On-Policy Control with Approximation

명승주 201621114

Intro

- Control Problem – $\hat{q}(s, a, w) \approx q_*(s, a), \;\; w \in \mathbb{R}^d$ finite dimensional weight vector
- Episodic Task : Easy to $\hat{v} \rightarrow \hat{q}$
- Continuing Task : Not Easy to $\hat{v} \rightarrow \hat{q}$ – cuz No Ending, No Terminal Reward!

  have to define new discounting way to get an optimal policy

  – To Get Genuine function approximation

  – Give up Discounting! -> new “Average-reward” formulation
  – with new “Differential” value function

Episodic Semi-Gradient Control

- Minimise MSE between $\hat{q}$ and q_π

$$J(\mathbf{w}) = \mathbb{E}_\pi\left[(q_\pi(S,A) - \hat{q}(S,A,\mathbf{w}))^2\right]$$

- Use Stochastic Gradient Descent to find local minimum
  -> update W to change direction

$$-\frac{1}{2}\nabla_\mathbf{w} J(\mathbf{w}) = (q_\pi(S,A) - \hat{q}(S,A,\mathbf{w}))\nabla_\mathbf{w}\hat{q}(S,A,\mathbf{w})$$

$$\Delta\mathbf{w} = \alpha(q_\pi(S,A) - \hat{q}(S,A,\mathbf{w}))\nabla_\mathbf{w}\hat{q}(S,A,\mathbf{w})$$

- EX – Episodic Semi-Gradient One-Step Sarsa

$$\Delta\mathbf{w} = \alpha(R_{t+1} + \gamma\hat{q}(S_{t+1}, A_{t+1}, \mathbf{w}) - \hat{q}(S_t, A_t, \mathbf{w}))\nabla_\mathbf{w}\hat{q}(S_t, A_t, \mathbf{w})$$

Episodic Semi-Gradient Control

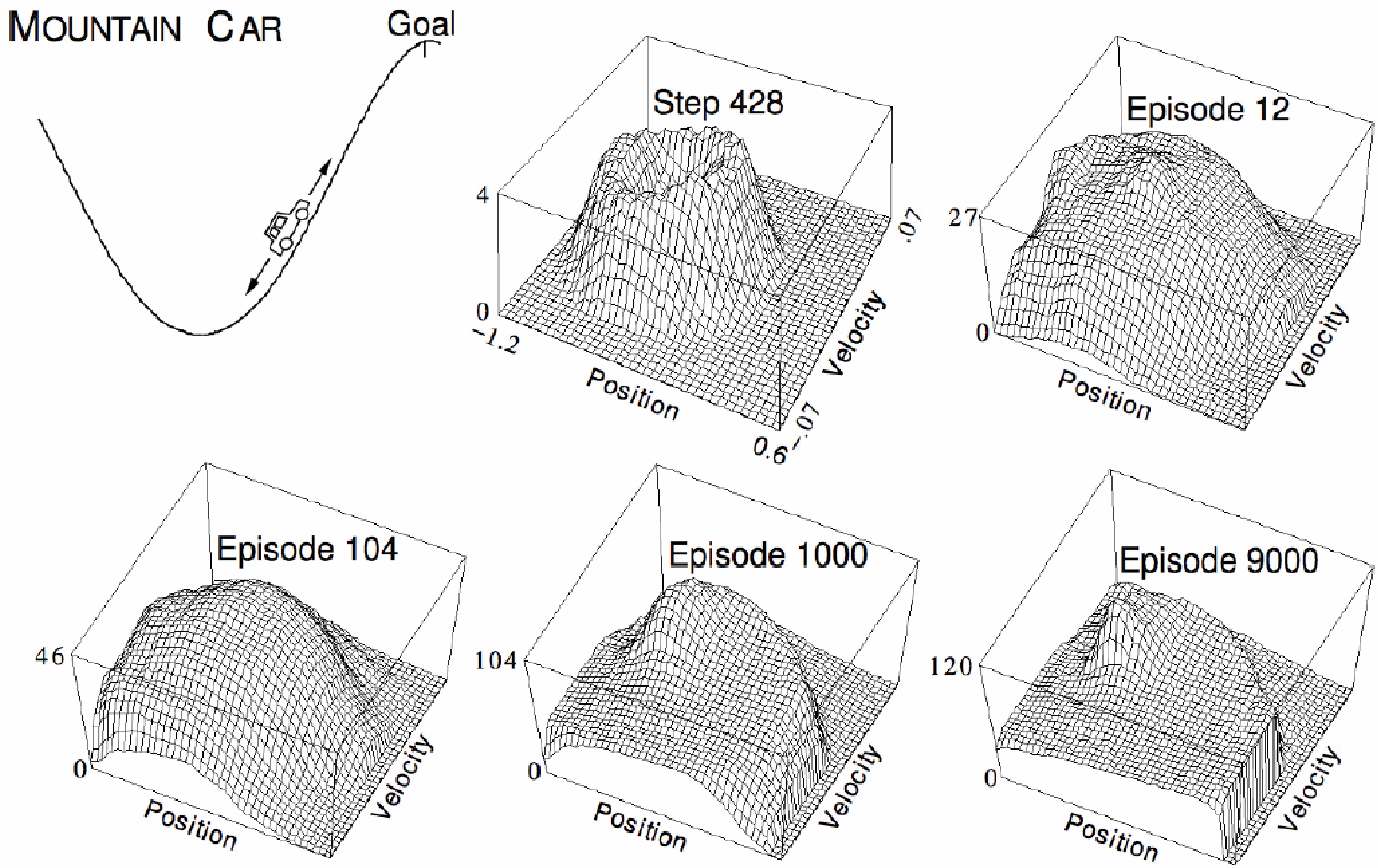
Episodic Semi-gradient Sarsa for Estimating $\hat{q} \approx q_*$

