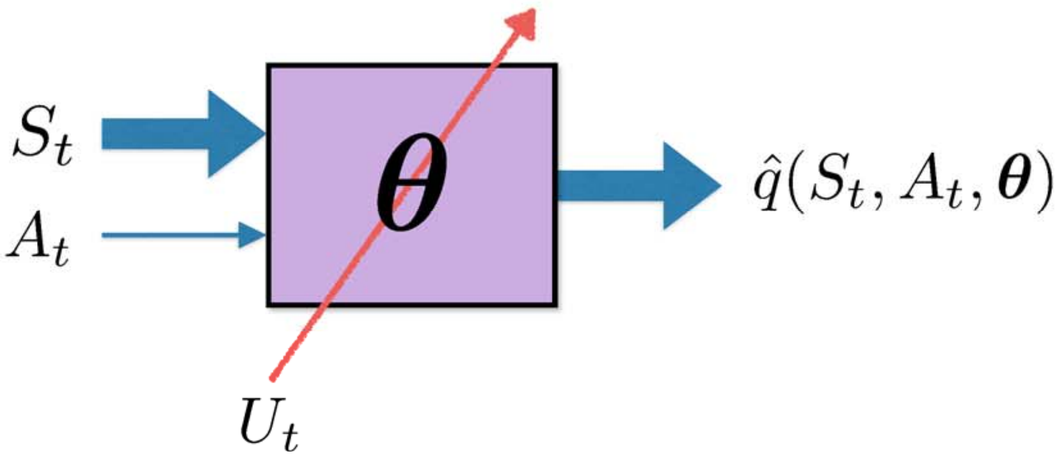
Class 5: On Policy Control With Approximation Chapter 10

Sutton slides/silver slides

What we learned last time

- Value-function approximation by stochastic gradient descent enables RL to be applied to arbitrarily large state spaces
- Most algorithms just carry over the Targets from the tabular case
- With bootstrapping (TD), we don’t get true gradient descent methods
  - but the linear, on-policy case is still guaranteed convergent
  - and learning is *faster* with *n*-step methods (*n*>1), as before
- For continuous state spaces, coarse/tile coding is a good strategy

Value function approximation (VFA) replaces the table with a general parameterized form



On-policy Control with Approximation

- (Semi-)gradient methods carry over to control in the usual way
    - Mountain Car example
- *n*-step methods carry over too, with the usual tradeoffs
- A new *average-reward* setting,
  with *differential* value functions and *differential* algorithms
    - Queuing example (tabular)
- The discounting setting is deprecated

(Semi-)gradient methods carry over to control in the usual on-policy GPI way

- Always learn the action-value function of the current policy
- Always act near-greedily wrt the current action-value estimates
- The learning rule is the same as in Chapter 9:

