Discounting – does it make sense?

Abhishek Naik
TTT
15th August 2019

Outline

- Why do we use discounting?
- Why do we have to let it go?
- What else could we do?

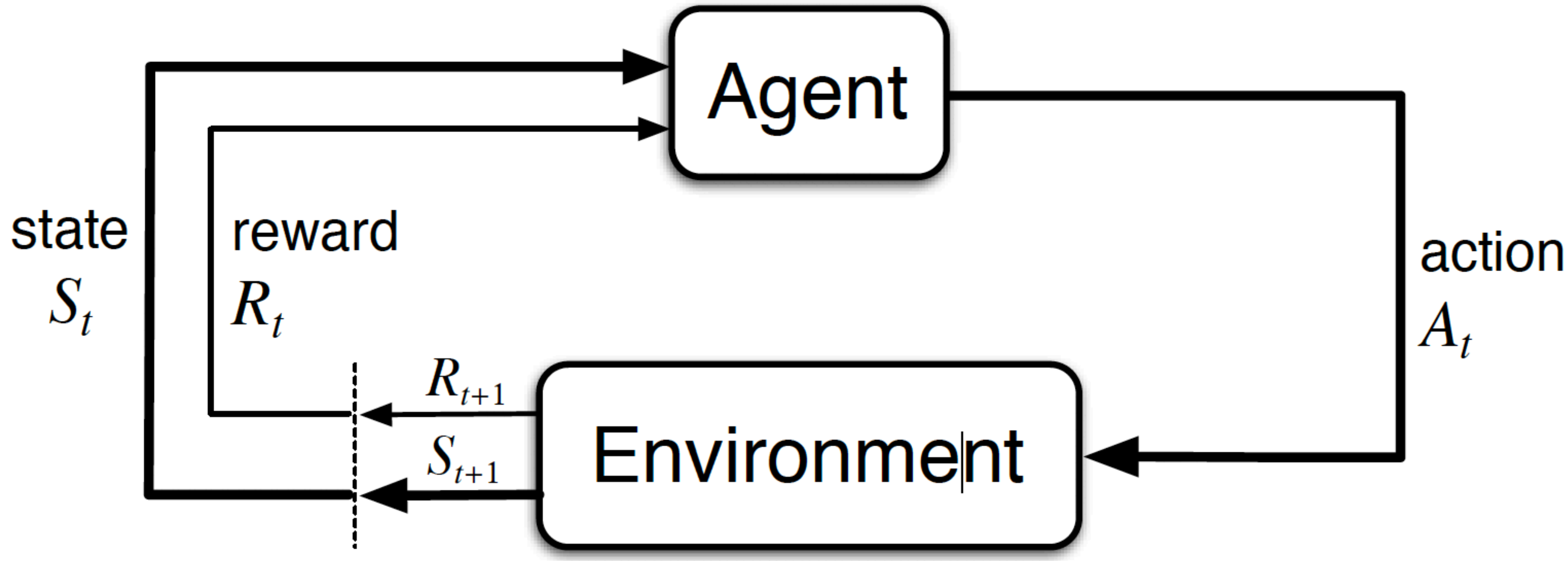


Figure 3.1: The agent–environment interaction in a Markov decision process.

Sutton and Barto (2018)

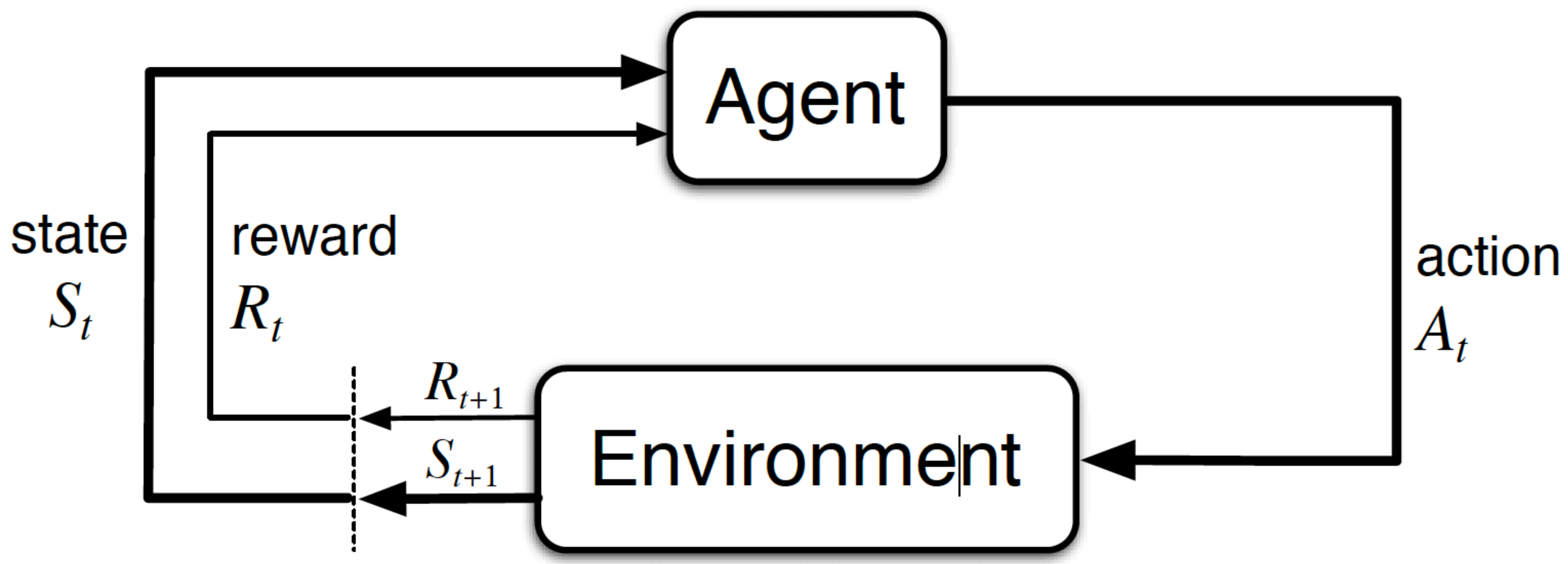


Figure 3.1: The agent–environment interaction in a Markov decision process.

