Approximate Policy Iteration and Variants

Chen-Yu Wei

Review 1: $Q^\star/V^\star$

$V^\star(s)$ ≔ maximum expected total reward starting from state s

$Q^\star(s,a)$ ≔ maximum expected total reward starting from state s and taking action a **for one step**, and then following the optimal strategy

Value Iteration to approximate $Q^\star/V^\star$: **(for finding the optimal policy)**

For $i = 1, 2, \dots$

$$Q_i(s,a) \leftarrow R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V_{i-1}(s') \quad \text{for all } (s,a)$$

$$V_i(s) = \max_a Q_i(s,a) \quad \text{for all } s$$

Review 2: Q^π / V^π

Fix a policy π

$V^\pi(s) :=$ expected total reward starting from state s and following policy π

$Q^\pi(s,a) :=$ expected total reward starting from state s and taking action a for one step, and then following policy π

$\pi \longrightarrow Q^\pi$

(*)

Approximate Q^π / V^π : **(for evaluating a given policy)**

For $i = 1, 2, \ldots$

$$Q_i(s,a) \leftarrow R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V_{i-1}(s') \quad \text{for all } (s,a)$$

$$V_i(s) = \sum_{a} \pi(a|s)\, Q_i(s,a) \quad \text{for all } s$$

Policy Iteration

Another way (other than value iteration) to find the optimal policy in an MDP

Policy Iteration

VI $\boxed{Q_i(s,a)} \leftarrow R(s,a) + \gamma \sum_{s'} p(s'|s,a) \max_{a'} Q_{i-1}(s',a')$

Policy Iteration

For $i = 1, 2, \ldots$

$$\forall s, \qquad \pi_i(s) \leftarrow \underset{a}{\operatorname{argmax}} \boxed{Q^{\pi_{i-1}}(s,a)}$$

π_{i-1}

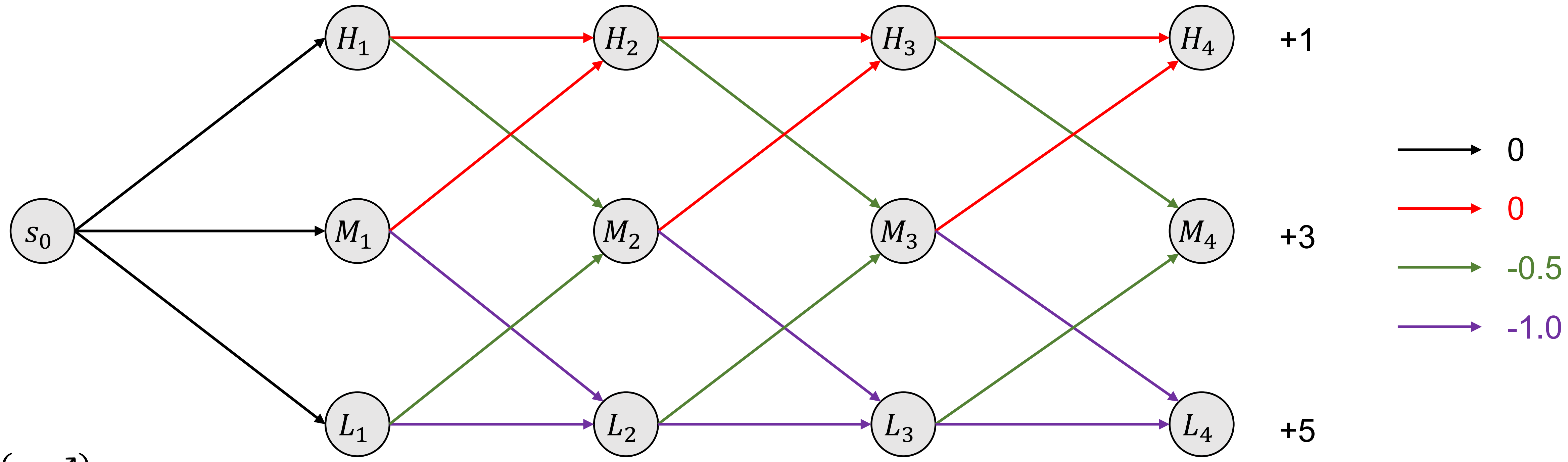
Call the (*) algo to calculate Q^{π_i}

In VI, there might not exist a policy π, such that $Q_i = Q^{\pi}$

Theorem (monotonic improvement). Policy Iteration ensures

$$\forall s, a, \qquad Q^{\pi_i}(s,a) \geq Q^{\pi_{i-1}}(s,a)$$

When converged (i.e., $\pi_i = \pi_{i-1}$), we have $\pi_i = \pi^{\star}$.



$Q^\pi(s_0, \nearrow) =$

$Q^\pi(s_0, \rightarrow) =$ $\quad Q^\pi(H_1, R) =$ $\quad Q^\pi(H_2, R) =$ $\quad Q^\pi(H_3, R) =$

