Average Reward Optimization with Multiple Discounting Reinforcement Learners

Chris Reinke[1], Eiji Uchibe[1,2], and Kenji Doya[1]

[1]Okinawa Institute of Science and Technology
904-0495 Okinawa, Japan
[2]ATR Computational Neuroscience Laboratories
619-0288 Kyoto, Japan
{chris.reinke,uchibe,doya}@oist.jp
http://groups.oist.jp/ncu

Abstract. Maximization of average reward is a major goal in reinforcement learning. Existing model-free, value-based algorithms such as R-Learning use average adjusted values. We propose a different framework, the Average Reward Independent Gamma Ensemble (AR-IGE). It is based on an ensemble of discounting Q-learning modules with a different discount factor for each module. Existing algorithms only learn the optimal policy and its average reward. In contrast, the AR-IGE learns different policies and their resulting average rewards. We prove the optimality of the AR-IGE in episodic and deterministic problems where rewards are given at several goal states. Furthermore, we show that the AR-IGE outperforms existing algorithms in such problems, especially in situations where policies have to be changed due to changes in the task. The AR-IGE represents a new way to optimize average reward that could lead to further improvements in the field.

Keywords: reinforcement learning, average reward, model-free, value-based, Q-learning, modular

1 Introduction

Reinforcement learning aims to maximize cumulative reward in decision-making tasks that are modeled as Markov decision processes (MDP) [9]. Two major goals can be differentiated. The maximization of (a) discounted reward or (b) average reward:

$$\text{(a)} \lim_{N\to\infty} \sum_{t=0}^{N} \gamma^t r_t \ , \qquad \text{(b)} \lim_{N\to\infty} \frac{1}{N+1} \sum_{t=0}^{N} r_t \ , \tag{1}$$

where $\gamma \in [0, 1]$ is a discount factor and r_t is the immediate reward at time t. Different value-based, model-free methods exist for both goals. To maximize

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-319-70087-8_81

discounted reward, methods such as Q-Learning [13] learn discounted values:

$$Q^\pi(s_t, a_t) = r_t + \gamma Q^\pi(s_{t+1}, a_{t+1}) \ .$$

Methods to maximize average reward, such as R-Learning [8], learn average adjusted values:

$$Q^\pi(s_t, a_t) = r_t - \rho^\pi + Q^\pi(s_{t+1}, a_{t+1}) \ ,$$

where ρ^π is the average reward that results from policy π and is defined by:

$$\rho^\pi = \lim_{N \to \infty} \frac{1}{N+1} \sum_{t=0}^{N} r(s_t, \pi(s_t)) \ .$$

Discounted reward methods such as Q-learning are simpler and more commonly used than average reward methods, because average reward methods have to learn two entities, the values $Q^\pi(s, a)$ and the average reward ρ^π. Nonetheless, maximization of average reward would be often preferable because it results in the maximum reward for the invested time.

In the field of policy search methods [2], it is usually assumed that the Markov chain for the given policy is ergodic and that there exists a unique stationary distribution that is equal to the limiting distribution. The stationary distribution is independent of the initial state. Although this assumption is theoretically convenient, it is problematic for MDPs with absorbing states as considered in this paper.

We propose a new method, the Average Reward Independent Gamma Ensemble (AR-IGE), which uses an ensemble of discounting Q-Learning modules to maximize average reward for a sub-class of MDPs: episodic, deterministic goal-only-reward MDPs. In contrast to typical average reward methods it does not require an estimate of the average reward ρ^π. Furthermore, instead of learning only the optimal policy and its average reward it learns several possible policies for a task and their average reward. This can be useful if, for example, the algorithm has to adapt to changes in the MDP. The AR-IGE is able to rapidly switch to another policy if the currently optimal policy becomes suboptimal.

The next section introduces relevant average reward algorithms and methods similar to the AR-IGE. Afterward, the AR-IGE and the MDPs for which it is designed are described, followed by the optimality proof of the algorithm. Finally its performance is compared to other average reward algorithms.

2 Related Work

Existing model-free, value-based average reward algorithms are proposed for continuous MDPs that are irreducible, aperiodic, and ergodic [6]. Nonetheless, they can also be applied to episodic MDPs, which are the scope of this paper. All algorithms update their Q-values after a transition from s_t to s_{t+1} with action a_t based on the received reward r_t and a stochastic estimation of the average

reward ρ:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_Q \left(r_t - \rho + \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right) .$$

α_Q is the learning rate for the values. Algorithms differ in their update method for the average reward estimation ρ.

R-Learning [8] was the first variant of these algorithms. Its average reward is updated by:

$$\rho \leftarrow \rho \ + \ \alpha_\rho \left(r_t - \rho + \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - \max_{a_t} Q(s_t, a_t) \right) ,$$

where α_ρ is the learning rate for the average reward.