Sutton and Barto (2018)

- In episodic problems:

- In continuing problems:

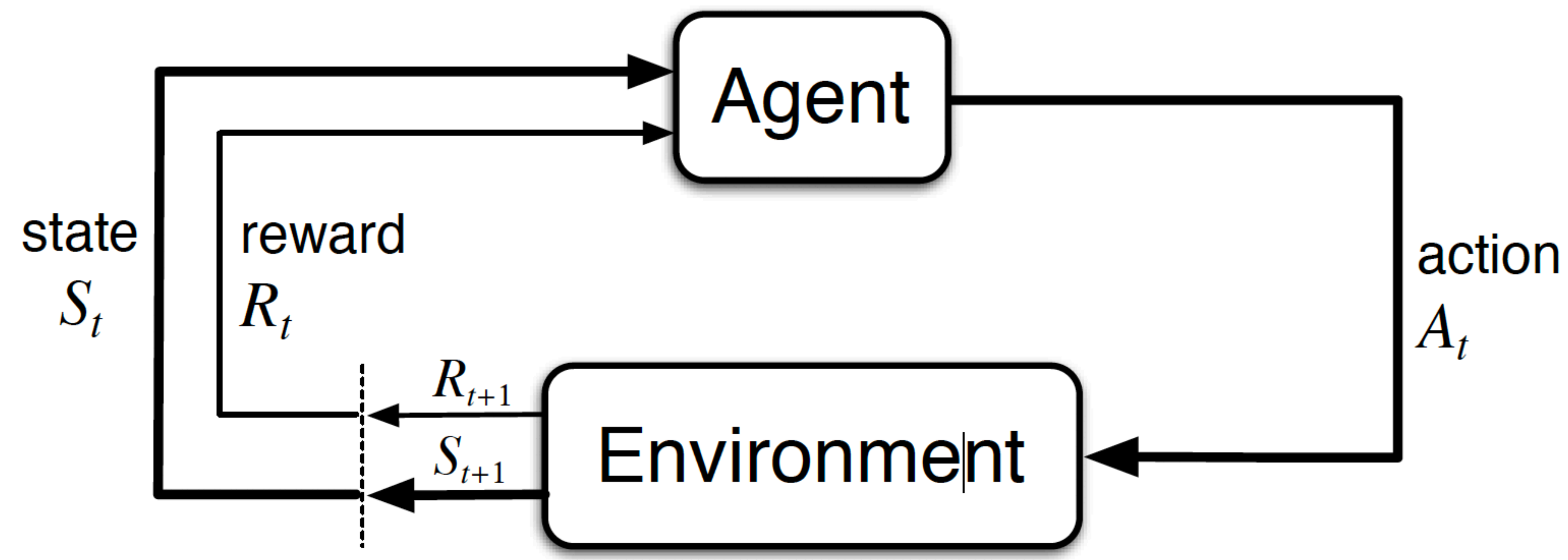


Figure 3.1: The agent–environment interaction in a Markov decision process.

Sutton and Barto (2018)

- In episodic problems:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \ldots + R_{t+T}$$

- In continuing problems:

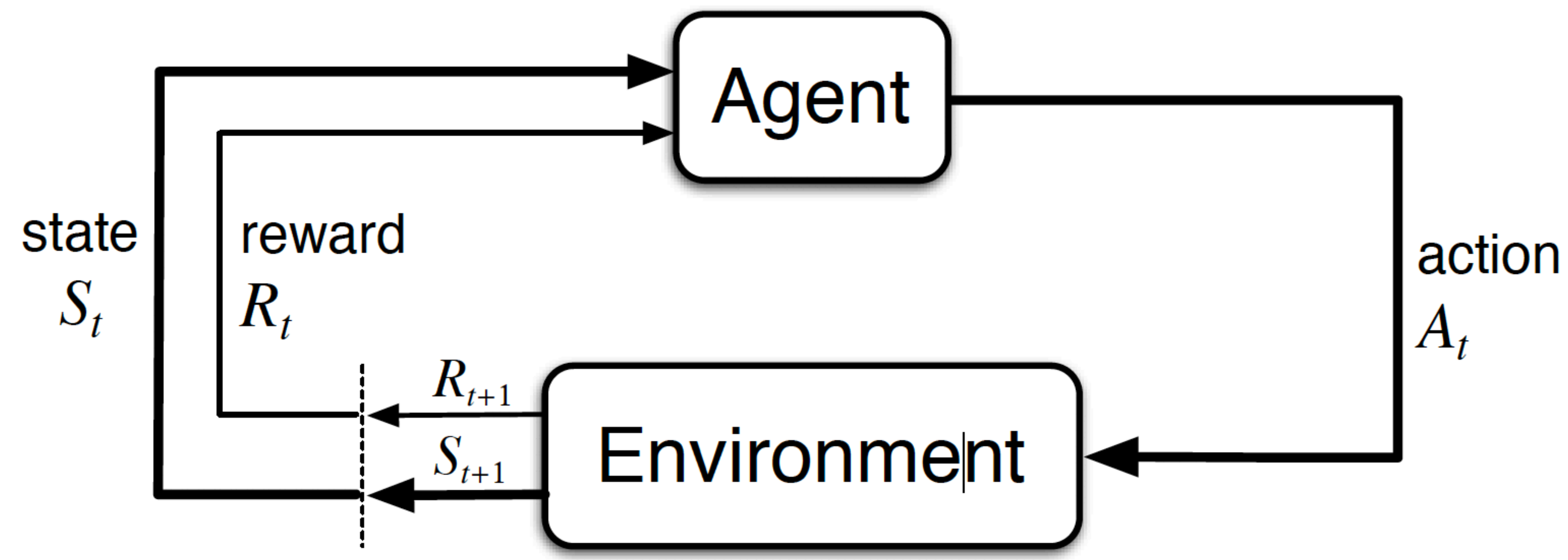


Figure 3.1: The agent–environment interaction in a Markov decision process.

Sutton and Barto (2018)

- In episodic problems:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \ldots + R_{t+T}$$

- In continuing problems:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \ldots \quad \infty \qquad ?$$

Temporal discounting

Temporal discounting

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \; \infty$$

$$\gamma \in [0,1)$$

Why do we love discounting?

- Mathematical convenience

Assume, without loss of generality, that $\mathcal{Q}_0(x, a) < \mathcal{R}/(1 - \gamma)$ and that $\mathcal{R} \geq 1$.

Given $\epsilon > 0$, choose s such that

$$\gamma^s \frac{\mathcal{R}}{1 - \gamma} < \frac{\epsilon}{6}.$$

Q-learning

Linear TD(λ) has been proved to converge in the on-policy case if the step-size parameter is reduced over time according to the usual conditions (2.7). Just as discussed in Section 9.4, convergence is not to the minimum-error weight vector, but to a nearby weight vector that depends on λ. The bound on solution quality presented in that section (9.14) can now be generalized to apply for any λ. For the continuing discounted case,

$$\overline{\mathrm{VE}}(\mathbf{w}_\infty) \quad \leq \quad \frac{1 - \gamma\lambda}{1 - \gamma} \min_{\mathbf{w}} \overline{\mathrm{VE}}(\mathbf{w}). \tag{12.8}$$

Linear TD(λ)

That is, the asymptotic error is no more than $\frac{1-\gamma\lambda}{1-\gamma}$ times the smallest possible error. As λ approaches 1, the bound approaches the minimum erro
In practice, however, $\lambda = 1$ is often the poorest choice, a
Figure 12.14.

Proof: Define the operator L on Q-value functions as

$$(LQ)(s, a) = R(s, a) + \gamma \sum_{s' \in S} P^a_{ss'} \bigotimes_{a'} Q(s', a'),$$

Sarsa

for all $(s, a) \in S \times A$. We can rewrite Eq. (C.1) as $Q(s, a) = (LQ)(s, a)$, which has a unique solution if L is contraction with respect to the max norm.