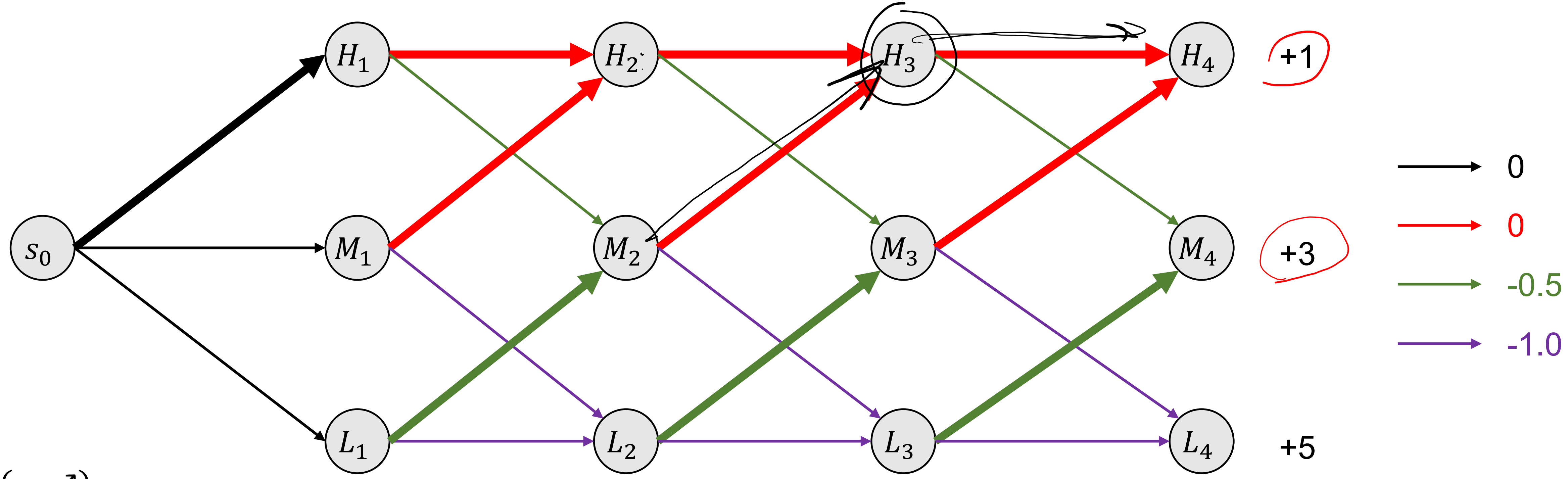
$Q^\pi(s_0, \searrow) =$ $\quad Q^\pi(H_1, G) =$ $\quad Q^\pi(H_2, G) =$ $\quad Q^\pi(H_3, G) =$

$Q^\pi(M_1, R) =$ $\quad Q^\pi(M_2, R) =$ $\quad Q^\pi(M_3, R) =$

$Q^\pi(M_1, P) =$ $\quad Q^\pi(M_2, P) =$ $\quad Q^\pi(M_3, P) =$

$Q^\pi(L_1, G) =$ $\quad Q^\pi(L_2, G) =$ $\quad Q^\pi(L_3, G) =$

$Q^\pi(L_1, P) =$ $\quad Q^\pi(L_2, P) =$ $\quad Q^\pi(L_3, P) =$



$Q^\pi(s_0, \nearrow) =$

$Q^\pi(s_0, \rightarrow) =$

$Q^\pi(s_0, \searrow) =$

$Q^\pi(H_1, R) =$

$Q^\pi(H_1, G) =$

$Q^\pi(M_1, R) =$

$Q^\pi(M_1, P) =$

$Q^\pi(L_1, G) =$

$Q^\pi(L_1, P) =$

$Q^\pi(H_2, R) = 1$

$Q^\pi(H_2, G) = -0.5 + 1$

$Q^\pi(M_2, R) = 0 + 1$

$Q^\pi(M_2, P) =$

$Q^\pi(L_2, G) =$

$Q^\pi(L_2, P) =$

$Q^\pi(H_3, R) = 0 + 1 = 1$

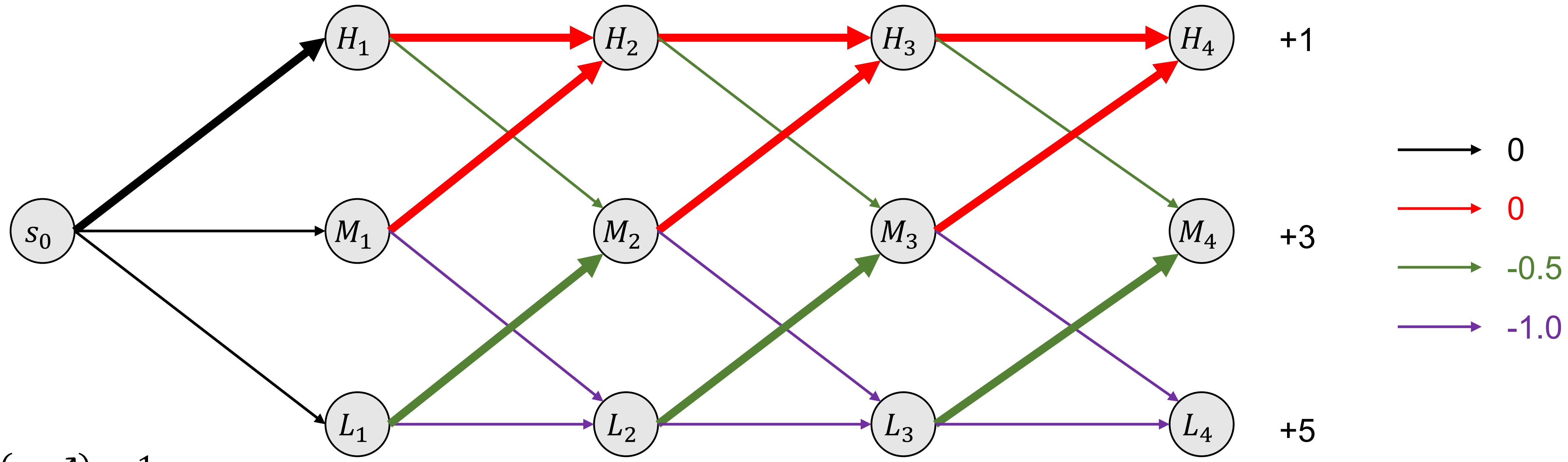
$Q^\pi(H_3, G) = -0.5 + 3 = 2.5$

$Q^\pi(M_3, R) = 1$

$Q^\pi(M_3, P) = 4$

$Q^\pi(L_3, G) =$

$Q^\pi(L_3, P) =$



$Q^{\pi}(s_0, \nearrow) = 1$

$Q^{\pi}(s_0, \rightarrow) = 1$ $\quad Q^{\pi}(H_1, R) = 1$ $\quad Q^{\pi}(H_2, R) = 1$ $\quad Q^{\pi}(H_3, R) = 1$