Input: a differentiable action-value function parameterization $\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}$
Algorithm parameters: step size $\alpha > 0$, small $\varepsilon > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)

Loop for each episode:
  $S, A \leftarrow$ initial state and action of episode (e.g., ε-greedy)
  Loop for each step of episode:
    Take action A, observe R, S'
    If S' is terminal:
      $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R - \hat{q}(S, A, \mathbf{w})\big]\nabla\hat{q}(S, A, \mathbf{w})$
      Go to next episode
    Choose A' as a function of $\hat{q}(S', \cdot, \mathbf{w})$ (e.g., ε-greedy)
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R + \gamma\hat{q}(S', A', \mathbf{w}) - \hat{q}(S, A, \mathbf{w})\big]\nabla\hat{q}(S, A, \mathbf{w})$
    $S \leftarrow S'$
    $A \leftarrow A'$

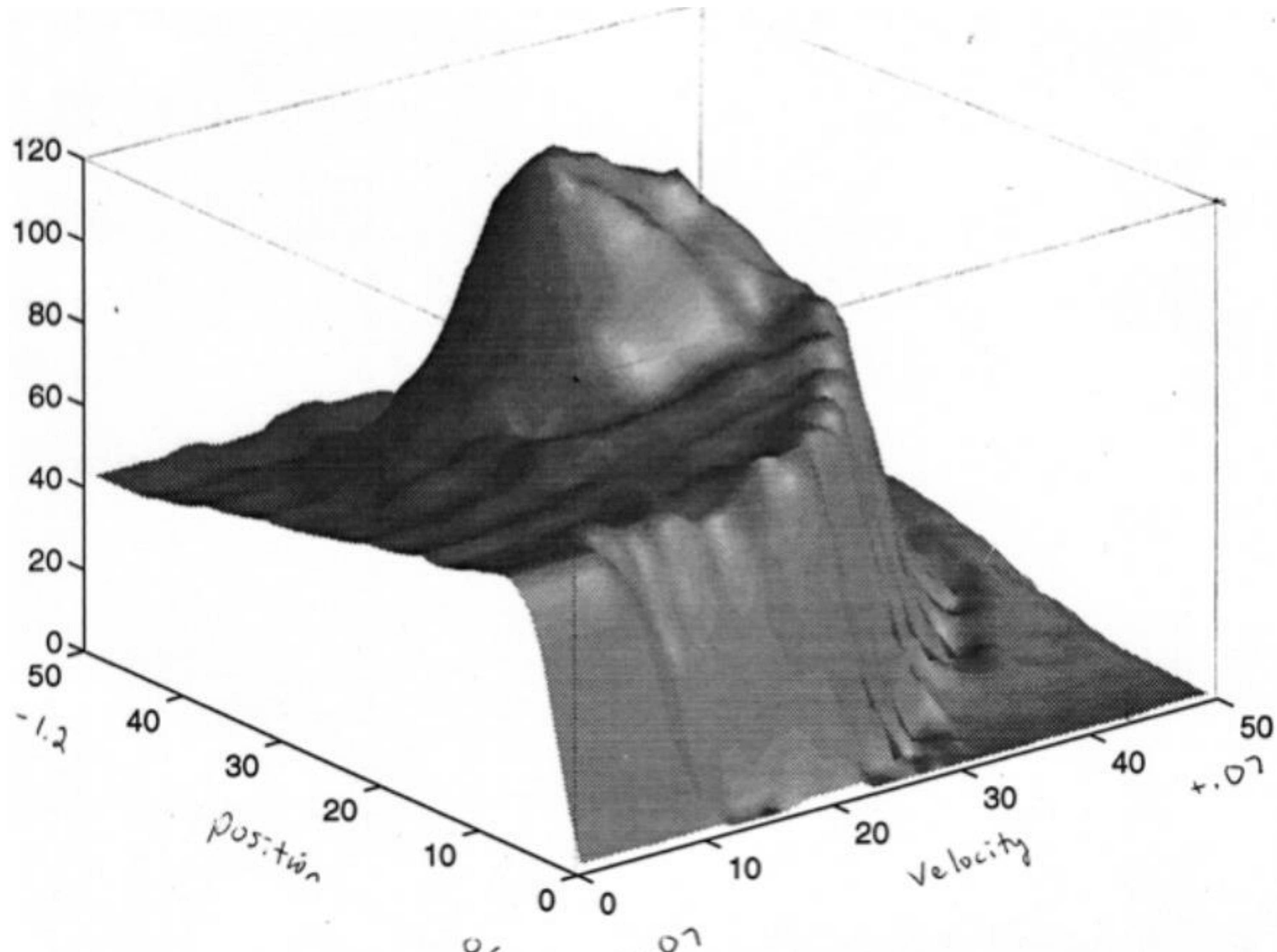
- Continuous하거나 action이 너무 많은 경우 이 방법이 적절하지 않음