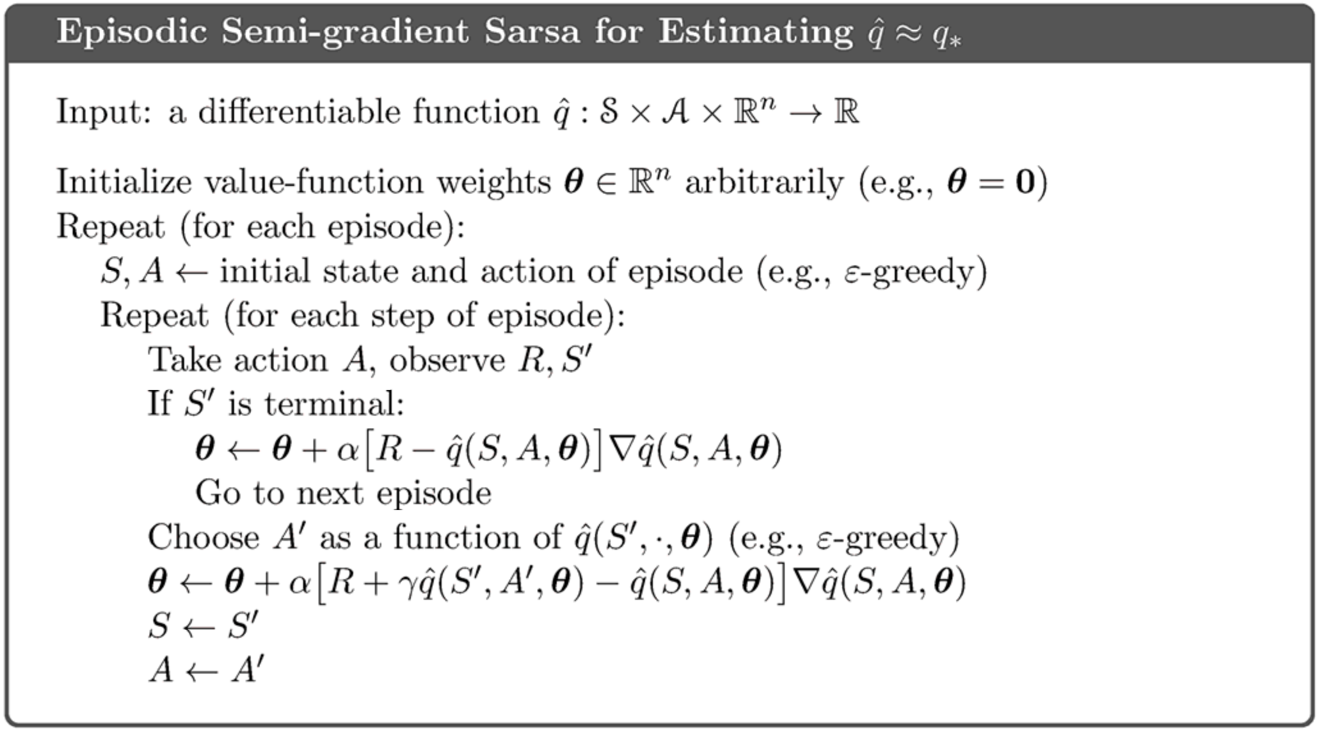
$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha \Big[U_t - \hat{q}(S_t, A_t, \boldsymbol{\theta}_t) \Big] \nabla \hat{q}(S_t, A_t, \boldsymbol{\theta}_t)$$

update target, e.g., $U_t = G_t$ (MC) $U_t = R_{t+1} + \gamma \hat{q}(S_{t+1}, A_{t+1}, \boldsymbol{\theta}_t)$ (Sarsa)

(Expected Sarsa) $U_t = R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) \hat{q}(S_{t+1}, a, \boldsymbol{\theta}_t)$ $U_t = \sum_{s',r} p(s', r|S_t, A_t) \Big[r + \gamma \sum_{a'} \pi(a'|s') \hat{q}(s', a', \boldsymbol{\theta}_t) \Big]$ (DP)

(Semi-)gradient methods carry over to control

$$\boldsymbol{\theta}_{t+1} \doteq \boldsymbol{\theta}_t + \alpha\Big[U_t - \hat{q}(S_t, A_t, \boldsymbol{\theta}_t)\Big]\nabla\hat{q}(S_t, A_t, \boldsymbol{\theta}_t)$$