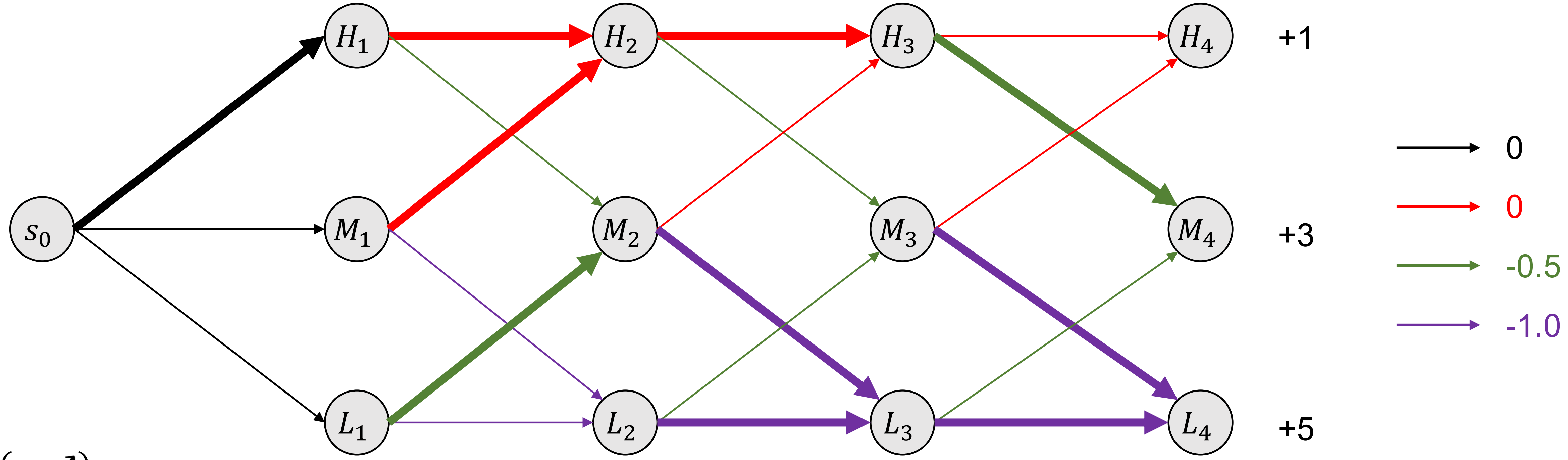
$Q^{\pi}(s_0, \searrow) = 0.5$ $\quad Q^{\pi}(H_1, G) = 0.5$ $\quad Q^{\pi}(H_2, G) = 0.5$ $\quad Q^{\pi}(H_3, G) = 2.5$

$Q^{\pi}(M_1, R) = 1$ $\quad Q^{\pi}(M_2, R) = 1$ $\quad Q^{\pi}(M_3, R) = 1$

$Q^{\pi}(M_1, P) = -0.5$ $\quad Q^{\pi}(M_2, P) = 1.5$ $\quad Q^{\pi}(M_3, P) = 4$

$Q^{\pi}(L_1, G) = 0.5$ $\quad Q^{\pi}(L_2, G) = 0.5$ $\quad Q^{\pi}(L_3, G) = 2.5$

$Q^{\pi}(L_1, P) = -0.5$ $\quad Q^{\pi}(L_2, P) = 1.5$ $\quad Q^{\pi}(L_3, P) = 4$



$Q^\pi(s_0, \nearrow) =$

$Q^\pi(s_0, \rightarrow) =$

$Q^\pi(s_0, \searrow) =$

$Q^\pi(H_1, R) =$ $\quad Q^\pi(H_2, R) =$ $\quad Q^\pi(H_3, R) =$

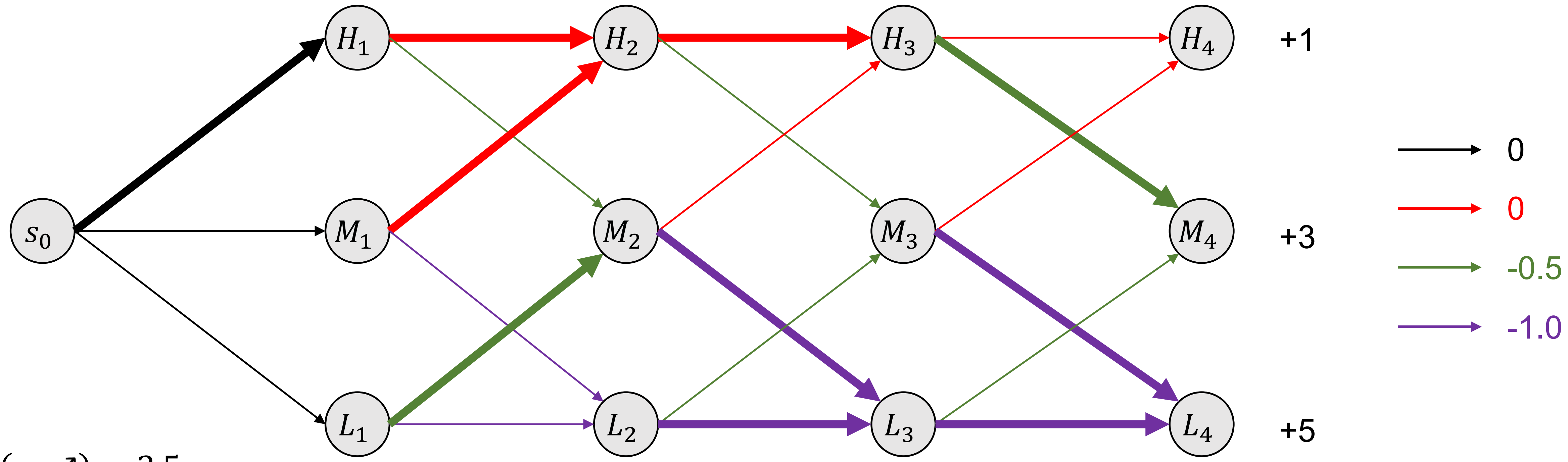
$Q^\pi(H_1, G) =$ $\quad Q^\pi(H_2, G) =$ $\quad Q^\pi(H_3, G) =$