EPSD – Mountain Car Task



- Position = (-1.2 , 0.6) / Velocity = (-0.07 , 0.07) / Reward = -1 (every time step)
- Continuous한 State와 Action -> Tile Coding 혹은 Radial Basis로 문제 해결

EPSD – Mountain Car Task



Episodic Semi-Gradient n-step Sarsa

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \hat{q}(S_{t+n}, A_{t+n}, \mathbf{w}_{t+n-1}),$$

Episodic semi-gradient n-step Sarsa for estimating $\hat{q} \approx q_*$ or q_π

Input: a differentiable action-value function parameterization $\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}$
Input: a policy π (if estimating q_π)
Algorithm parameters: step size $\alpha > 0$, small $\varepsilon > 0$, a positive integer n
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)
All store and access operations (S_t, A_t, and R_t) can take their index mod $n+1$

```
Loop for each episode:
    Initialize and store S_0 ≠ terminal
    Select and store an action A_0 ~ π(·|S_0) or ε-greedy wrt q̂(S_0, ·, w)
    T ← ∞
    Loop for t = 0, 1, 2, ... :
    |   If t < T, then:
    |       Take action A_t
    |       Observe and store the next reward as R_{t+1} and the next state as S_{t+1}
    |       If S_{t+1} is terminal, then:
    |           T ← t + 1
    |       else:
    |           Select and store A_{t+1} ~ π(·|S_{t+1}) or ε-greedy wrt q̂(S_{t+1}, ·, w)
    |   τ ← t − n + 1   (τ is the time whose estimate is being updated)
    |   If τ ≥ 0:
    |       G ← Σ_{i=τ+1}^{min(τ+n,T)} γ^{i−τ−1} R_i
    |       If τ + n < T, then G ← G + γ^n q̂(S_{τ+n}, A_{τ+n}, w)          (G_{τ:τ+n})
    |       w ← w + α [G − q̂(S_τ, A_τ, w)] ∇q̂(S_τ, A_τ, w)
    Until τ = T − 1
```

Average Reward

- Key idea : No discounting in Continuing Task cuz Reward Sequence

- Average Reward – The quality of a policy π

$$
\begin{aligned}
r(\pi) &\doteq \lim_{h\to\infty} \frac{1}{h} \sum_{t=1}^{h} \mathbb{E}[R_t \mid S_0, A_{0:t-1} \sim \pi] \\
&= \lim_{t\to\infty} \mathbb{E}[R_t \mid S_0, A_{0:t-1} \sim \pi], \\
&= \sum_s \mu_\pi(s) \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a) r,
\end{aligned}
$$