SMART [1, 5] calculates the average reward by summing over all perceived rewards during the experiment and dividing it by the total number of time steps:

$$\rho \leftarrow \frac{\sum_{t=0}^{T} r_t}{T+1} ,$$

where $T+1$ is the total number of time steps up to the current time point.

rSMART [3] is variant of SMART for which a convergence proof of the algorithm was developed. It uses a relaxed update of the average reward:

$$\rho \leftarrow \rho + \alpha_\rho \left(\frac{\rho \cdot (T-1) + r_t}{T} \right) .$$

rmSMART [3] is a further variant of SMART which uses a Robbins-Monro update of the average reward:

$$\rho \leftarrow \rho + \alpha_\rho (r_t - \rho) .$$

Recently, CSV-Learning was proposed [14] where ρ is not updated during the learning but set to an initial guess. This algorithm outperforms existing approaches, but we excluded it from our analysis because of the required initial guess of the average reward.

In contrast to existing average reward methods, the AR-IGE is inspired by research about human decision-making, which suggests that distinct brain regions in the striatum exist, where each learns values for choices with a different discount factor [10]. Based on these results, a framework similar to the AR-IGE was proposed [4]. It also uses modules that learn discounted values, but it has a different value definition. Its aim is to model the hyperbolic discounting behavior that is observed in human decision-making. Under specific conditions it is also able to maximize average reward, but it can not guarantee optimality [7].

3 Goal-Only-Reward MDPs

The AR-IGE is primarily designed for deterministic *goal-only-reward* MDPs (see Fig. 1 for an example). MDPs are defined as a tupel (S, A, T, R) with a set of

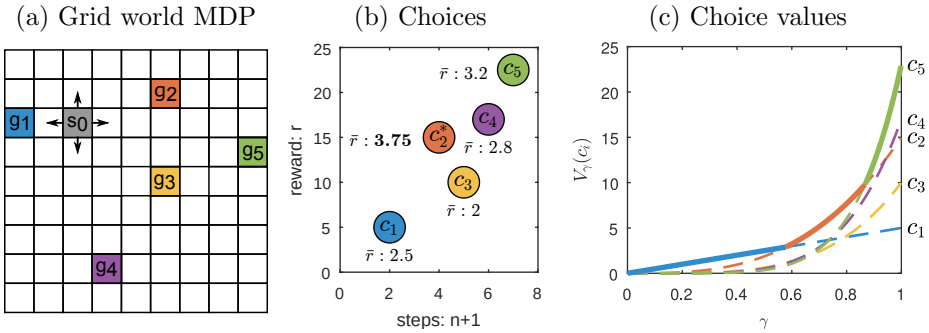

Fig. 1. (a) A deterministic, goal-only-reward grid world. The agent starts in s_0 and can move in 4 directions. It receives a reward by reaching one of 5 goal states. (b) The choice set $C(s_0)$ shows the reward r, the minimum number of steps $n+1$, and the average reward $\bar{r} = \frac{r}{n+1}$ for reaching each goal. (c) The choice values show that the AR-IGE learns policies for c_1, c_5 and the optimal choice c_2^* (solid line).

states S, a set of actions A, a deterministic transition function $s' = T(s,a)$ $(s, s' \in S,\, a \in A)$ and a reward function $R(s) \in \mathbb{R}$. MDPs are episodic, i.e. they have episodes that begin in a start state $s_0 \in S$ and end if a goal state $g \in G \subseteq S$ is reached. For goal-only-reward MDPs, a reward is only given if a goal state is reached: $\forall s \notin G: \; R(s) = 0$. The goal is to maximize the average reward per episode:

$$\max_\pi \frac{\sum_{t=0}^{N} r_t}{N+1} \;=\; \max_\pi \frac{r_N}{N+1} \;,$$

where $N+1$ is the number of steps until a goal state is reached. If the start state is the same over episodes, then this also maximizes the average reward over episodes.

4 The Average Reward Independent Gamma Ensemble

The AR-IGE (Algorithm 1) is composed of γ-modules. Each module is a Q-function $Q_\gamma(s,a)$ with a different discount factor $\gamma \in (0,1)$. Discount factors $\Gamma = (\gamma_1, \ldots, \gamma_M)$ are sorted $(\gamma_i < \gamma_{i+1})$ without loss of generality.