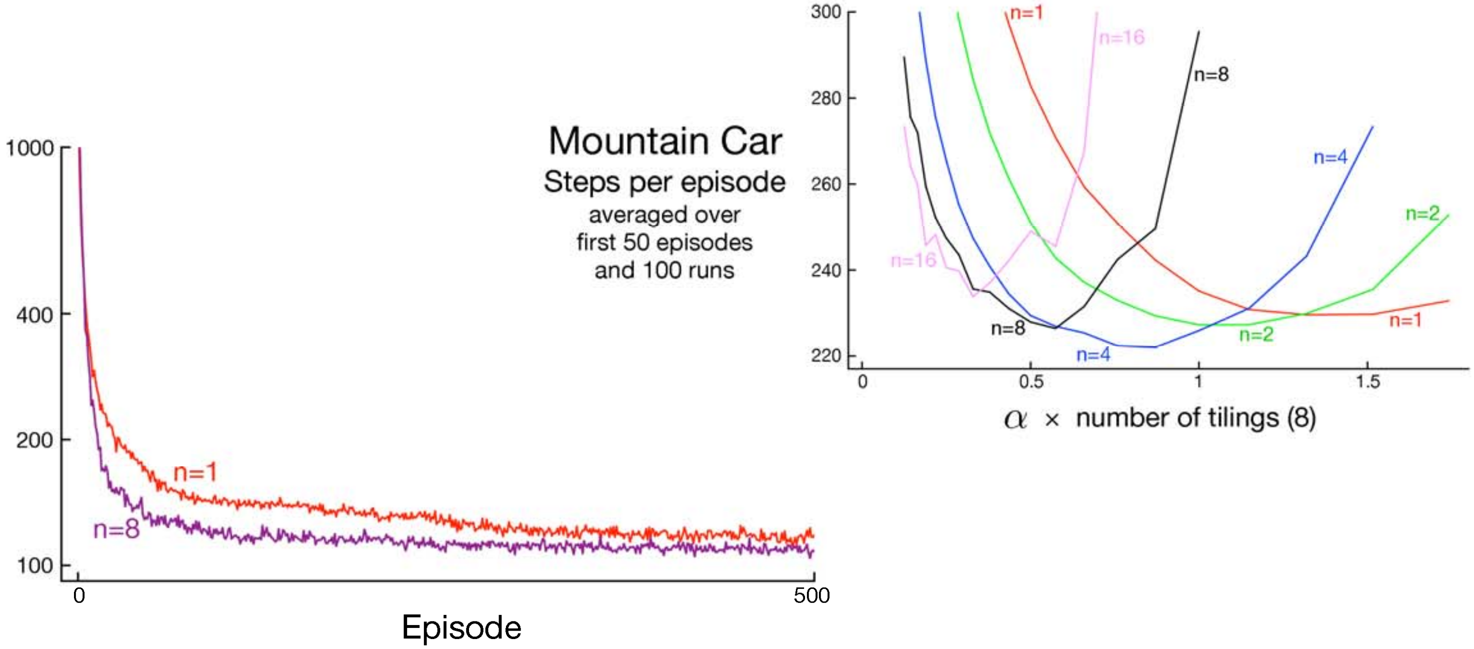
Episodic Semi-gradient Sarsa for Estimating $\hat{q} \approx q_*$

```
Input: a differentiable function $\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^n \to \mathbb{R}$

Initialize value-function weights $\boldsymbol{\theta} \in \mathbb{R}^n$ arbitrarily (e.g., $\boldsymbol{\theta} = \mathbf{0}$)
Repeat (for each episode):
    $S, A \leftarrow$ initial state and action of episode (e.g., $\varepsilon$-greedy)
    Repeat (for each step of episode):
        Take action $A$, observe $R, S'$
        If $S'$ is terminal:
            $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\big[R - \hat{q}(S, A, \boldsymbol{\theta})\big]\nabla\hat{q}(S, A, \boldsymbol{\theta})$
            Go to next episode
        Choose $A'$ as a function of $\hat{q}(S', \cdot, \boldsymbol{\theta})$ (e.g., $\varepsilon$-greedy)
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\big[R + \gamma\hat{q}(S', A', \boldsymbol{\theta}) - \hat{q}(S, A, \boldsymbol{\theta})\big]\nabla\hat{q}(S, A, \boldsymbol{\theta})$
        $S \leftarrow S'$
        $A \leftarrow A'$
```

n-step semi-gradient Sarsa is better for *n*>1

$$\boldsymbol{\theta}_{t+n} \doteq \boldsymbol{\theta}_{t+n-1} + \alpha \left[G_t^{(n)} - \hat{q}(S_t, A_t, \boldsymbol{\theta}_{t+n-1}) \right] \nabla \hat{q}(S_t, A_t, \boldsymbol{\theta}_{t+n-1}), \qquad 0 \leq t < T$$