$Q^\pi(M_1, R) =$ $\quad Q^\pi(M_2, R) =$ $\quad Q^\pi(M_3, R) =$

$Q^\pi(M_1, P) =$ $\quad Q^\pi(M_2, P) =$ $\quad Q^\pi(M_3, P) =$

$Q^\pi(L_1, G) =$ $\quad Q^\pi(L_2, G) =$ $\quad Q^\pi(L_3, G) =$

$Q^\pi(L_1, P) =$ $\quad Q^\pi(L_2, P) =$ $\quad Q^\pi(L_3, P) =$

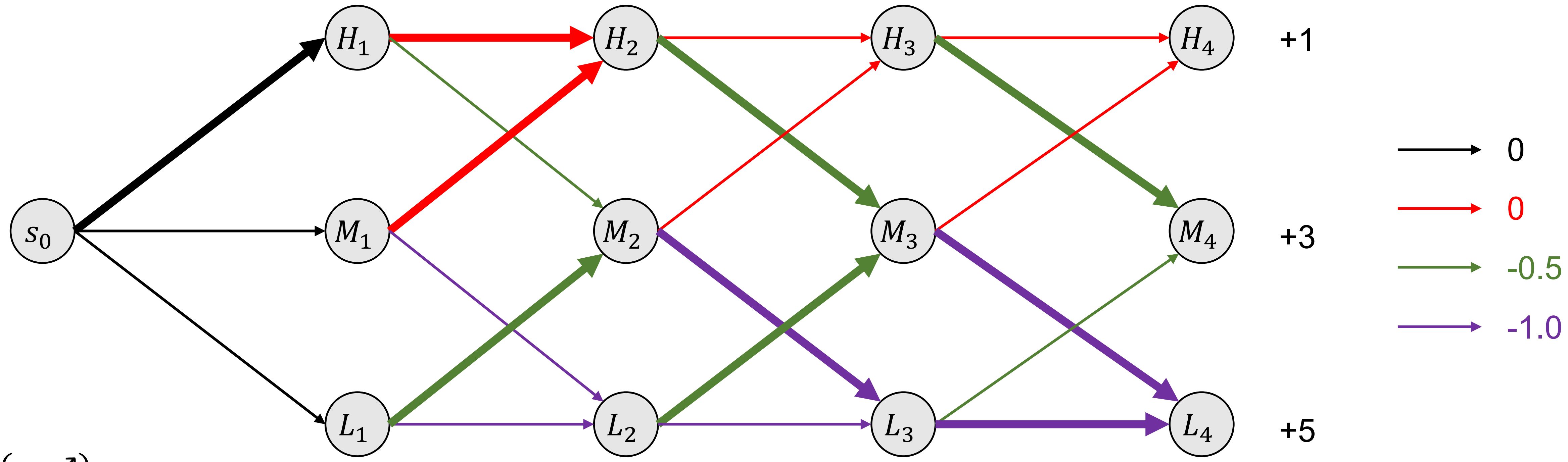


$Q^\pi(s_0, \nearrow) = 2.5$

$Q^\pi(s_0, \rightarrow) = 2.5$

$Q^\pi(s_0, \searrow) = 2.5$

$Q^\pi(H_1, R) = 2.5$	$Q^\pi(H_2, R) = 2.5$	$Q^\pi(H_3, R) = 1$
$Q^\pi(H_1, G) = 2.5$	$Q^\pi(H_2, G) = 3.5$	$Q^\pi(H_3, G) = 2.5$
$Q^\pi(M_1, R) = 2.5$	$Q^\pi(M_2, R) = 2.5$	$Q^\pi(M_3, R) = 1$
$Q^\pi(M_1, P) = 2$	$Q^\pi(M_2, P) = 3$	$Q^\pi(M_3, P) = 4$
$Q^\pi(L_1, G) = 2.5$	$Q^\pi(L_2, G) = 3.5$	$Q^\pi(L_3, G) = 2.5$
$Q^\pi(L_1, P) = 2$	$Q^\pi(L_2, P) = 3$	$Q^\pi(L_3, P) = 4$