In the beginning of each episode, one γ-module is selected. Its values are used to define the policy, for example, with an ϵ-Greedy action selection, until the episode ends. During the episode, values of all γ-modules will be updated after each observation of (s_t, a_t, r_t, s_{t+1}) by the standard Q-learning update:

$$Q_\gamma(s_t,a_t) \leftarrow Q_\gamma(s_t,a_t) + \alpha_Q \left(r_t + \gamma \max_{a_{t+1}} Q_\gamma(s_{t+1},a_{t+1}) - Q_\gamma(s_t,a_t) \right),$$

with learning rate $\alpha_Q \in [0,1]$. Because Q-learning is an off-policy algorithm, all modules can be updated after an observation, regardless of the module that was used to define the policy [9].

Algorithm 1: Average Reward Independent γ-Ensemble

Input:
Learning rate: $\alpha_Q \in [0, 1]$
Sorted list of M discount factors: $\Gamma = (\gamma_1, \ldots, \gamma_M)$ with $\gamma_i \in (0, 1)$
Module selection parameter: ϵ_M
initialize $\forall \gamma \in \Gamma : \; Q_\gamma(s, a) \leftarrow 0$
repeat (for each episode)
  initialize s
  for i **from** *1* **to** $M - 1$ **do**
    $\bar{R}(i) \leftarrow \bar{r}_s(\gamma_i, \gamma_{i+1})$ (Eq. 2)
  end
  $\gamma^* \leftarrow \Gamma\left(\epsilon - \text{Greedy module selection}(\bar{R}, \epsilon_M)\right)$
  repeat (for each step in episode)
    choose an action a derived from $Q_{\gamma^*}(s, a)$
    take action a, observe r, s'
    forall the $\gamma \in \Gamma$ **do**
      $Q_\gamma(s, a) \leftarrow Q_\gamma(s, a) + \alpha_Q \left(r + \gamma \max_{a'} Q_\gamma(s', a') - Q_\gamma(s, a)\right)$
    end
    $s \leftarrow s'$
  until s *is goal state*
until *termination*

Over time, each module will learn its optimal policy, i.e. the policy with the minimum number of steps to reach a goal state. The optimal goal state of a module depends on its discount factor γ, because it defines how strongly the reward of a goal is discounted over the steps to reach it (Fig.1, c). For example, small γ's have strong discounting and prefer goals that are reachable with few steps.

To maximize the average reward, at the beginning of each episode the AR-IGE calculates the average reward for each module pair $\bar{r}_s(\gamma_i, \gamma_{i+1})$ by iterating over the neighbors in the list of modules Γ. The average reward of a module pair can be calculated by:

$$\bar{r}_s(\gamma_a, \gamma_b) = \frac{V_{\gamma_a}(s)}{{\gamma_a}^{n_s(\gamma_a, \gamma_b)} \left(n_s(\gamma_a, \gamma_b) + 1\right)} \; , \tag{2}$$

with $V_\gamma(s) = \max_a Q_\gamma(s, a)$ and the number of steps to their goal state:

$$n_s(\gamma_a, \gamma_b) = \frac{\log(V_{\gamma_a}(s)) - \log(V_{\gamma_b}(s))}{\log(\gamma_a) - \log(\gamma_b)} \; . \tag{3}$$

Section 5.3 shows the derivation of Eq. 2 and 3. Then a module γ^* is selected with an ϵ-Greedy selection strategy based on their average reward. The policy of the selected module is used for the action selection during the episode. The module selection chooses with probability $1 - \epsilon_M$ the module with the highest

average reward and with probability ϵ a random module. Sometimes choosing a non-optimal module allows AR-IGE to explore different policies, which can be useful in tasks that change over time.

5 Optimality

This section proves the optimality of the AR-IGE to maximize average reward for deterministic, goal-only-reward MDPs. The following assumptions are required: (a) The Q-functions of all γ-modules converge to their optimal values as proven by [11, 12]. (b) The number of modules goes to infinity: $M \to \infty$. (c) The discount factors Γ are evenly spread between 0 and 1. Nonetheless, in practice the number of modules needs only to be reasonably high as we show experimentally in Section 6.

Based on these assumptions and if the following two points are true, the optimality is proven: 1) The policies of optimal Q-functions $Q^*_{\gamma^*}(s, a)$ for certain discount factors $\gamma^* \in (0, 1)$ result in the maximum average reward of an episode (Theorem 1). 2) The average reward in which policies of Q-functions result can be calculated (Theorem 2).

The first point and the assumptions (a,b,c) ensure that γ-modules exist which maximize the average reward. The second point ensures that a module with the maximum average reward policy can be identified and used. The next section introduces some preliminaries followed by the proofs for both points. We assume for simplification that all rewards are non-negative ($R(s) \geq 0$).