To see that L is a contraction, consider two Q-value functions Q and Q'. We have $|LQ - LQ'| \leq \gamma \max_{s'} |\bigotimes_{a'} Q(s', a') - \bigotimes_{a'} Q'(s', a')| < |Q - Q'|$, where we have used Lemma 5, the fact that $\gamma < 1$, and the non-expansion property of $\bigotimes$. $\square$

Why do we love discounting?

- Mathematical convenience

Assume, without loss of generality, that $\mathcal{Q}_0(x, a) < \mathcal{R}/(1 - \gamma)$ and that $\mathcal{R} \geq 1$.

Given $\epsilon > 0$, choose s such that

$$\gamma^s \frac{\mathcal{R}}{1 - \gamma} < \frac{\epsilon}{6}.$$

Q-learning

Linear TD(λ) has been proved to converge in the on-policy case if the step-size parameter is reduced over time according to the usual conditions (2.7). Just as discussed in Section 9.4, convergence is not to the minimum-error weight vector, but to a nearby weight vector that depends on λ. The bound on solution quality presented in that section (9.14) can now be generalized to apply for any λ. For the continuing discounted case,

$$\overline{\mathrm{VE}}(\mathbf{w}_\infty) \;\leq\; \frac{1 - \gamma\lambda}{1 - \gamma} \min_{\mathbf{w}} \overline{\mathrm{VE}}(\mathbf{w}). \tag{12.8}$$

Linear TD(λ)

That is, the asymptotic error is no more than $\frac{1-\gamma\lambda}{1-\gamma}$ times the smallest possible error. As λ approaches 1, the bound approaches the minimum erro
In practice, however, $\lambda = 1$ is often the poorest choice, a
Figure 12.14.

Proof: Define the operator L on Q-value functions as

$$(LQ)(s, a) = R(s, a) + \gamma \sum_{s' \in S} P^a_{ss'} \bigotimes_{a'} Q(s', a'),$$

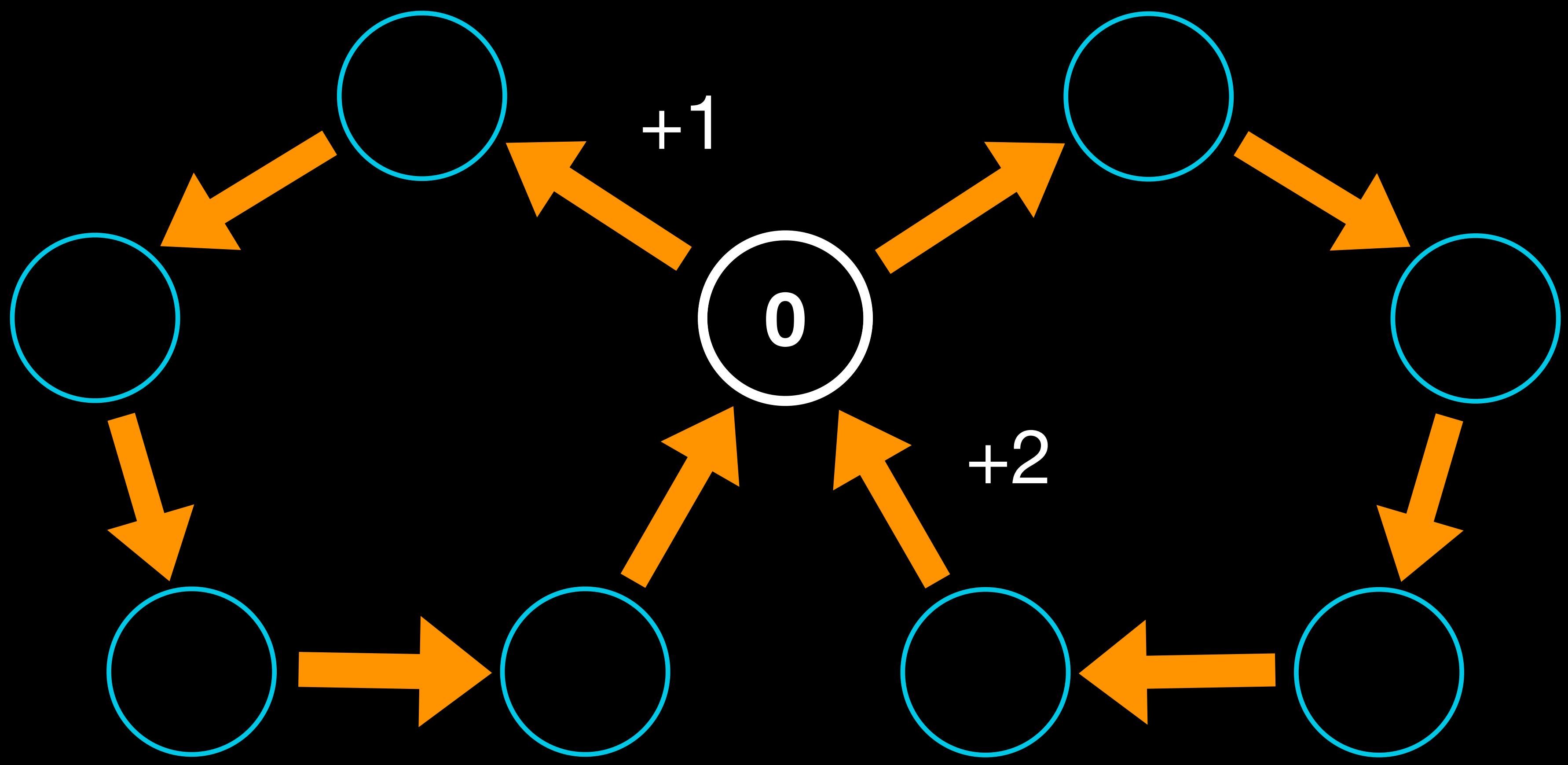
Sarsa

for all $(s, a) \in S \times A$. We can rewrite Eq. (C.1) as $Q(s, a) = (LQ)(s, a)$, which has a unique solution if L is contraction with respect to the max norm.