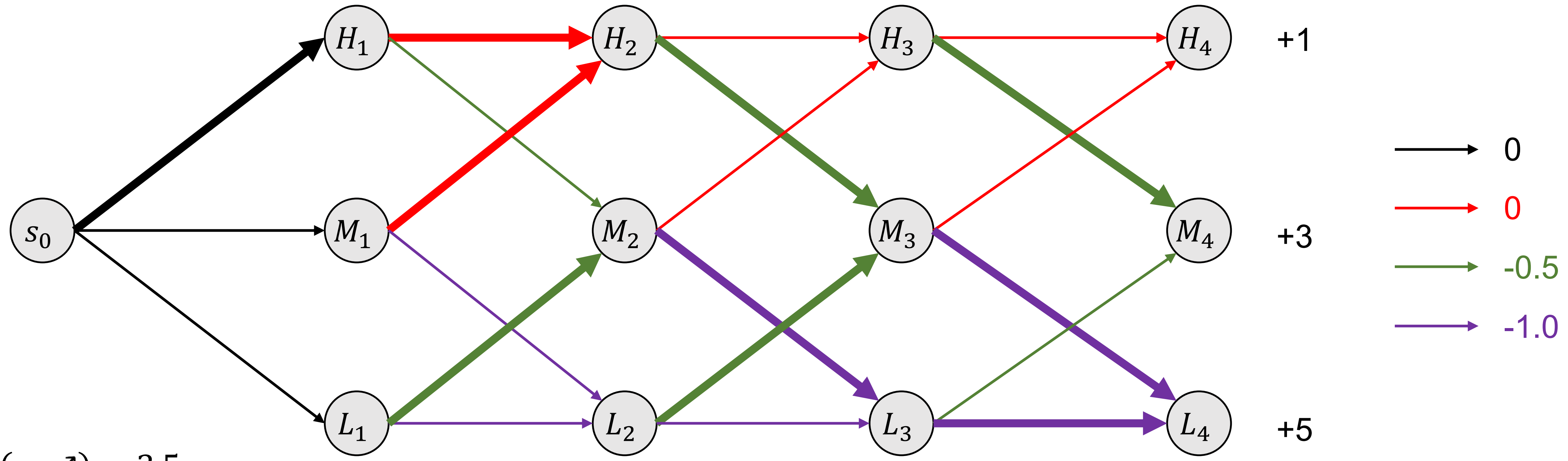


$Q^\pi(s_0, \nearrow) =$

$Q^\pi(s_0, \rightarrow) =$

$Q^\pi(s_0, \searrow) =$

$Q^\pi(H_1, R) =$	$Q^\pi(H_2, R) =$	$Q^\pi(H_3, R) =$
$Q^\pi(H_1, G) =$	$Q^\pi(H_2, G) =$	$Q^\pi(H_3, G) =$
$Q^\pi(M_1, R) =$	$Q^\pi(M_2, R) =$	$Q^\pi(M_3, R) =$
$Q^\pi(M_1, P) =$	$Q^\pi(M_2, P) =$	$Q^\pi(M_3, P) =$
$Q^\pi(L_1, G) =$	$Q^\pi(L_2, G) =$	$Q^\pi(L_3, G) =$
$Q^\pi(L_1, P) =$	$Q^\pi(L_2, P) =$	$Q^\pi(L_3, P) =$



$Q^\pi(s_0, \nearrow) = 3.5$

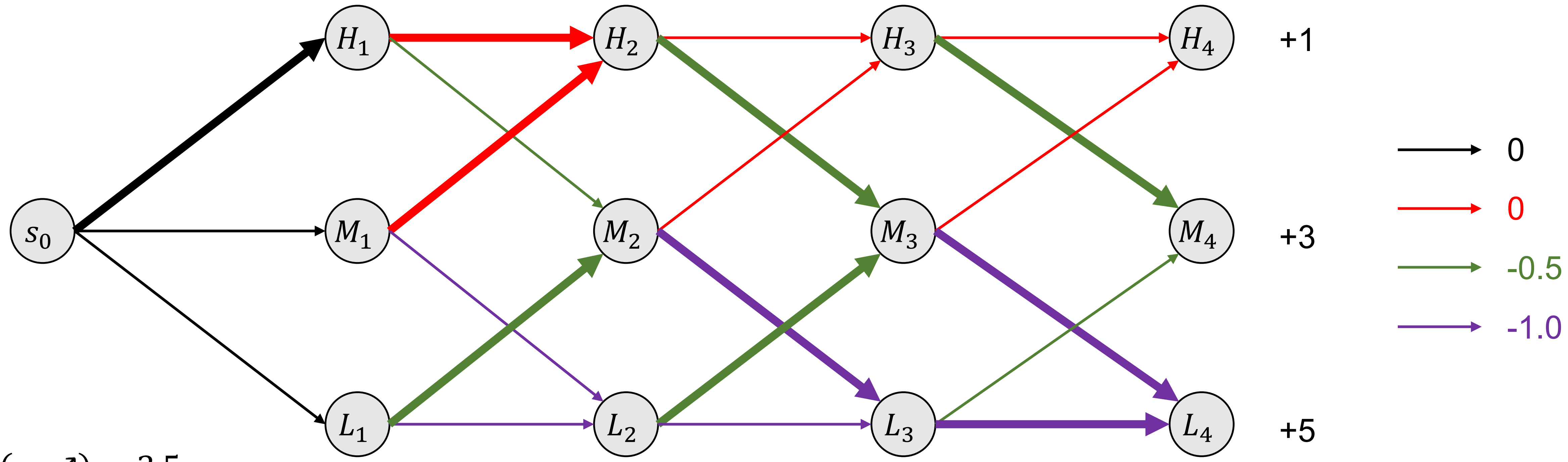
$Q^\pi(s_0, \rightarrow) = 3.5$

$Q^\pi(s_0, \searrow) = 2.5$

$Q^\pi(H_1, R) = 3.5$	$Q^\pi(H_2, R) = 2.5$	$Q^\pi(H_3, R) = 1$
$Q^\pi(H_1, G) = 2.5$	$Q^\pi(H_2, G) = 3.5$	$Q^\pi(H_3, G) = 2.5$
$Q^\pi(M_1, R) = 3.5$	$Q^\pi(M_2, R) = 2.5$	$Q^\pi(M_3, R) = 1$
$Q^\pi(M_1, P) = 2.5$	$Q^\pi(M_2, P) = 3$	$Q^\pi(M_3, P) = 4$
$Q^\pi(L_1, G) = 2.5$	$Q^\pi(L_2, G) = 3.5$	$Q^\pi(L_3, G) = 2.5$
$Q^\pi(L_1, P) = 2.5$	$Q^\pi(L_2, P) = 3$	$Q^\pi(L_3, P) = 4$

$\underline{\pi_i(s)} = \arg\max Q^{\pi_{i-1}}(s,a)$

$\underline{\pi_i = \pi_{i-1}}$



$Q^\pi(s_0, \nearrow) = 3.5$

$Q^\pi(s_0, \rightarrow) = 3.5$

$Q^\pi(s_0, \searrow) = 2.5$

$Q^\pi(H_1, R) = 3.5$	$Q^\pi(H_2, R) = 2.5$	$Q^\pi(H_3, R) = 1$
$Q^\pi(H_1, G) = 2.5$	$Q^\pi(H_2, G) = 3.5$	$Q^\pi(H_3, G) = 2.5$
$Q^\pi(M_1, R) = 3.5$	$Q^\pi(M_2, R) = 2.5$	$Q^\pi(M_3, R) = 1$
$Q^\pi(M_1, P) = 2.5$	$Q^\pi(M_2, P) = 3$	$Q^\pi(M_3, P) = 4$
$Q^\pi(L_1, G) = 2.5$	$Q^\pi(L_2, G) = 3.5$	$Q^\pi(L_3, G) = 2.5$
$Q^\pi(L_1, P) = 2.5$	$Q^\pi(L_2, P) = 3$	$Q^\pi(L_3, P) = 4$