5.1 Preliminaries

We first define the concept of choice that is used throughout the proof. A *choice* $c \in C(s)$ describes a possible solution, i.e. a trajectory with the minimum number of steps to reach a goal state $g \in G$, for a deterministic goal-only-reward MDP if started in state $s \in S$. Each choice is a tupel (r, n) with the expected reward $r = E[R(g)]$ and the minimum number of steps n reduced by 1 to reach the goal state. Steps are reduced by 1 to make them compatible with the standard formulation of discounted rewards (Eq. 1) where the steps to reach a goal state are counted by $t = [0, 1, \ldots, n]$. Therefore, the average reward of a choice is $\bar{r}(c) = \frac{r}{n+1}$. The set of all possible choices $C(s)$ in an MDP holds a choice for each reachable goal state from start state s (Fig.1, b).

The value of a choice $V_\gamma(c)$ is the expected discounted future reward from using the optimal policy to reach its goal state. The value depends on the discount factor γ: $V_\gamma(c) = E[r_0 + \gamma r_1 + \ldots + \gamma^n r_n] = \gamma^n r$.

A choice is also an outcome of an optimal Q-function if its greedy policy is used because the policy takes the minimum number of steps to reach a goal state. The choice describes the resulting reward and the number of steps that are needed. The outcome of Q-function Q_γ with discount factor γ is the choice with the maximum value for this γ: $\max_{c \in C(s)} V_\gamma(c)$ (Fig.1, c).

Choices are either *dominated* or *non-dominated*. A choice $\hat{c}_1 \in \hat{C}(s)$ dominates another choice $\check{c}_2 \in \check{C}(s)$ ($\hat{c}_1 \gg \check{c}_2$) if it is shorter $n_1 \leq n_2$ and it has a higher reward $r_1 \geq r_2$. A choice is dominated $\check{c} \in \check{C}(s_I)$ if there exists at least one other choice that dominates it ($\exists \hat{c} \in C(s_I) : \ \hat{c} \gg \check{c}$). The set of non-dominated choices can be ordered ($\hat{c}_1 < \ldots < \hat{c}_\mathrm{K}$) based on the number of steps required, from the shortest to the longest: $n_{i-1} < n_i < n_{i+1}$. This also results in an ordering of their reward values: $r_{i-1} < r_i < r_{i+1}$.

5.2 Existence of the Optimal Average Reward Policy

With help of the introduced concepts about choices, the first point of the optimality proof can be shown:

Theorem 1. *For each start state $s \in S$ exist discount factors $\gamma^* \in (0,1)$ for which the greedy policies $\pi^*_{\gamma^*}(s,a) = \mathrm{argmax}_a\, Q^*_{\gamma^*}(s,a)$ of their optimal Q-functions result in the maximum average reward of an episode.*

To prove Theorem 1 we show that the optimal choice $c^* = \mathrm{argmax}_{c \in C(s)}\, \bar{r}(c)$ is the outcome of Q-functions for certain discount factors γ^*. We first show that dominated choices can be ignored because they cannot be maximum average reward solutions (Lemma 1). In a second step we show that for the optimal non-dominated choice, a discount factor region exists for which it is the outcome of Q-functions (Lemma 2).

Lemma 1. *Dominated choices $\check{C}(s)$ cannot be the maximum average reward choice: $c^* \notin \check{C}(s)$.*

We can ignore dominated choices because a dominated choice $\check{c}_2$ ($c_1 \gg \check{c}_2$) has a smaller reward $r_1 \geq r_2 > 0$ and more steps $0 < n_1 \leq n_2$ compared to a dominating choice $\hat{c}_1$. Thus, its average reward is smaller than that of the choice that dominates it: $\frac{1}{n_1+1} \geq \frac{1}{n_2+1} \ \Leftrightarrow \ \frac{r_1}{n_1+1} \geq \frac{r_2}{n_2+1}$. Therefore, the maximum average reward choice can only be a non-dominated choice, and we only need to show that for the optimal non-dominated choice a discount factor region exists for which it is the outcome of Q-functions.

Lemma 2. *Discount factors $\gamma^* \in (0,1)$ exist for which the non-dominated choice with the maximum average reward $\hat{c}^* \in \hat{C}(s_I)$ is an outcome of Q-functions.*

Three cases have to be considered, depending on the position of the optimal choice $\hat{c}^*$ in the ordered set of non-dominated choices $\hat{C} = (\hat{c}_1 < \ldots < \hat{c}_K)$. The three cases are: 1) $\hat{c}^*$ is the first non-dominated choice ($\hat{c}^* = \hat{c}_1$), 2) the last choice ($\hat{c}^* = \hat{c}_\mathrm{K}$) or 3) an intermediate choice ($\hat{c}^* \in [\hat{c}_2, ..., \hat{c}_{\mathrm{K}-1}]$).