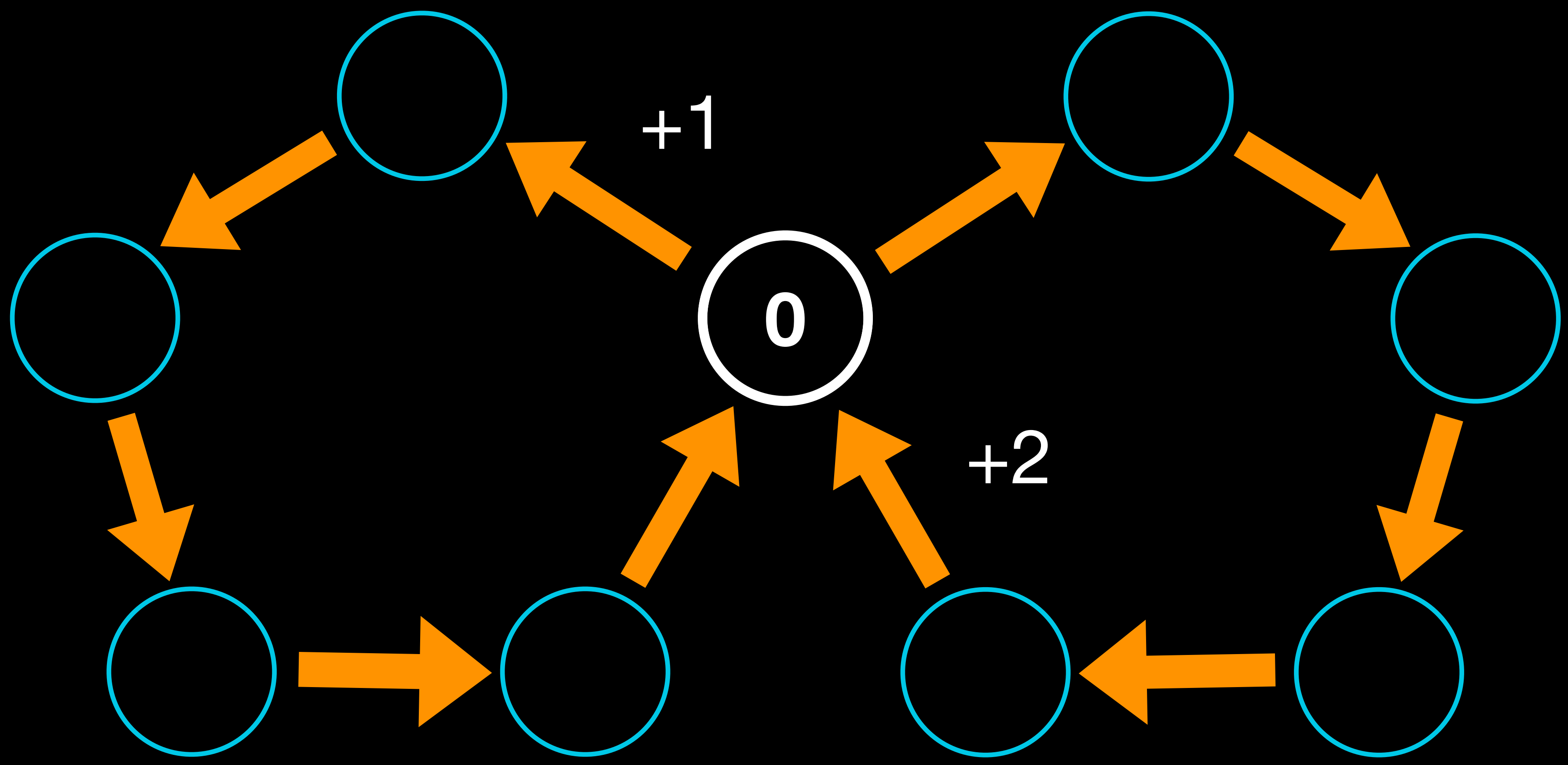
To see that L is a contraction, consider two Q-value functions Q and Q'. We have $|LQ - LQ'| \leq \gamma \max_{s'} |\bigotimes_{a'} Q(s', a') - \bigotimes_{a'} Q'(s', a')| < |Q - Q'|$, where we have used Lemma 5, the fact that $\gamma < 1$, and the non-expansion property of $\bigotimes$. $\square$