- $\mu_\pi(s)$ – Steady State Distribution
  - $\lim_{t\to\infty} \Pr\{S_t = s \mid A_{0:t-1} \sim \pi\}$
  - Reach some point in the process where the distributions will no longer change
  - State i, the probability step to an accessible state j will be the same

- Ergodicity
  - MDP starts or early decision made by agent are meaningless
  - Depend on Policy and MDP transition probabilities

- https://math.stackexchange.com/questions/133214/what-does-the-steady-state-represent-to-a-markov-chain

Average Reward

- New define of return in Average reward setting = Differential return

$$G_t \doteq R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + R_{t+3} - r(\pi) + \cdots.$$

- Simply remove all discounting factor and replace all rewards by (reward – true average reward)

$$= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\Big[r + \gamma v_\pi(s')\Big], \quad \text{for all } s \in \mathcal{S},$$

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

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

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

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

Average Reward – TD Error

- Episodic TD Error

$$\delta_t \doteq R_{t+1} + \gamma V(S_{t+1}) - V(S_t).$$

- Differential form of the TD Error

$$\delta_t \doteq R_{t+1} - \bar{R}_{t+1} + \hat{v}(S_{t+1}, \mathbf{w}_t) - \hat{v}(S_t, \mathbf{w}_t),$$

$$\delta_t \doteq R_{t+1} - \bar{R}_{t+1} + \hat{q}(S_{t+1}, A_{t+1}, \mathbf{w}_t) - \hat{q}(S_t, A_t, \mathbf{w}_t),$$

- $\bar{R} = estimate\ at\ time\ t\ of\ the\ average\ reward\ r(\pi)$

$$\left[R + \gamma \hat{q}(S', A', \mathbf{w}) - \hat{q}(S, A, \mathbf{w})\right]$$

$$\delta_t \doteq R_{t+1} - \bar{R}_{t+1} + \hat{q}(S_{t+1}, A_{t+1}, \mathbf{w}_t) - \hat{q}(S_t, A_t, \mathbf{w}_t),$$

-> $\bar{R} = (1-\gamma)\hat{q}(S, A, w)$ 으로 추정 (???)

Average Reward

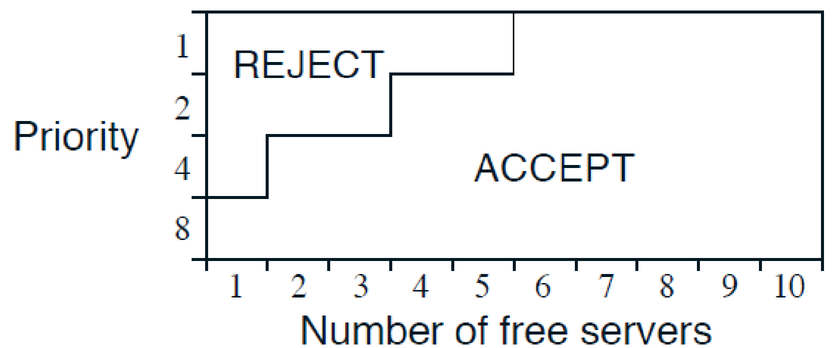
Differential semi-gradient Sarsa for estimating $\hat{q} \approx q_*$

Input: a differentiable action-value function parameterization $\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \rightarrow \mathbb{R}$
Algorithm parameters: step sizes $\alpha, \beta > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)
Initialize average reward estimate $\bar{R} \in \mathbb{R}$ arbitrarily (e.g., $\bar{R} = 0$)