To prove each case, we need to know the discount factor regions $\gamma_{\hat{c}_1}$ and $\gamma_{\hat{c}_2}$ for which two given non-dominated choices ($\hat{c}_1 < \hat{c}_2$) are the outcome. We can do this by defining the γ_E for which both of their values are equal:

$$V_{\gamma_E}(\hat{c}_1) = V_{\gamma_E}(\hat{c}_2) \quad \Leftrightarrow \quad {\gamma_E}^{n_1} r_1 = {\gamma_E}^{n_2} r_2 \quad \Leftrightarrow \quad \gamma_E = \left(\frac{r_1}{r_2}\right)^{\frac{1}{n_2 - n_1}} \ .$$

Having γ_E it is obvious that choice $\hat{c}_1$ is the outcome for $0 \leq \gamma_{\hat{c}_1} < \gamma_E$ and $\hat{c}_2$ for $\gamma_E < \gamma_{\hat{c}_2} \leq 1$. With help of γ_E we can show that for each case, discount factors γ^* exist for which the optimal choice is an outcome.

The first and the second case are true, because the first and the last non-dominated choice will be solutions for discount factors near 0 and 1. More formally, the first choice $\hat{c}_1$ is the outcome for $0 < \gamma^* < \min_{\forall \hat{c}_i > \hat{c}_1} \gamma_E(\hat{c}_1, \hat{c}_i)$. To prove its existence, it is enough to show the general case:

$$\forall \hat{c}_i > \hat{c}_1 : \quad \gamma_E(\hat{c}_1, \hat{c}_i) > 0 \quad \Leftrightarrow \quad \left(\frac{r_1}{r_i}\right)^{\frac{1}{n_i - n_1}} > 0 \ ,$$

which is true because $x^y > 0$ for $x > 0$.

In the second case, the last non-dominated choice $\hat{c}_\mathrm{K}$ is the outcome of Q-functions with $\max_{\forall \hat{c}_i < \hat{c}_\mathrm{K}} \gamma_E(\hat{c}_i, \hat{c}_\mathrm{K}) < \gamma^* < 1$. To prove its existence, it is enough to show again the general case:

$$\forall \hat{c}_i < \hat{c}_\mathrm{K} : \quad \gamma_E(\hat{c}_i, \hat{c}_\mathrm{K}) < 1 \quad \Leftrightarrow \quad \left(\frac{r_i}{r_\mathrm{K}}\right)^{\frac{1}{n_\mathrm{K} - n_i}} < 1 \ ,$$

which is true because $r_\mathrm{K} > r_i \Leftrightarrow \frac{r_i}{r_\mathrm{K}} < 1$ and $0 < x^y < 1$ for $0 < x < 1$ and $y > 0$.

In the third case, the optimal choice is intermediate: $\hat{c}^* \in [\hat{c}_2, \ldots, \hat{c}_{\mathrm{K}-1}]$. If discount factors exist for which the optimal choice $\hat{c}^*$ is an outcome, they are in the region: $\max_{\forall \hat{c}_i < \hat{c}^*} \gamma_E(\hat{c}_i, \hat{c}^*) < \gamma^* < \min_{\forall \hat{c}_k > \hat{c}^*} \gamma_E(\hat{c}^*, \hat{c}_k)$. We can show that this region exists by proving the general case:

$$\begin{aligned} \forall \hat{c}_i < \hat{c}^*, \hat{c}_k > \hat{c}^* : \quad & \gamma_E(\hat{c}_i, \hat{c}^*) < \gamma_E(\hat{c}^*, \hat{c}_k) \ \Leftrightarrow \ \left(\tfrac{r_i}{r^*}\right)^{\frac{1}{n^* - n_i}} < \left(\frac{r^*}{r_k}\right)^{\frac{1}{n_k - n^*}} \\ & \Leftrightarrow \ \left(\frac{r^*}{r_k}\right)^{\frac{1}{n_k - n^*}} - \left(\tfrac{r_i}{r^*}\right)^{\frac{1}{n^* - n_i}} > 0 \ . \end{aligned} \tag{4}$$

We can construct a lower bound for the left hand side using the fact that c^* is the choice with the maximum average reward. Using $\frac{r^*}{n^*+1} > \frac{r_k}{n_k+1} \Leftrightarrow \frac{r^*}{r_k} > \frac{n^*+1}{n_k+1}$ and $\frac{r^*}{n^*+1} > \frac{r_i}{n_i+1} \Leftrightarrow \frac{r^*}{r_i} > \frac{n^*+1}{n_i+1} \Leftrightarrow \frac{r_i}{r^*} < \frac{n_i+1}{n^*+1}$ the lower bound is given by: $\left(\frac{r^*}{r_k}\right)^{\frac{1}{n_k - n^*}} - \left(\frac{r_i}{r^*}\right)^{\frac{1}{n^* - n_i}} > \left(\frac{n^*+1}{n_k+1}\right)^{\frac{1}{n_k - n^*}} - \left(\frac{n_i+1}{n^*+1}\right)^{\frac{1}{n^* - n_i}}$.