The side-effects of discounting

The side-effects of discounting



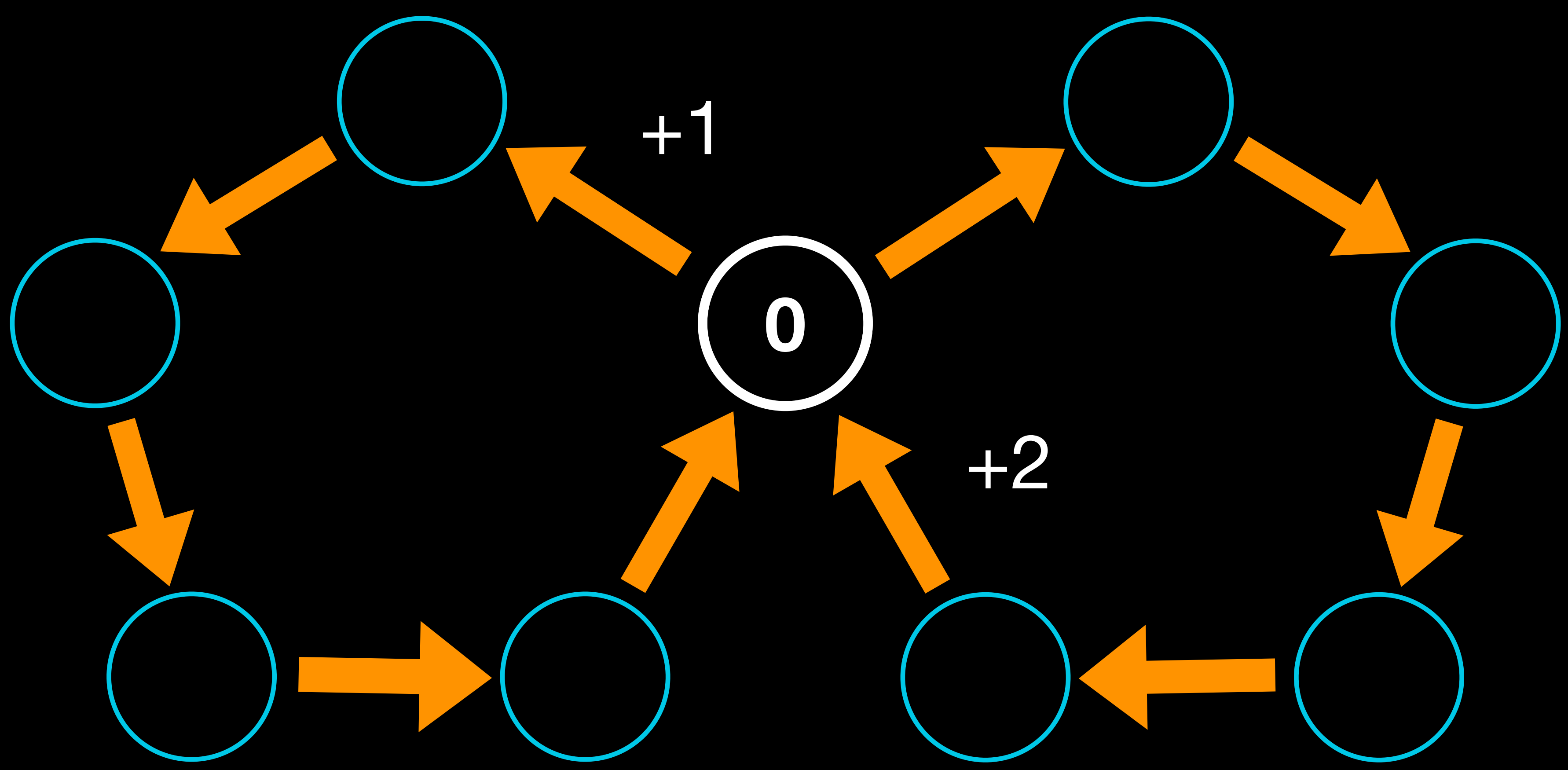
The side-effects of discounting



$$v_L^\gamma(S) = \frac{1}{1-\gamma^5}$$

$$v_R^\gamma(S) = \frac{2\gamma^4}{1-\gamma^5}$$

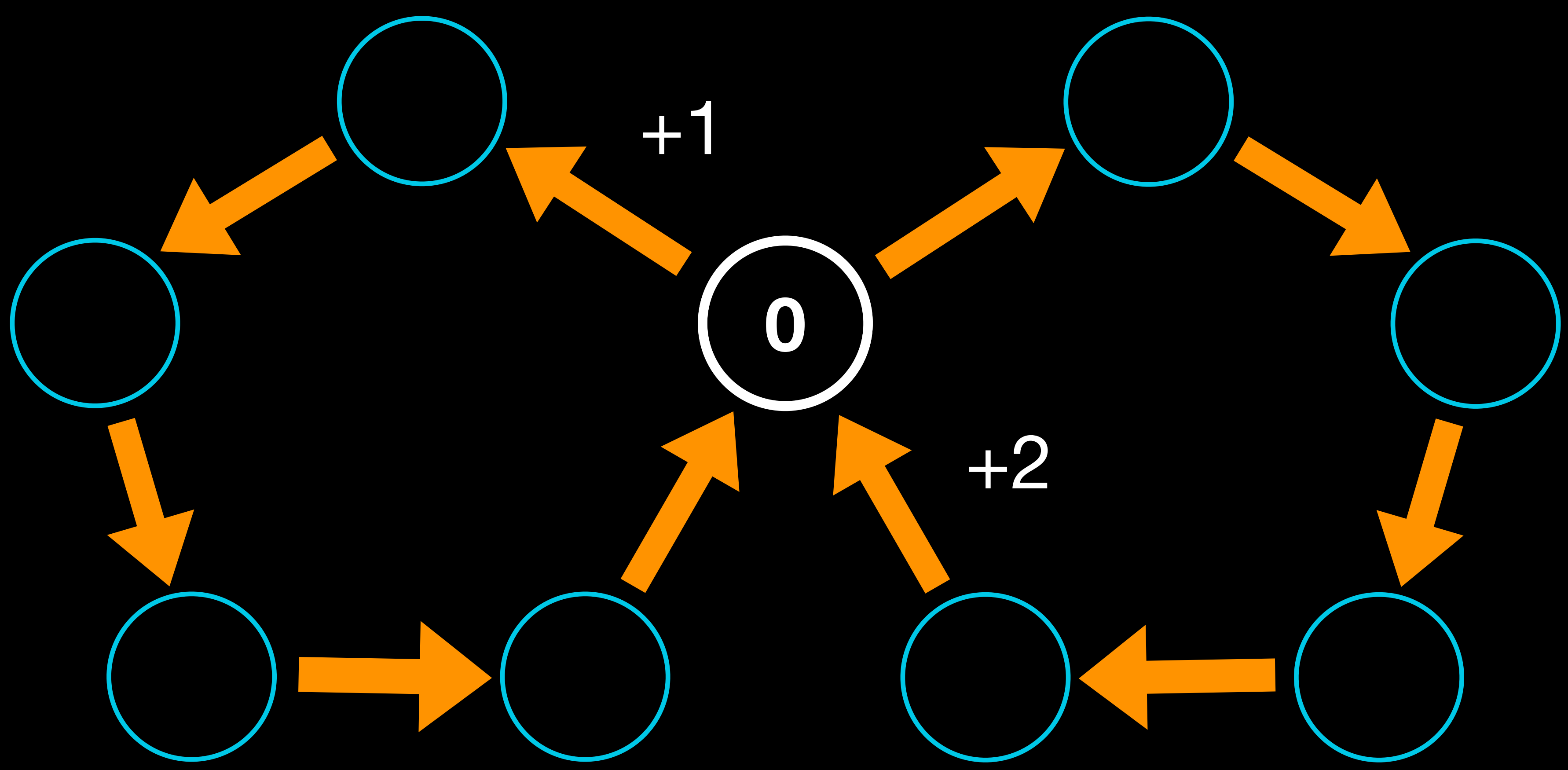
The side-effects of discounting



γ	v_L^γ	v_R^γ
0.5	<u>**1**</u>	0.13
0.9	2.44	<u>**3.20**</u>

$$v_L^\gamma(S) = \frac{1}{1-\gamma^5} \qquad v_R^\gamma(S) = \frac{2\gamma^4}{1-\gamma^5}$$

