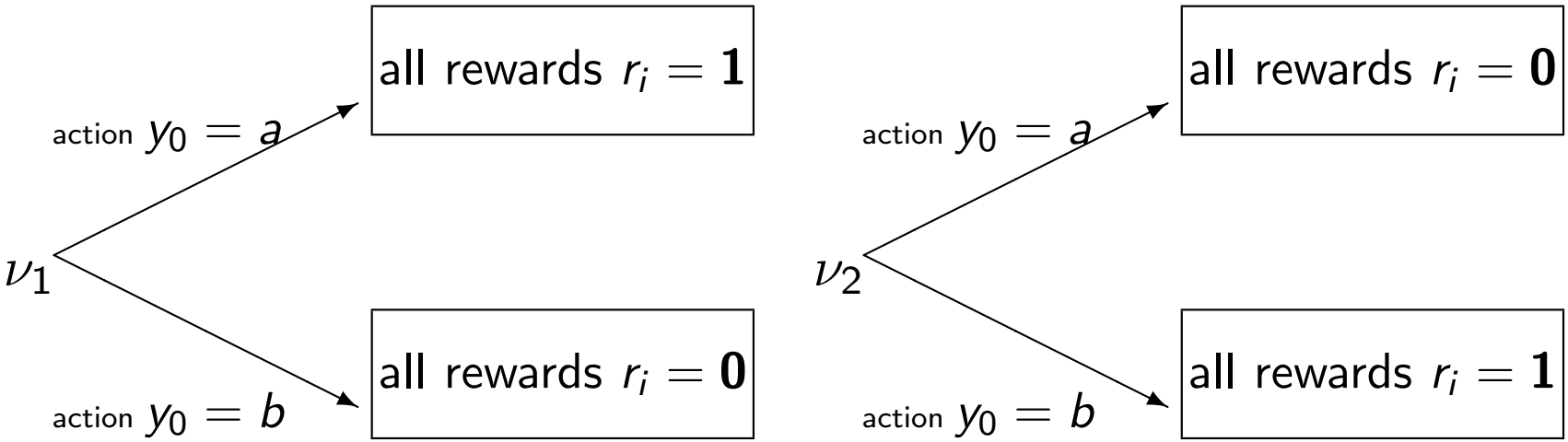
Define the reward $r_k = 1$ if $x_k = y_k$ and $r_k{=}0$ otherwise. The agent seeks to maximize the number of correct predictions. The observations y_k are generated according to some probability distribution independent of predictions.

Theorem (Solomonoff 78)

There exists a policy (a measure) ξ such that for any computable measure μ

$$|\xi(x_k = 1|x_{<k}) - \mu(x_k = 1|x_{<k})| \to 0$$

Does there exist a universal policy for the class of all computable environments? *No.*
Consider an example.