Therefore, to prove Eq. 4 we need to show that the lower bound is larger than 0. This can be done by reformulating it using the monotonic increasing logarithm ($x < y \ \Leftrightarrow \ \ln(x) < \ln(y)$):

$$\begin{aligned} \forall \hat{c}_i < \hat{c}^*, \hat{c}_k > \hat{c}^* : \quad & \left(\frac{n^*+1}{n_k+1}\right)^{\frac{1}{n_k - n^*}} - \left(\frac{n_i+1}{n^*+1}\right)^{\frac{1}{n^* - n_i}} > 0 \\ & \Leftrightarrow \ln\left(\left(\frac{n^*+1}{n_k+1}\right)^{\frac{1}{n_k - n^*}}\right) - \ln\left(\left(\frac{n_i+1}{n^*+1}\right)^{\frac{1}{n^* - n_i}}\right) > 0 \\ & \Leftrightarrow -(n^* - n_i)\ln\left(\frac{n_k+1}{n^*+1}\right) - (n_k - n^*)\ln\left(\frac{n_i+1}{n^*+1}\right) > 0 \ . \end{aligned}$$

With $\forall x > 0: \ \ln(x) \leq x - 1 = -\ln(x) \geq -(x-1)$ and where equality holds for $x = 1$ another lower bound can be constructed that is 0:

$$\forall \hat{c}_i < \hat{c}^*, \hat{c}_k > \hat{c}^*: \quad -(n^* - n_i)\ln\left(\tfrac{n_k+1}{n^*+1}\right) - (n_k - n^*)\ln\left(\tfrac{n_i+1}{n^*+1}\right) > -(n^* - n_i)\left(\tfrac{n_k+1}{n^*+1} - 1\right) - (n_k - n^*)\left(\tfrac{n_i+1}{n^*+1} - 1\right) = 0 \ .$$

This concludes the proof of Eq. 4 and Lemma 2. Combining Lemma 1 and 2 we can see that the maximum average reward choice is a non-dominated choice and that discount factors $\gamma^* \in (0, 1)$ exist for which it is the outcome of Q-functions (Theorem 1).

5.3 Identification of the Optimal Policy

After proving that γ-modules exist that maximize the average reward, we show that the average reward of modules can be calculated. If this is true, the AR-IGE can correctly identify and use the optimal module for action selection.

The AR-IGE computes the average reward of modules by Eq. 2 using the values of two neighboring modules ($\gamma_a \approx \gamma_b$). The values of neighboring modules are similar because their discount factors are similar. Therefore, their outcome, i.e. their choice $c = (r, n)$, is usually the same (Fig.1, c). As a result we can calculate n based on their values:

Theorem 2. *Given two optimal Q-functions with discount factors γ_a and γ_b that have the same choice as an outcome, the average reward for an episode that starts in state $s \in S$ can be computed by Eq. 2.*

If two Q-functions result in the same choice $c = (r, n)$ their values can be used to compute the number of steps n to reach the goal state as defined in Eq. 3:

$$\begin{matrix} V_{\gamma_a}(s) = {\gamma_a}^n r \\ V_{\gamma_b}(s) = {\gamma_b}^n r \end{matrix} \quad \Leftrightarrow \quad \frac{V_{\gamma_a}(s)}{{\gamma_a}^n} = \frac{V_{\gamma_b}(s)}{{\gamma_b}^n} \quad \Leftrightarrow \quad n = \frac{\log(V_{\gamma_a}(s)) - \log(V_{\gamma_b}(s))}{\log(\gamma_a) - \log(\gamma_b)} \ . \tag{5}$$

The number of steps n can be used with the value of one of the Q-functions to compute r:

$$V_{\gamma_a}(s) = {\gamma_a}^n r \quad \Leftrightarrow \quad r = \frac{V_{\gamma_a}(s)}{{\gamma_a}^n} \ . \tag{6}$$

Given r (Eq.6) and n (Eq.5) the average reward $\frac{r}{n+1}$ can be computed as defined in Eq. 2.

A problem occurs if the choice of two neighboring modules is different. In this case, the computed $\bar{r}_s$ and n_s are wrong. Nonetheless, these cases can be detected by comparing them to neighboring pairs. If $n_s(\gamma_a, \gamma_b) \neq n_s(\gamma_b, \gamma_c) \neq n_s(\gamma_c, \gamma_d)$, then the modules of pair (γ_b, γ_c) are ignored.