$\pi_{i+1} = \pi_i \Rightarrow$ converged

Policy Evaluation with Samples

Policy Iteration

For $k = 1,\ 2, \ldots$

Calculate $Q^{\pi_k}(s, a) \quad \forall s, a$ Policy Evaluation

$\pi_{k+1}(s) = \underset{a}{\operatorname{argmax}}\, Q^{\pi_k}(s, a) \quad \forall s$ Policy Improvement

Policy Iteration with Samples

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

(s_i, a_i, r_i, s_i')

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

Data collection

Evaluate $Z_k(s,a) \approx Q^{\pi_{\theta_k}}(s,a)$ for $s = s_1, \ldots, s_N$ and all a

or $Z_k(s,a) \approx Q^{\pi_{\theta_k}}(s,a) - b_k(s)$for $s = s_1, \ldots, s_N$ and all a

Policy Evaluation

Update θ_{k+1} from θ_k using the estimators $\{Z_k(s_i, a)\}_{i=1}^{N}$

Using any technique we introduced for policy-based contextual bandits

Policy Improvement

Monte Carlo Estimators

Policy Evaluation with Monte Carlo Estimator

Recall

$r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \gamma^3 r_{i+3} + \quad + \gamma^\tau r_{i+\tau}$

π_{θ_k}

$$Q^\pi(s_i, a_i) = \mathbb{E}[R(s_i, a_i) + \gamma R(s_{i+1}, a_{i+1}) + \gamma^2 R(s_{i+2}, a_{i+2}) + \cdots + \gamma^\tau R(s_{i+\tau}, a_{i+\tau}) \mid \text{following } \pi]$$

(Expected sum of reward starting from (s_i, a_i) and following s_{i+1}) End of episode

A natural estimator $Z_k(s_i, a)$ with $\mathbb{E}[Z_k(s_i, a)] = Q^{\pi_{\theta_k}}(s_i, a)$:

$Q^{\pi_{\theta_k}}(s_i, a_i)$

$$Z_k(s_i, a) = \frac{r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^\tau r_{i+\tau} - b(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\}$$

Contextual bandit special case: $Z_k(x_i, a) = \hat{r}(x_i, a) = \dfrac{r_i - b(x_i)}{\pi_{\theta_k}(a|x_i)} \mathbb{I}\{a_i = a\}$ (see e.g. Page 34 here)

Policy Iteration with Samples (w/ Monte Carlo Estimator)

For $k = 1, 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot \mid s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot \mid s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

Define for $i = 1, 2, \ldots, N$ and for all a:

$$Z_k(s_i, a) = \frac{r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^\tau r_{i+\tau} - b_k(s_i)}{\pi_{\theta_k}(a \mid s_i)} \mathbb{I}\{a_i = a\}$$

$$\theta_{k+1} = \operatorname*{argmax}_\theta \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(\sum_a \pi_\theta(a \mid s_i)\, Z_k(s_i, a) - \frac{1}{\eta} \mathrm{KL}\big(\pi_\theta(\cdot \mid s_i), \pi_{\theta_k}(\cdot \mid s_i)\big) \right) \right\}$$

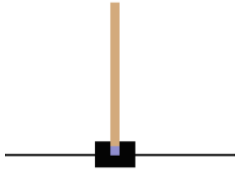
$$= \operatorname*{argmax}_\theta \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(\frac{\pi_\theta(a_i \mid s_i)}{\pi_{\theta_k}(a_i \mid s_i)} \big(r_i + \gamma r_{i+1} + \cdots + \gamma^\tau r_{i+\tau} - b_k(s_i)\big) - \frac{1}{\eta} \mathrm{KL}\big(\pi_\theta(\cdot \mid s_i), \pi_{\theta_k}(\cdot \mid s_i)\big) \right) \right\}$$

Data collection

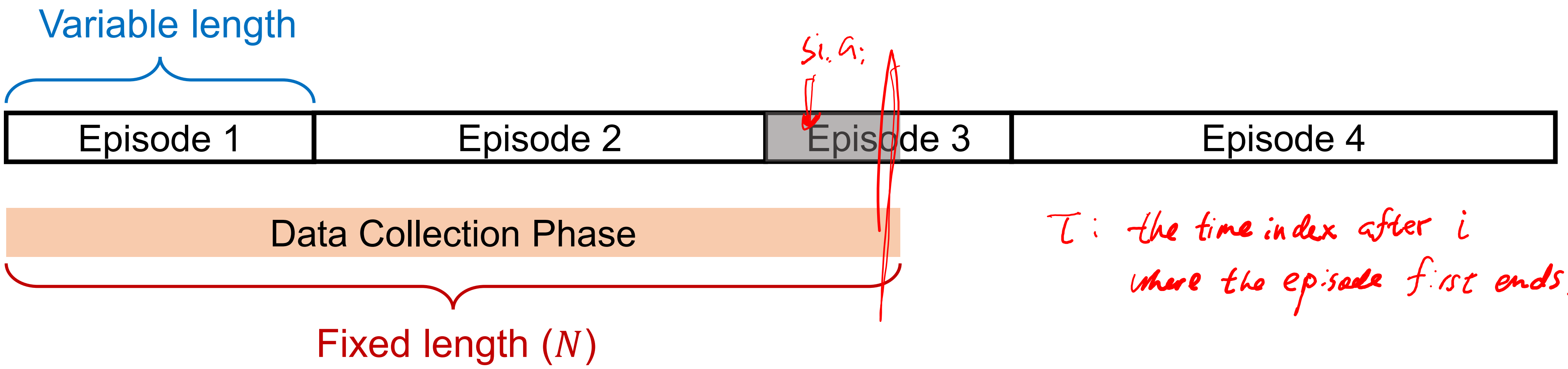
Policy Evaluation

Policy Improvement

A Caveat



The **episode end** may go beyond the **end of the data collection phase**



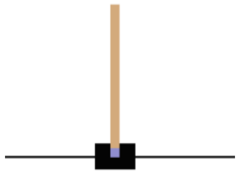
τ : the time index after i where the episode first ends.