The side-effects of discounting



γ	v_L^γ	v_R^γ
0.5	**<u>1</u>**	0.13
0.9	2.44	**<u>3.20</u>**

$$v_R^\gamma(S) > v_L^\gamma(S)$$

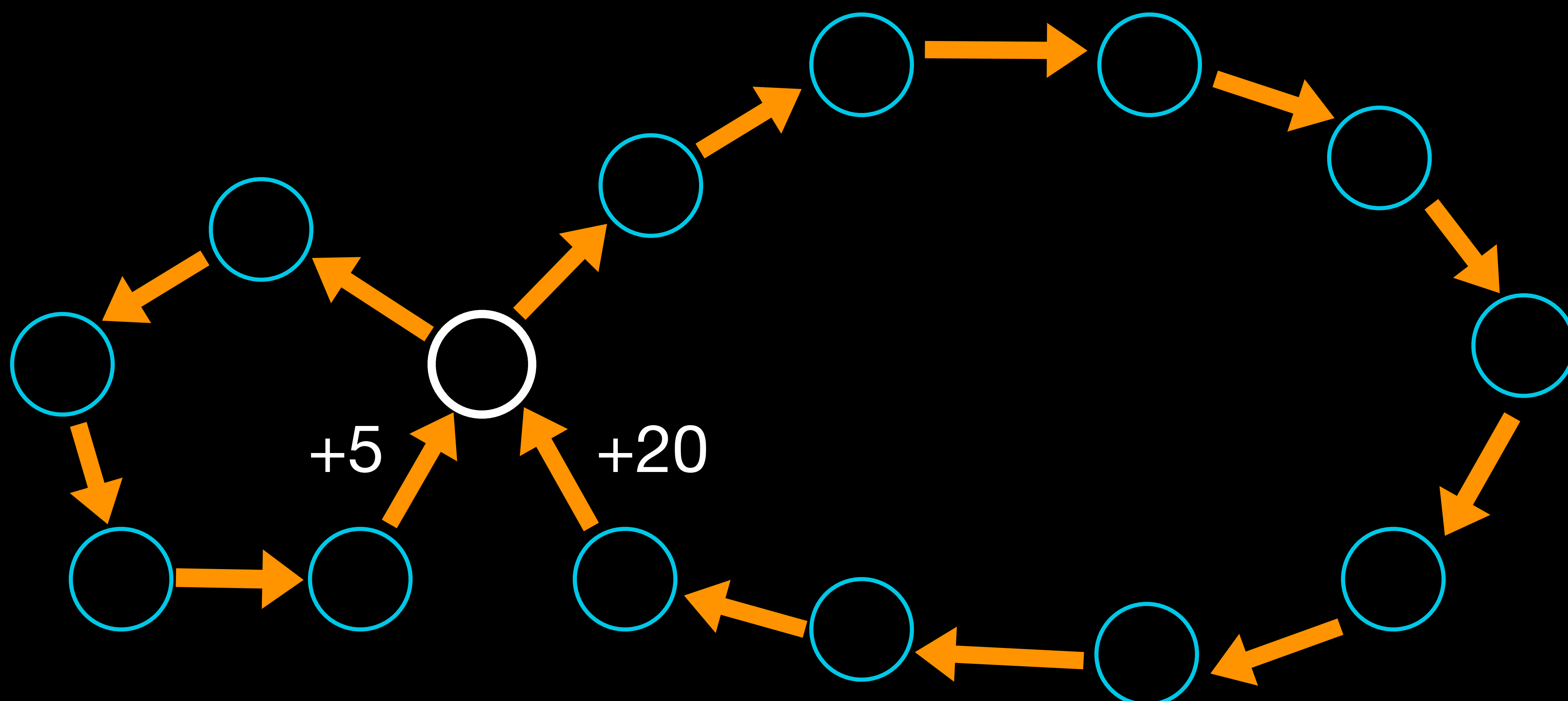
when

$$\gamma > 0.84$$

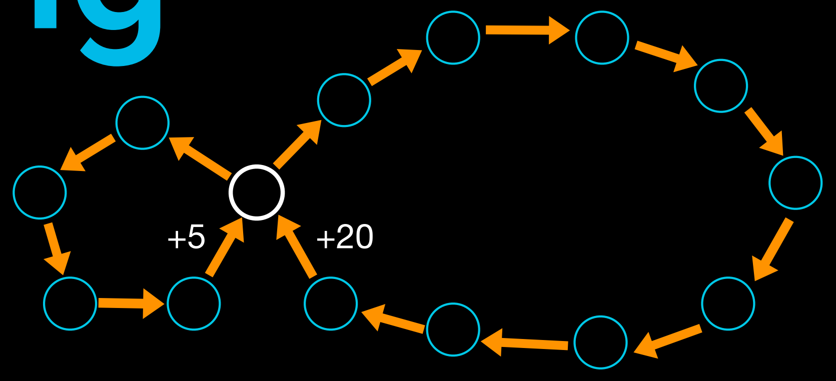
$$v_L^\gamma(S) = \frac{1}{1-\gamma^5}$$

$$v_R^\gamma(S) = \frac{2\gamma^4}{1-\gamma^5}$$

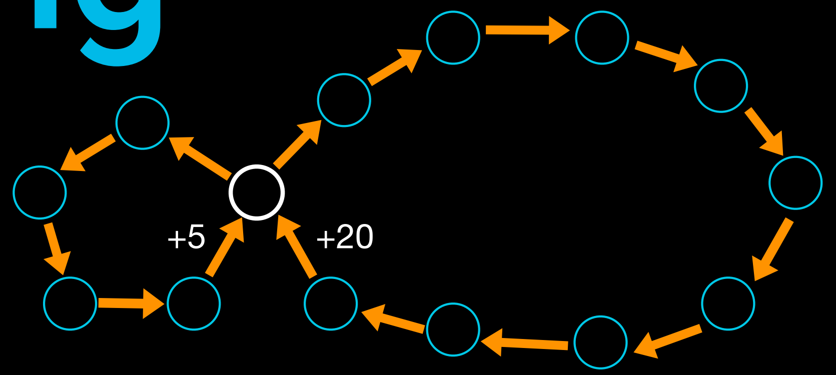
The side-effects of discounting

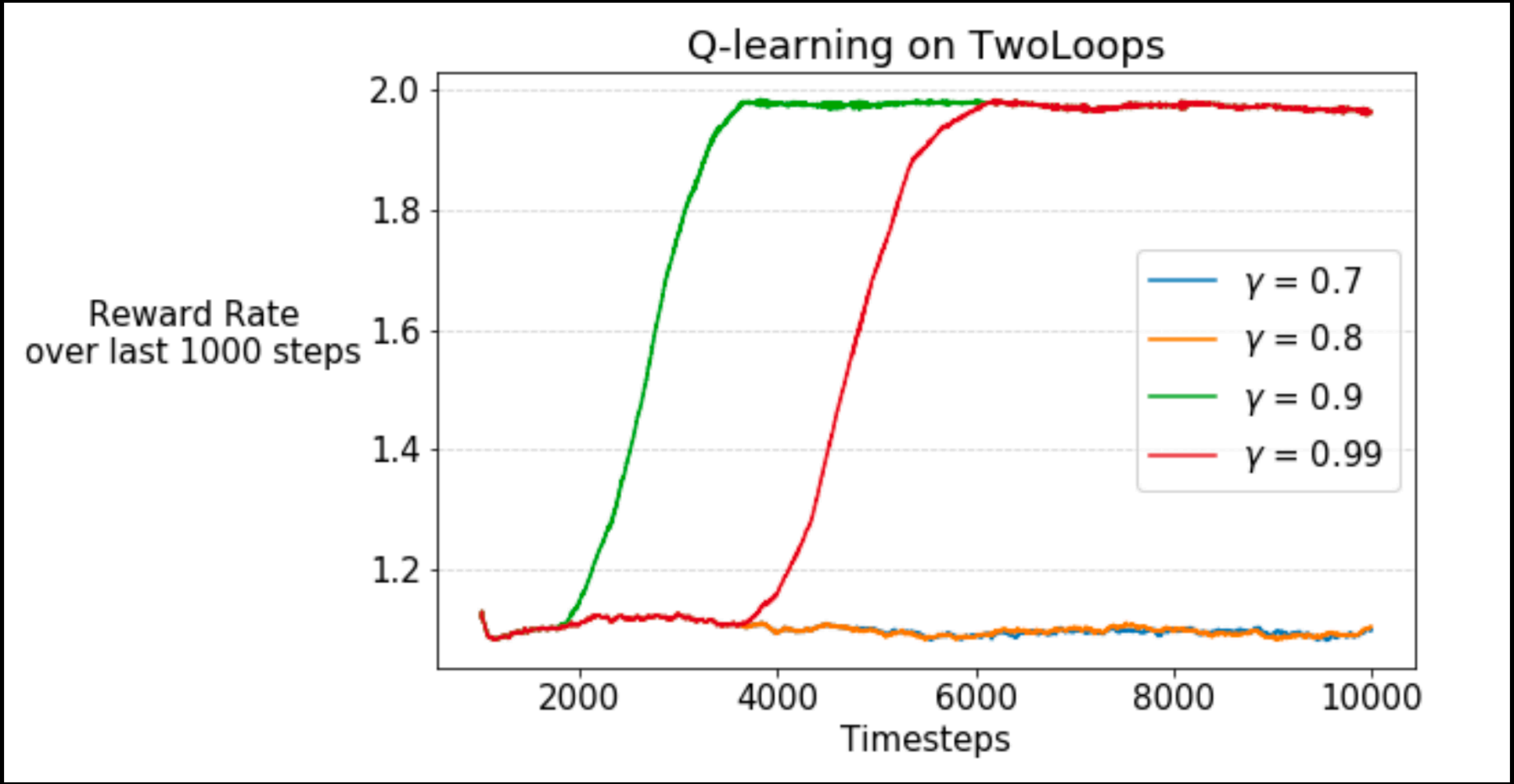


The side-effects of discounting



The side-effects of discounting





But the book says
the discount factor doesn't matter, right?