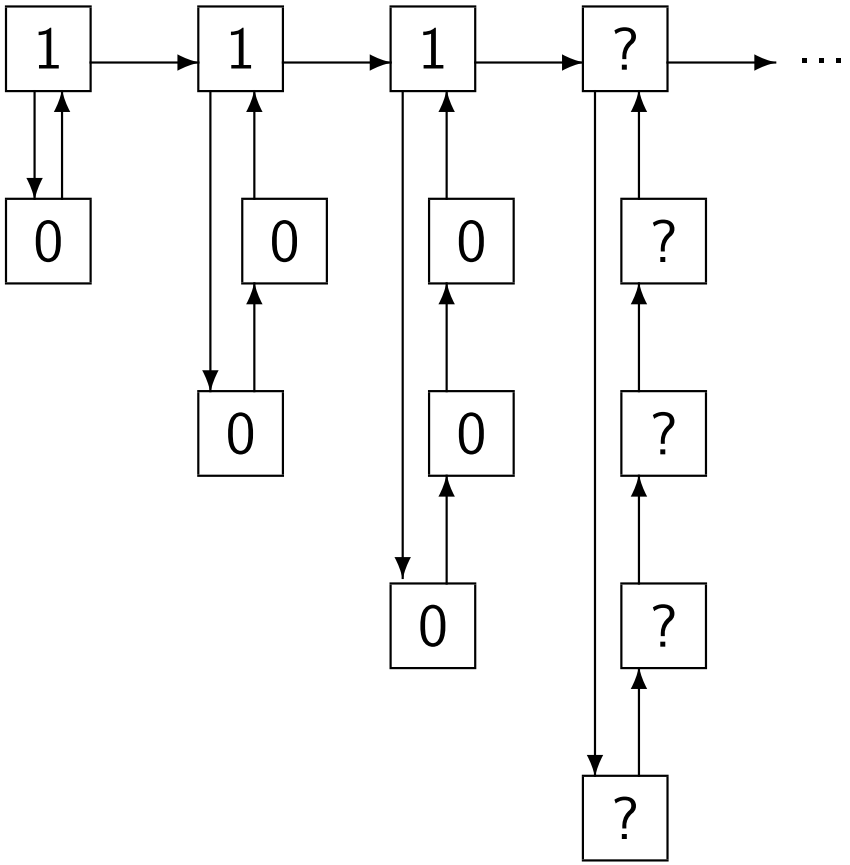
Already for the class $\mathcal{C} = \{\nu_1, \nu_2\}$ there is no policy which attains the best value in both environments $\nu_i \in \mathcal{C}$, even asymptotically.

The scope

For which classes of environments does a universal (self-optimizing) policy exist?

The aim is to find as general as possible classes of environments which "forgive" wrong steps.

Slowly forgiving



Define the problem more formally.

For an environment ν and a policy p define

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

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 there is a limiting average value $V(\nu,p) =: V$.
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 explorable environments $\mathcal{C}$ if $V(\nu,p) = V^*_\nu$ for every $\nu \in \mathcal{C}$.

Positive examples

Self-optimizing policies exist for the class of finite ergodic MDPs (probably the most popular class of environments in reinforcement learning), for the class of sequence prediction problems, and for some others. We try to identify the general requirements for the existence of self-optimizing policies.

Definition (value-stable environments)

An explorable environment ν is *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 $\frac{1}{n} r^\nu_{1..n} \to V^*_\nu$, $d_\nu(k,\varepsilon) = o(k)$, $\sum_{n=1}^\infty \varphi_\nu(n,\varepsilon) < \infty$ for every fixed ε, and for every k and every history $z_{<k}$ there exists a policy $p = p_\nu^{z_{<k}}$ such that

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

Suppose that a person A has made k possibly suboptimal actions and after that "realized" how to act optimally. 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 main theorem

Theorem (value-stable⇒self-optimizing)

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

Examples

Finite ergodic Markov decision processes (MDPs) and some classes of finite partially observable MDPs are value-stable. Certain mixing conditions imply value-stability. There are many value-stable environments beyond finite (PO)MDPs.

For an ergodic MDP $d(n, \varepsilon) \equiv 0$, $r_i = const$, $\varphi(n, \varepsilon)$ decay exponentially fast.

non-markovian value-stable environments: an example

Infinitely armed bandit:

There is a countable family $\{\zeta_i : i \in \mathbb{N}\}$ 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 Y is $\{g, u, d\}$. 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 u and d is 0.

It is easy to see that the environment ν just constructed is value-stable with $d(k, \varepsilon) = 1$.

Moreover, if we change the reaction to action d so that it moves the agent not to the arm 0 but some $d(k)$ arms down, where k is the current position, we can make $d(k)$ as close to linear in k as desired.

Necessity of the conditions

$d(n, \varepsilon) = o(n)$ can not be relaxed to $O(n)$.
$\varphi(n, \varepsilon)$ can not be allowed to depend arbitrary on the history $z_{<k}$.
Examples are easy to construct.

Open questions

To find necessary and sufficient conditions: $\varphi(n, \varepsilon)$ can be allowed to depend on past observations in some way.

To find conditions on not necessarily countable classes $\mathcal{C}$ which would guarantee the existence of self-optimizing policies.