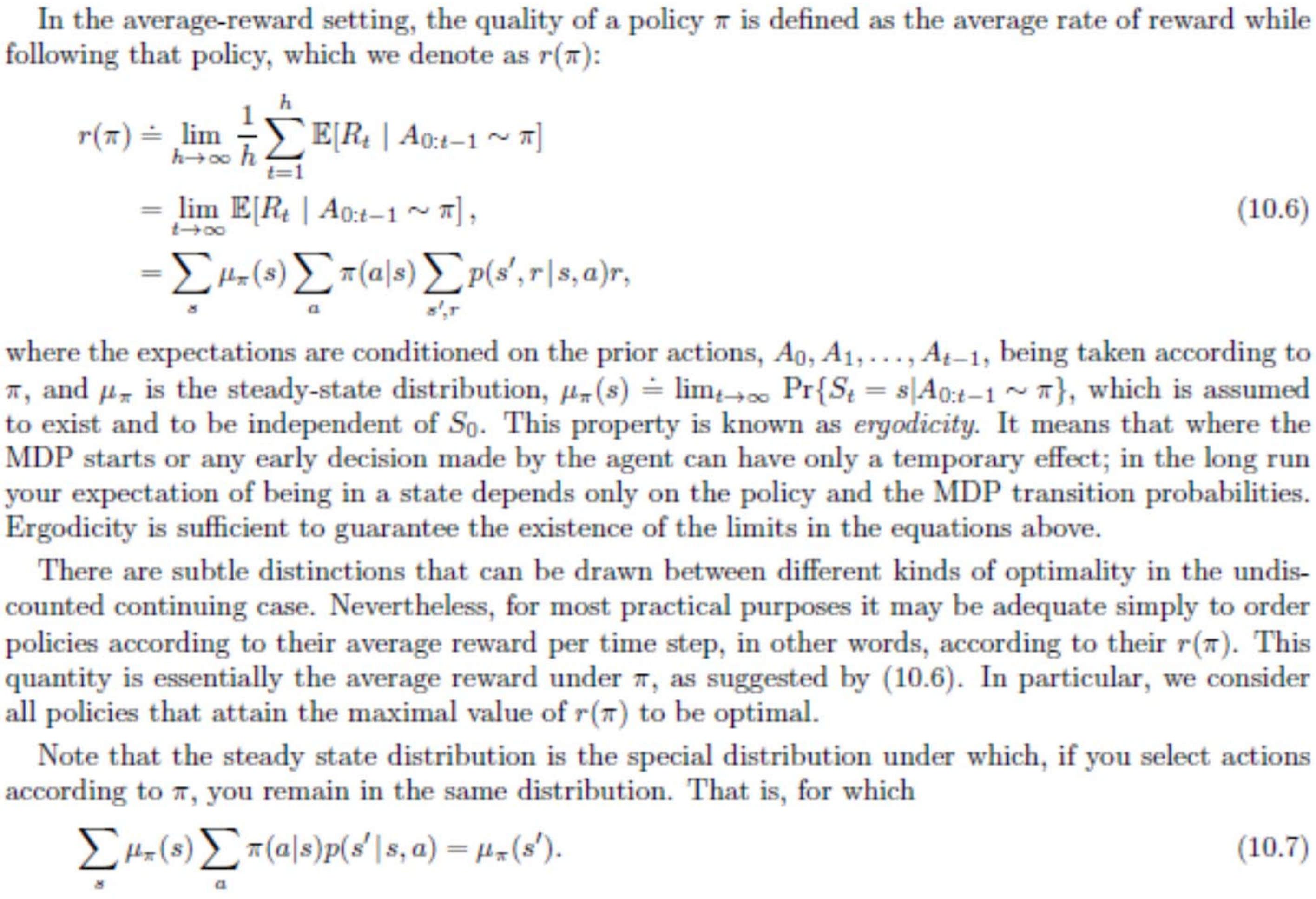
Mountain Car
Steps per episode
log scale
averaged over 100 runs

Mountain Car
Steps per episode
averaged over
first 50 episodes
and 100 runs

On-policy Control with Approximation

- (Semi-)gradient methods carry over to control in the usual way
  - Mountain Car example
- *n*-step methods carry over too, with the usual tradeoffs
- A new average-reward setting,
  with differential value functions and differential algorithms
  - Queuing example (tabular)
- The discounting setting is deprecated

Average reward: A new problem setting for continuing tasks



In the average-reward setting, the quality of a policy π is defined as the average rate of reward while following that policy, which we denote as $r(\pi)$:

$$\begin{aligned} r(\pi) &\doteq \lim_{h\to\infty} \frac{1}{h} \sum_{t=1}^{h} \mathbb{E}[R_t \mid A_{0:t-1} \sim \pi] \\ &= \lim_{t\to\infty} \mathbb{E}[R_t \mid A_{0:t-1} \sim \pi]\,, \qquad (10.6) \\ &= \sum_s \mu_\pi(s) \sum_a \pi(a|s) \sum_{s',r} p(s',r\,|\,s,a) r, \end{aligned}$$

where the expectations are conditioned on the prior actions, $A_0, A_1, \ldots, A_{t-1}$, being taken according to π, and μ_π is the steady-state distribution, $\mu_\pi(s) \doteq \lim_{t\to\infty} \Pr\{S_t = s|A_{0:t-1} \sim \pi\}$, which is assumed to exist and to be independent of S_0. This property is known as *ergodicity*. It means that where the MDP starts or any early decision made by the agent can have only a temporary effect; in the long run your expectation of being in a state depends only on the policy and the MDP transition probabilities. Ergodicity is sufficient to guarantee the existence of the limits in the equations above.