The Futility of Discounting in Continuing Problems

Perhaps discounting can be saved by choosing an objective that sums discounted values over the distribution with which states occur under the policy:

$$
\begin{aligned}
J(\pi) &= \sum_s \mu_\pi(s) v_\pi^\gamma(s) && \text{(where } v_\pi^\gamma \text{ is the discounted value function)} \\
&= \sum_s \mu_\pi(s) \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) \left[r + \gamma v_\pi^\gamma(s') \right] && \text{(Bellman Eq.)} \\
&= r(\pi) + \sum_s \mu_\pi(s) \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) \gamma v_\pi^\gamma(s') && \text{(from (10.7))} \\
&= r(\pi) + \gamma \sum_{s'} v_\pi^\gamma(s') \sum_s \mu_\pi(s) \sum_a \pi(a|s) p(s'|s, a) && \text{(from (3.4))} \\
&= r(\pi) + \gamma \sum_{s'} v_\pi^\gamma(s') \mu_\pi(s') && \text{(from (10.8))} \\
&= r(\pi) + \gamma J(\pi) \\
&= r(\pi) + \gamma r(\pi) + \gamma^2 J(\pi) \\
&= r(\pi) + \gamma r(\pi) + \gamma^2 r(\pi) + \gamma^3 r(\pi) + \cdots \\
&= \frac{1}{1-\gamma} r(\pi).
\end{aligned}
$$

The proposed discounted objective orders policies identically to the undiscounted (average reward) objective. The discount rate γ does not influence the ordering!

But the book says the discount factor doesn't matter, right?

The Futility of Discounting in Continuing Problems

Perhaps discounting can be saved by choosing an objective that sums discounted values over the distribution with which states occur under the policy:

$$
\begin{aligned}
\boxed{J(\pi) = \sum_s \mu_\pi(s) v_\pi^\gamma(s)} & & \text{(where } v_\pi^\gamma \text{ is the discounted value function)} \\
&= \sum_s \mu_\pi(s) \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) \left[r + \gamma v_\pi^\gamma(s')\right] & \text{(Bellman Eq.)} \\
&= r(\pi) + \sum_s \mu_\pi(s) \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) \gamma v_\pi^\gamma(s') & \text{(from (10.7))} \\
&= r(\pi) + \gamma \sum_{s'} v_\pi^\gamma(s') \sum_s \mu_\pi(s) \sum_a \pi(a|s) p(s'|s, a) & \text{(from (3.4))} \\
&= r(\pi) + \gamma \sum_{s'} v_\pi^\gamma(s') \mu_\pi(s') & \text{(from (10.8))} \\
&= r(\pi) + \gamma J(\pi) \\
&= r(\pi) + \gamma r(\pi) + \gamma^2 J(\pi) \\
&= r(\pi) + \gamma r(\pi) + \gamma^2 r(\pi) + \gamma^3 r(\pi) + \cdots \\
&= \frac{1}{1-\gamma} r(\pi).
\end{aligned}
$$

<u>if</u> that is the objective

The proposed discounted objective orders policies identically to the undiscounted (average reward) objective. The discount rate γ does not influence the ordering!

Additionally, problems of function approximation

- Remember, the policy improvement theorem does not hold in the function-approximation setting.
- In the tabular setting, we could compare two policies by a state-wise comparison of the value function.
- In the function-approximation setting, this cannot be done.

Discounting –
does it make sense?

Discounting – does it make sense?

- The choice of discount factor matters.

Discounting – does it make sense?

- The choice of discount factor matters.

- We don't have methods that can feasibly follow a discounted objective in which the discount factor does not matter.

Discounting – does it make sense?

- The choice of discount factor matters.
- We don’t have methods that can feasibly follow a discounted objective in which the discount factor does not matter.
- In the function-approximation setting, we don’t even have a decent way to compare/order policies.

A viable alternative –
The Average Reward Formulation

A viable alternative –
The Average Reward Formulation

- Objective:

A viable alternative –
The Average Reward Formulation

- Objective:

$$J(\theta) \doteq r(\pi) \doteq \lim_{h\to\infty} \frac{1}{h}\mathbb{E}_{\pi}[R_{t+1} + R_{t+2} + \ldots + R_{t+h}]$$

A viable alternative – The Average Reward Formulation

- Objective:

$$J(\theta) \doteq r(\pi) \doteq \lim_{h \to \infty} \frac{1}{h} \mathbb{E}_\pi[R_{t+1} + R_{t+2} + \ldots + R_{t+h}]$$

- Differential return:

A viable alternative –
The Average Reward Formulation

- Objective:

$$J(\theta) \doteq r(\pi) \doteq \lim_{h \to \infty} \frac{1}{h} \mathbb{E}_\pi[R_{t+1} + R_{t+2} + \ldots + R_{t+h}]$$

- Differential return:

$$G_t \doteq R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + \ldots$$

A viable alternative –
The Average Reward Formulation

- Objective:

$$J(\theta) \doteq r(\pi) \doteq \lim_{h\to\infty} \frac{1}{h}\mathbb{E}_\pi[R_{t+1} + R_{t+2} + \ldots + R_{t+h}]$$

- Differential return:

$$G_t \doteq R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + \ldots$$

- Differential value function:

A viable alternative – The Average Reward Formulation

- Objective:

$$J(\theta) \doteq r(\pi) \doteq \lim_{h\to\infty} \frac{1}{h}\mathbb{E}_\pi[R_{t+1} + R_{t+2} + \ldots + R_{t+h}]$$

- Differential return:

$$G_t \doteq R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + \ldots$$

- Differential value function:

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \,|\, S_t = s] \\ &= \sum_a \pi(a\,|\,s) \sum_{s',r} p(s', r\,|\,s, a)\big[r - r(\pi) + v_\pi(s') - v_\pi(s)\big] \end{aligned}$$

A viable alternative – The Average Reward Formulation

- Objective:

$$J(\theta) \doteq r(\pi) \doteq \lim_{h \to \infty} \frac{1}{h} \mathbb{E}_\pi[R_{t+1} + R_{t+2} + \ldots + R_{t+h}]$$

- Differential return:

$$G_t \doteq R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + \ldots$$

- Differential value function:

$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \,|\, S_t = s] \\ &= \sum_a \pi(a \,|\, s) \sum_{s',r} p(s', r \,|\, s, a) \big[r - r(\pi) + v_\pi(s') - v_\pi(s)\big] \end{aligned}$$

A viable alternative – The Average Reward Formulation

- Objective:

$$J(\theta) \doteq r(\pi) \doteq \lim_{h\to\infty} \frac{1}{h} \mathbb{E}_\pi[R_{t+1} + R_{t+2} + \ldots + R_{t+h}]$$