Therefore, for part of the data (the gray segment above), we're unable to create correct Monte Carlo estimator

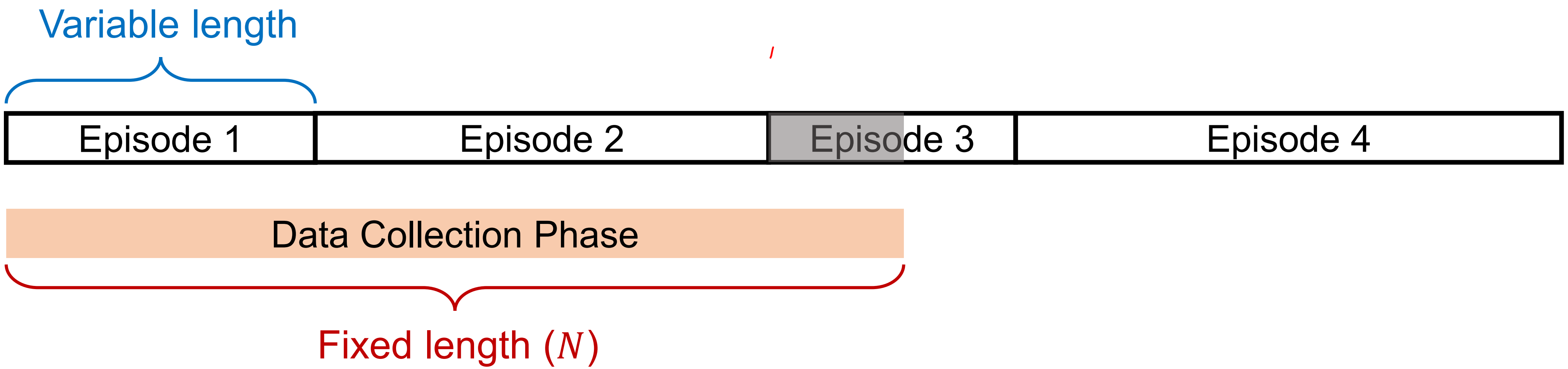
$Q^{\pi_{\theta_k}}(s_i, a_i)$

$$Z_k(s_i, a) = \frac{r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^\tau r_{i+\tau} - b_k(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\}$$

A Caveat



The **episode end** may go beyond the **end of the data collection phase**



Solutions:

- Use variable length data collection phase that always include complete episodes
- Drop the incomplete-episode samples (the gray part)
- Use alternative estimators we will discuss next

For $k = 1, 2, \cdots,$

For $i = 1, 2, \cdots, N$:

Choose $a_i \sim \pi_{\theta_k}(\cdot | s_i)$

Receive $r_i \sim R(s_i, a_i)$, $s_i' \sim P(\cdot | s_i, a_i)$

$s_{i+1} = s_i'$ if the episode continues , and $s_{i+1} \sim \rho$ if the episode terminates

// This procedure gives us $(s_1, a_1, r_1, s_2, a_2, r_2, \cdots\cdots s_N, a_N, r_N)$

$Q^{\pi_{\theta_k}}(s_i, a_i)$

Define

$$Z_k(s_i, a) = \frac{\sum_{j=i}^{\square} \gamma^{j-i} r_j - \text{baseline}}{\pi_{\theta_k}(a | s_i)} \mathbb{1}\{a_i = a\}$$

create estimator $Z_k(s,a)$ for $Q^{\pi_{\theta_k}}(s,a)$

$$\hat{r}_k(x_i, a) = \frac{r_i - \text{baseline}}{\pi_{\theta_k}(a | x_i)} \mathbb{1}\{a_i = a\} \quad \text{(contextual bandits)}$$

$$\theta_{k+1} = \arg\max_{\theta} \left\{ \sum_{i=1}^{N} \pi_{\theta}(a | s_i) \underline{Z_k(s_i, a)} - \frac{1}{\eta} KL\left(\pi_{\theta}(\cdot | s_i), \pi_{\theta_k}(\cdot | s_i)\right) \right\}$$

Temporal Difference Estimators

Recall: Policy Iteration with Samples

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

Data collection

$Z_k(s, a) \quad \frac{r_i - b(x_i)}{\pi(a|x_i)}$

Evaluate $Z_k(s, a) \approx Q^{\pi_{\theta_k}}(s, a)$ for $s = s_1, \ldots, s_N$ and all a

or $Z_k(s, a) \approx Q^{\pi_{\theta_k}}(s, a) - b_k(s)$for $s = s_1, \ldots, s_N$ and all a

Policy Evaluation

Update θ_{k+1} from θ_k using the estimators $\{Z_k(s_i, a)\}_{i=1}^{N}$

Using any technique we introduced for policy-based contextual bandits

Policy Improvement

More General Ways to Create $Q^{\pi_{\theta_k}}(s, a)$ Estimators

Our goal is a create an estimator $Z_k(s_i, a)$ with $\mathbb{E}[Z_k(s_i, a)] = Q^{\pi_{\theta_k}}(s_i, a)$

Previously we set

$$Z_k(s_i, a) = \frac{r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^\tau r_{i+\tau} - b_k(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\}$$

just samples from the environment

In general, the following is a valid estimator:

$$Z_k(s_i, a) = \frac{\text{Any estimation of } Q^{\pi_{\theta_k}}(s_i, a_i) - b_k(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\}$$