There are subtle distinctions that can be drawn between different kinds of optimality in the undiscounted continuing case. Nevertheless, for most practical purposes it may be adequate simply to order policies according to their average reward per time step, in other words, according to their $r(\pi)$. This quantity is essentially the average reward under π, as suggested by (10.6). In particular, we consider all policies that attain the maximal value of $r(\pi)$ to be optimal.

Note that the steady state distribution is the special distribution under which, if you select actions according to π, you remain in the same distribution. That is, for which

$$\sum_s \mu_\pi(s) \sum_a \pi(a|s) p(s'\,|\,s,a) = \mu_\pi(s'). \qquad (10.7)$$

A new goal for continuing tasks: Maximizing average reward per time step

$$\text{Maximize } \eta(\pi) \doteq \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[R_t \mid A_{0:t-1} \sim \pi]$$
$$= \lim_{t\to\infty} \mathbb{E}[R_t \mid A_{0:t-1} \sim \pi] \,,$$
$$= \sum_s d_\pi(s) \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) r$$

assuming that these limits exist
is known as the *ergodicity* property

$d_\pi : \mathcal{S} \to [0, 1]$ is the steady-state distribution under π,
also known as the on-policy distribution:

$$d_\pi(s) \doteq \lim_{t\to\infty} \Pr\{S_t = s|A_{0:t-1} \sim \pi\}$$

$\eta(\pi)$ is the *average* amount of *reward* received per time step

The Differential Return

In the average reward setting, everything is new

$\cancel{\gamma}$

- Returns: $G_t \doteq R_{t+1} - \eta(\pi) + R_{t+2} - \eta(\pi) + R_{t+3} - \eta(\pi) + \cdots$

- Bellman Eqs:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{r,s'} p(s',r|s,a)\Big[r - \eta(\pi) + v_\pi(s')\Big],$$

$$q_\pi(s,a) = \sum_{r,s'} p(s',r|s,a)\Big[r - \eta(\pi) + \sum_{a'} \pi(a'|s')q_\pi(s',a')\Big],$$

prediction

$$v_*(s) = \max_a \sum_{r,s'} p(s',r|s,a)\Big[r - \eta(\pi) + v_*(s')\Big], \text{ and}$$

$$q_*(s,a) = \sum_{r,s'} p(s',r|s,a)\Big[r - \eta(\pi) + \max_{a'} q_*(s',a')\Big]$$

control

- Update targets:

$$U_t \doteq R_{t+1} - \bar{R}_t + \hat{q}(S_{t+1}, A_{t+1}, \boldsymbol{\theta}) \text{ or } U_t \doteq R_{t+1} - \bar{R}_t + \hat{v}(S_{t+1}, \boldsymbol{\theta})$$

estimate of $\eta(\pi)$

Differential semi-gradient Sarsa for estimating $\hat{q} \approx q_*$

```
Input: a differentiable function $\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^n \rightarrow \mathbb{R}$
Parameters: step sizes $\alpha, \beta > 0$

Initialize value-function weights $\boldsymbol{\theta} \in \mathbb{R}^n$ arbitrarily (e.g., $\boldsymbol{\theta} = \mathbf{0}$)
Initialize average reward estimate $\bar{R}$ arbitrarily (e.g., $\bar{R} = 0$)
Initialize state $S$, and action $A$

Repeat (for each step):
    Take action $A$, observe $R, S'$
    Choose $A'$ as a function of $\hat{q}(S', \cdot, \boldsymbol{\theta})$ (e.g., $\varepsilon$-greedy)
    $\delta \leftarrow R - \bar{R} + \hat{q}(S', A', \boldsymbol{\theta}) - \hat{q}(S, A, \boldsymbol{\theta})$
    $\bar{R} \leftarrow \bar{R} + \beta\delta$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\delta\nabla\hat{q}(S, A, \boldsymbol{\theta})$
    $S \leftarrow S'$
    $A \leftarrow A'$
```

Conclusions

- Control is straightforward in the on-policy, episodic, linear case
- For the continuing case, we need the average-reward setting
  - which is a lot like just replacing R_t with $R_t - \eta(\pi)$ everywhere
  - where $\eta(\pi)$ is the average reward per step, or its estimate
- We should probably never use discounting as a control objective
- Formal results (bounds) exist for the linear, on-policy case
  - we get chattering near a good solution, not convergence