- Differential return:

$$G_t \doteq R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + \ldots$$

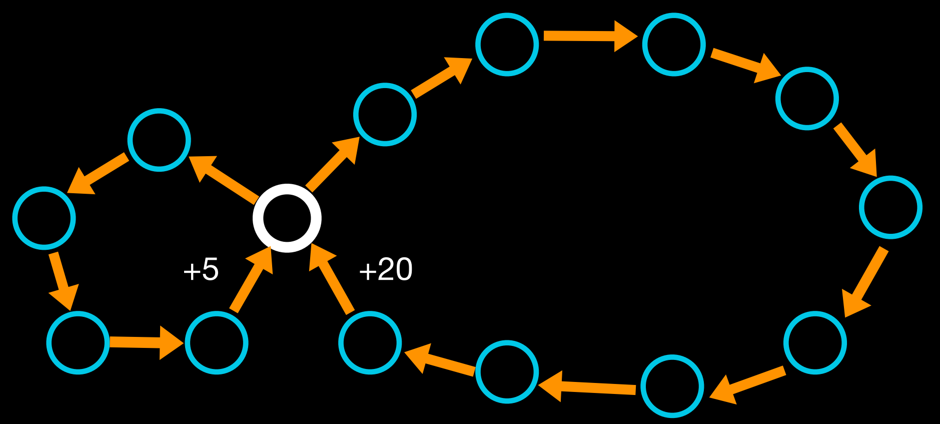
- Differential value function:

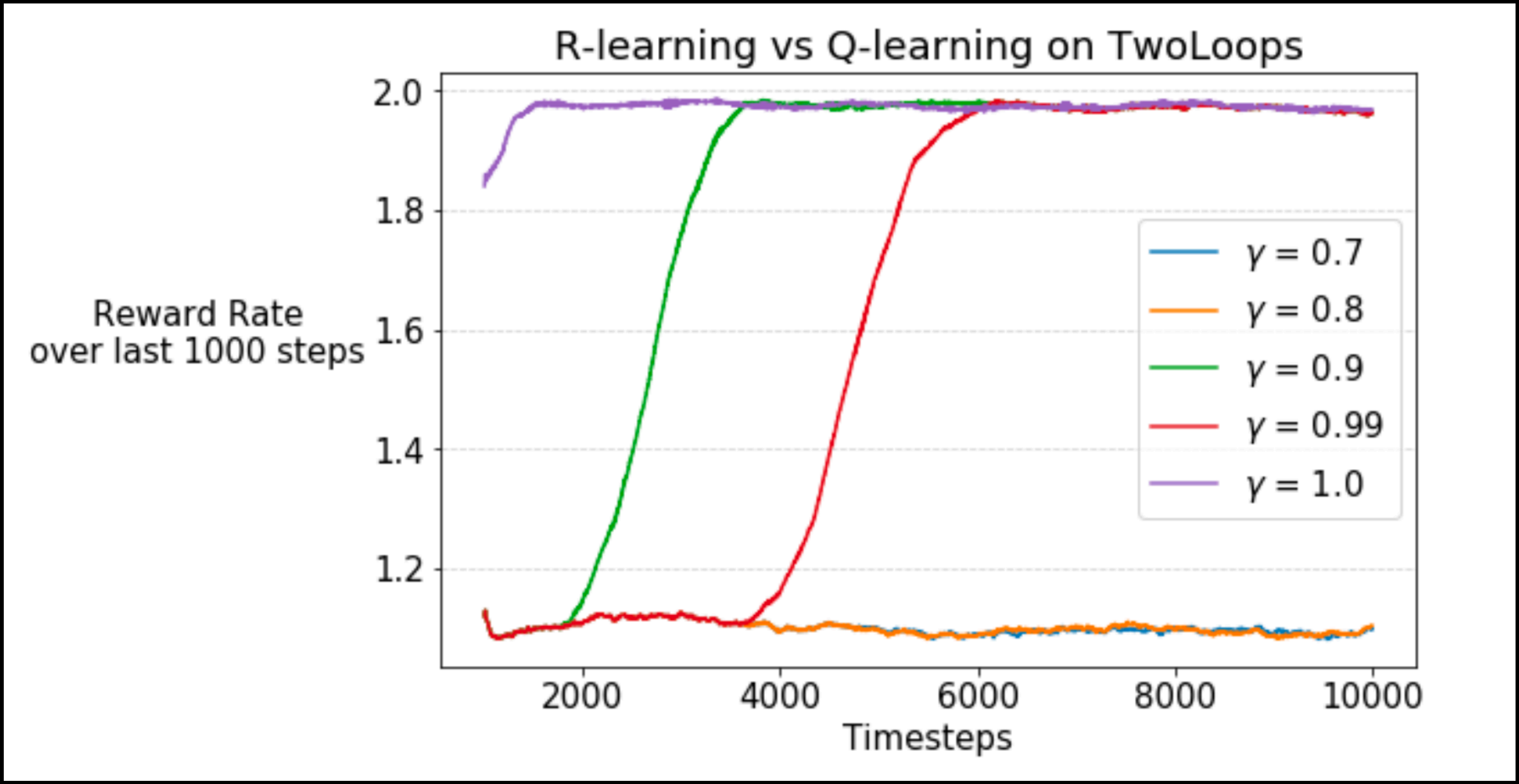
$$\begin{aligned} v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \,|\, S_t = s] \\ &= \sum_a \pi(a \,|\, s) \sum_{s',r} p(s', r \,|\, s, a) \big[r - r(\pi) + v_\pi(s') - v_\pi(s)\big] \end{aligned}$$

Can compare the average reward $r(\pi)$

Does it work?

Does it work?


+5
+20


R-learning vs Q-learning on TwoLoops
Reward Rate
over last 1000 steps
2.0
1.8
1.6
1.4
1.2
2000
4000
6000
8000
10000
Timesteps
γ = 0.7
γ = 0.8
γ = 0.9
γ = 0.99
γ = 1.0

Challenges
(i.e., Opportunities)

Challenges
(i.e., Opportunities)

- Under-studied!
  - Very few algorithms, with no comprehensive studies of the strengths/weaknesses, assumptions, etc.
  - Unclear how to perform planning, off-policy learning, use options, etc…

Challenges
(i.e., Opportunities)

- Under-studied!
  - Very few algorithms, with no comprehensive studies of the strengths/weaknesses, assumptions, etc.
  - Unclear how to perform planning, off-policy learning, use options, etc…
- No suite of domains
  - Need to build one for testing our algorithms systematically.

Takeaways

- In continuing problems, discounting doesn’t make sense.
- The Average Reward formulation seems to be a viable alternative, with so many open problems!

Thank you!

(More) Questions?

Stretch slides

What are some interesting continuing domains?

- Inventory control
- Clinical trials
- Robot navigation
- Access control / queuing systems
    - Job scheduling, Packet routing

But discounting works, right…

- In *episodic* domains where actions don't really have long-term effects.
- For Chess and Go, AlphaGo did *not* use discounting.

Abstract

In continuing problems, a discount factor is commonly used to ensure that the potentially-infinite return per state is a finite number. In this talk, we will discuss how this problem setting is problematic, and how the average reward formulation is a viable alternative.