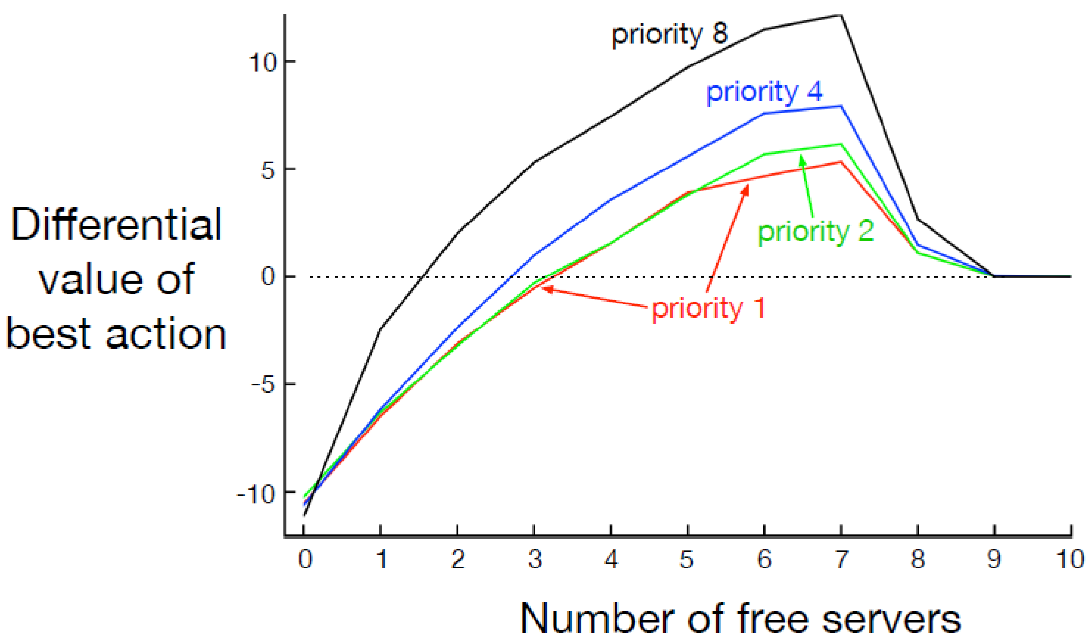
Initialize state S, and action A
Loop for each step:
    Take action A, observe R, S'
    Choose A' as a function of $\hat{q}(S', \cdot, \mathbf{w})$ (e.g., ε-greedy)
    $\delta \leftarrow R - \bar{R} + \hat{q}(S', A', \mathbf{w}) - \hat{q}(S, A, \mathbf{w})$
    $\bar{R} \leftarrow \bar{R} + \beta\delta$
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha\delta\nabla\hat{q}(S, A, \mathbf{w})$
    $S \leftarrow S'$
    $A \leftarrow A'$

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha\big[R + \gamma\hat{q}(S', A', \mathbf{w}) - \hat{q}(S, A, \mathbf{w})\big]\nabla\hat{q}(S, A, \mathbf{w})$$

AR – Access Control Queuing Task



POLICY



VALUE FUNCTION

Deprecating the Discounted Setting

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!

- Continuing Problem -> Reward Sequence

-> Averaging Rewards to evaluate Performance

- $\gamma(\pi)$ still exist!

Differential Semi-Gradient n-step Sarsa

- Differential Reward / Error with function approximation in n-step TD

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}),$$

$$G_{t:t+n} \doteq R_{t+1} - \bar{R}_{t+1} + R_{t+2} - \bar{R}_{t+2} + \cdots + R_{t+n} - \bar{R}_{t+n} + \hat{q}(S_{t+n}, A_{t+n}, \mathbf{w}_{t+n-1}),$$

$$\delta_t \doteq G_{t:t+n} - \hat{q}(S_t, A_t, \mathbf{w}),$$

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

Input: a differentiable function $\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}$, a policy π
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)
Initialize average-reward estimate $\bar{R} \in \mathbb{R}$ arbitrarily (e.g., $\bar{R} = 0$)
Algorithm parameters: step size $\alpha, \beta > 0$, a positive integer n
All store and access operations (S_t, A_t, and R_t) can take their index mod $n+1$

Initialize and store S_0 and A_0
Loop for each step, $t = 0, 1, 2, \ldots$:
  Take action A_t
  Observe and store the next reward as R_{t+1} and the next state as S_{t+1}
  Select and store an action $A_{t+1} \sim \pi(\cdot|S_{t+1})$, or ε-greedy wrt $\hat{q}(S_{t+1}, \cdot, \mathbf{w})$
  $\tau \leftarrow t - n + 1$ (τ is the time whose estimate is being updated)
  If $\tau \geq 0$:
    $\delta \leftarrow \sum_{i=\tau+1}^{\tau+n}(R_i - \bar{R}) + \hat{q}(S_{\tau+n}, A_{\tau+n}, \mathbf{w}) - \hat{q}(S_\tau, A_\tau, \mathbf{w})$
    $\bar{R} \leftarrow \bar{R} + \beta\delta$
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha\delta\nabla\hat{q}(S_\tau, A_\tau, \mathbf{w})$

Summary

- Episodic Semi-gradient Control
- Continuing Task – No discount / Average reward Setting
- Differential versions of value function / Bellman Equation / TD Error