6 Experiments

We compared the AR-IGE to the classical algorithms described in Section 2 in two problem domains: tree MDPs and grid world MDPs. For each domain, 100 MDPs have been randomly generated. Tree MDPs have the form of directed, rooted tree graphs. The start state is the root of the graph. Paths of different lengths lead to goal states. Their number is uniformly sampled between 3 to 6. Paths branch from each other at certain states where the agent has to choose between them. The length of each path and the reward for reaching its goal state are uniformly sampled between 1 and 25. Branch points between paths are also sampled uniformly.

Grid world MDPs have a 9x9 state space, as shown in Fig. 1. Each MDP has one randomly placed start state and a set of 3 to 6 randomly placed goal states. One goal state was randomly selected as the optimal average reward choice and its reward r^* was uniformly drawn between 1 and 25. The reward of every other goal was uniformly drawn between 0 and $n \cdot r^*$ where n is the minimum number of steps to reach the goal from the start state.

Each algorithm was executed for 100 runs per MDP to measure its mean performance. Each run was divided in three phases. Phase 1 tested how the algorithms learn a new MDP (tree MDPs: 1000 episodes, grid world MDPs: 3000 episodes). Phase 2 and 3 (tree MDPs: 500 episodes, grid world MDPs: 1500 episodes) tested how algorithms adapt to changes in the MDP. In Phase 2, the

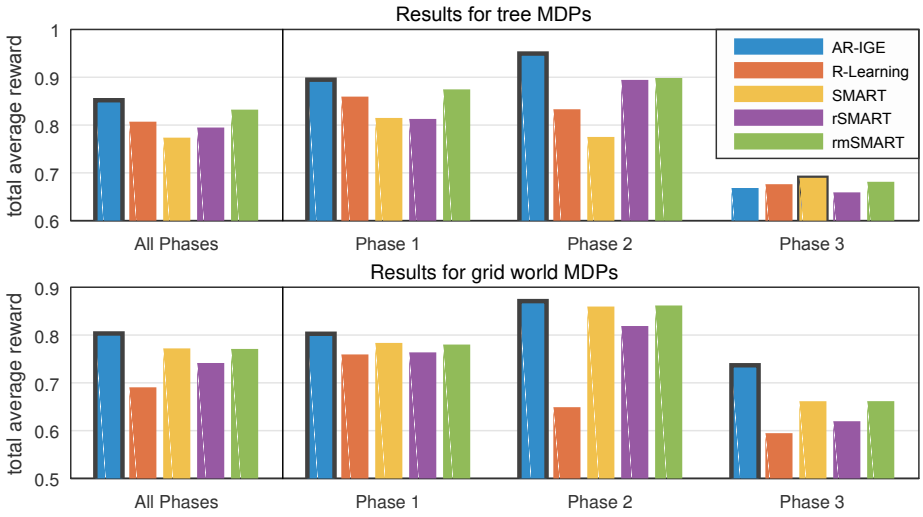

Fig. 2. AR-IGE performs better than classical algorithms in randomly generated tree and grid world MDPs. Performance is measured by the mean total normalized average reward over 100 randomly generated MDPs and 100 runs per MDP. Each run had 3 phases. Parameters of algorithms were optimized to maximize performance over all phases. The best algorithm per phase is marked with a black edge.

reward of the optimal average reward goal was reduced to make it non-optimal, whereas in Phase 3 the reward of a non-optimal goal was increased to make it optimal. In these phases, algorithms had to learn to switch to the new optimal goal.

The AR-IGE had 99 modules with $\Gamma = [0.01 : 0.01 : 0.99]$. An ϵ-Greedy action selection ($\epsilon(k) = \min(\epsilon_0 \cdot d^k, \epsilon_{min})$) that decayed over episodes was used for all algorithms. Its parameters $(\epsilon_0, d, \epsilon_{min})$ and all other learning parameters for the AR-IGE (α_Q, ϵ_M) and the other algorithms (α_Q, α_ρ) were optimized with a grid search. Algorithm performance was measured by the mean of the total average reward over all episodes of 100 runs per MDP and 100 MDPs per problem domain.

The results show that the AR-IGE achieves the highest total average reward over all phases for both task domains (Fig. 2). It also has the highest total average reward per individual phase, beside Phase 3 for the tree domain. In particular, it performs better than the classical algorithms in Phase 2 of the tree domain and Phase 3 of the grid world domain. This is the result of its ability to learn different policies in parallel for different γ-modules. The policies go to different goal states in the MDP. Therefore, if the optimal goal becomes non-optimal, the AR-IGE can switch to a different policy which goes to the new optimal goal. In contrast, the classical algorithms needed to completely relearn their values and their average reward prediction ρ in Phase 2 and 3.

We performed further experiments to test how many γ-modules the AR-IGE needs to successfully optimize the average reward. All experiments used the same tree and grid world MDPs and the same experimental procedure as the previous experiments. Learning parameters of the AR-IGE were set to the best parameters identified by the grid search from the previous experiments (Fig. 2). The number of γ-modules was set between 2 and 99 modules. For each setting of the number of modules, γ parameters were linearly spaced between 0 and 1. The results show that for both MDP domains, optimal average reward performance

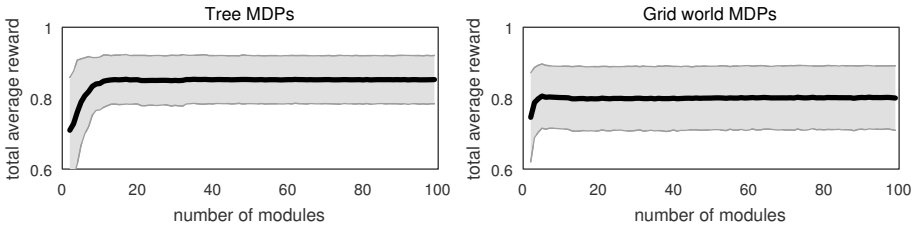

Fig. 3. The performance of the AR-IGE with different numbers of γ-modules shows that only a few modules are needed to reach the optimal performance. The mean and the standard deviation of the normalized total average reward over 100 sampled tree (left) and the grid world MDPs (right) are shown. For tree MDPs, 12 modules were enough to reach the highest average reward performance, and for grid world MDPs, 6 modules were sufficient.

can be reached with relatively few modules (Fig. 3). For tree MDPs, 12 modules were necessary and for grid world MDPs, 6 modules were required.

7 Conclusion

We propose the Average Reward Independent Gamma Ensemble (AR-IGE), a new, unique model-free algorithm to maximize average reward. It consists of an ensemble of discounting Q-Learning modules. We proved its convergence toward the optimal policy for deterministic, goal-only-reward MDPs. Compared to existing average reward algorithms, such as R-Learning, it does not require a stochastic estimation of the average reward ρ in the MDP. It learns different policies in the MDP that represent alternative choices and it is able to calculate their average reward. This is advantageous if the MDP changes, because the algorithm can switch to an alternative policy that behaves better in the changed MDP. The AR-IGE can give rise to a new class of value-based average reward algorithms in reinforcement learning.

Acknowledgement

We thank Tadashi Kozuno for his help with parts of the optimality proof.

References

1. Das, T.K., Gosavi, A., Mahadevan, S., Marchalleck, N.: Solving semi-markov decision problems using average reward reinforcement learning. Management Science 45(4), 560–574 (1999)
2. Deisenroth, M.P., Neumann, G., Peters, J.: A survey on policy search for robotics. Foundations and Trends in Robotics 2(1-2), 1–142 (2011)
3. Gosavi, A.: Reinforcement learning for long-run average cost. European Journal of Operational Research 155(3), 654–674 (2004)
4. Kurth-Nelson, Z., Redish, A.D.: Temporal-difference reinforcement learning with distributed representations. PloS One 4(10), e7362 (2009)
5. Mahadevan, S., Marchalleck, N., Das, T.K., Gosavi, A.: Self-improving factory simulation using continuous-time average-reward reinforcement learning. In: Proceedings of the 14th International Conference on Machine Learning. pp. 202–210 (1997)
6. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc., New York, NY, USA, 1st edn. (1994)
7. Reinke, C., Uchibe, E., Doya, K.: Maximizing the average reward in episodic reinforcement learning tasks. In: 2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS). pp. 420–421. IEEE (2015)
8. Schwartz, A.: A reinforcement learning method for maximizing undiscounted rewards. In: Proceedings of the tenth international conference on machine learning. vol. 298, pp. 298–305 (1993)
9. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. Cambridge Univ Press (1998)

10. Tanaka, S.C., Schweighofer, N., Asahi, S., Shishida, K., Okamoto, Y., Yamawaki, S., Doya, K.: Serotonin differentially regulates short-and long-term prediction of rewards in the ventral and dorsal striatum. PLoS One 2(12), e1333 (2007)
11. Tsitsiklis, J.N.: Asynchronous stochastic approximation and q-learning. Machine Learning 16(3), 185–202 (1994)
12. Watkins, C.J., Dayan, P.: Q-learning. Machine learning 8(3-4), 279–292 (1992)
13. Watkins, C.J.C.H.: Learning from delayed rewards. Ph.D. thesis, University of Cambridge England (1989)
14. Yang, S., Gao, Y., An, B., Wang, H., Chen, X.: Efficient average reward reinforcement learning using constant shifting values. In: Thirtieth AAAI Conference on Artificial Intelligence (2016)