Using Another Neural Network to Approximate V^{π}

$$Q^{\pi}(s_i, a_i) = \mathbb{E}[R(s_i, a_i) + \gamma R(s_{i+1}, a_{i+1}) + \cdots + \gamma^{\tau} R(s_{i+\tau}, a_{i+\tau}) | \text{ following } \pi]$$

$$= \mathbb{E}[R(s_i, a_i) + \gamma V^{\pi}(s_{i+1}) | \text{ following } \pi]$$

$$= \mathbb{E}[R(s_i, a_i) + \gamma R(s_{i+1}, a_{i+1}) + \gamma^2 V^{\pi}(s_{i+2}) | \text{ following } \pi]$$

$$= \mathbb{E}[R(s_i, a_i) + \gamma R(s_{i+1}, a_{i+1}) + \gamma^2 R(s_{i+2}, a_{i+2}) + \gamma^3 V^{\pi}(s_{i+1}) | \text{ following } \pi]$$

$$\vdots$$

$$= E\left[R(s_i, a_i) + \gamma R(s_{i+1}, a_{i+1}) + \cdots + \gamma^{\tau} R(s_{i+\tau}, a_{i+\tau})\right.$$

For example, the following is an estimator for $Q^{\pi_{\theta_k}}(s_i, a)$:

$$\approx \gamma(r_{i+1} + \gamma r_{i+2} + \gamma^2 r_{i+3} + \cdots)$$

$$Z_k(s_i, a) = \frac{r_i + \gamma \hat{V}(s_{i+1}) - b_k(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\} \qquad \text{where } \hat{V} \approx V^{\pi_{\theta_k}}$$

Using Another Neural Network to Approximate V^{π}

How to estimate V^{π}?

With true reward and transition:

Repeat:

$$V_{k+1}(s) \leftarrow \sum_{a} \pi(a|s) \left(R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V_k(s') \right) \qquad \text{for all } s$$

With samples $(s_1, a_1, r_1, s_2, a_2, r_2, \dots)$ collected from π and neural network $V_{\phi}(s)$:

Repeat:

$$\phi_{k+1} \leftarrow \underset{\phi}{\operatorname{argmin}} \; \frac{1}{N} \sum_{i=1}^{N} \left(V_{\phi}(s_i) - r_i - \gamma V_{\phi_k}(s_{i+1}) \right)^2$$

Policy Iteration with Samples (w/ TD Estimator)

For $k = 1,\ 2, \dots$

For $i = 1, 2, \dots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

Data collection

Define for $i = 1, 2, \dots, N$ and for all a:

$$Z_k(s_i, a) = \frac{\boldsymbol{r_i + \gamma V_{\phi_k}(s_{i+1})} - b_k(s_i)}{\pi_{\theta_k}(a|s_i)}\ \mathbb{I}\{a_i = a\}$$

Policy Evaluation

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} \left\{\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{a}\pi_\theta(a|s_i)\, Z_k(s_i, a) - \frac{1}{\eta}\mathrm{KL}\big(\pi_\theta(\cdot\,|s_i), \pi_{\theta_k}(\cdot\,|s_i)\big)\right)\right\}$$

Perform several times: $\phi \leftarrow \phi - \alpha\nabla_\phi \frac{1}{N}\sum_{i=1}^{N}\left(V_\phi(s_i) - r_i - \gamma V_{\phi_k}(s_{i+1})\right)^2$

Policy Improvement

+ V_ϕ update

Policy Iteration with Samples (w/ TD Estimator)

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

Data collection

Define for $i = 1, 2, \ldots, N$ and for all a:

$$Z_k(s_i, a) = \frac{r_i + \gamma r_{i+1} + \gamma^2 V_{\phi_k}(s_{i+2}) - b(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\}$$

Policy Evaluation

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{a} \pi_\theta(a|s_i)\, Z_k(s_i, a) - \frac{1}{\eta} \mathrm{KL}\big(\pi_\theta(\cdot\,|s_i), \pi_{\theta_k}(\cdot\,|s_i)\big) \right) \right\}$$

Perform several times: $\phi \leftarrow \phi - \alpha \nabla_\phi \frac{1}{N} \sum_{i=1}^{N} \left(V_\phi(s_i) - r_i - \gamma V_{\phi_k}(s_{i+1}) \right)^2$

Policy Improvement

+ V_ϕ update

$$Z_k(s_i, a) = \frac{r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^\tau r_{i+\tau} - b(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}(a_i = a)$$

MC

high variance
unbiased

$$= \frac{r_i + \gamma V_{\phi_k}(s_{i+1}) - b(s_i)}{\pi_\theta(a|s_i)} \mathbb{I}(a_i = a)$$

TD

lower variance
more biased

$$= \frac{r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^m r_{i+m} + \gamma^{m+1} V_{\phi_k}(s_{i+m+1})}{\pi_\theta(a|s_i)}$$

(general)

If $\tau < m$

$$\frac{r_i + \gamma r_{i+1} + \cdots + \gamma^\tau r_{i+\tau}}{\pi_\theta(a|s_i)}$$

A Family of Z_k Estimators

$$Z_k(s_i, a) = \frac{\boldsymbol{r_i + \gamma r_{i+1} + \cdots + \gamma^{\tau-i} r_\tau} - b_k(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\}$$

Monte Carlo (MC) Estimator

$$Z_k(s_i, a) = \frac{\boldsymbol{r_i + \gamma V_\phi(s'_{i+1})} - b_k(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\}$$

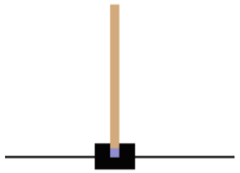
Temporal Difference (TD) Estimator

$$Z_k(s_i, a) = \frac{\boldsymbol{r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^m r_{i+m} + V_\phi(s'_{i+m+1})} - b_k(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\}$$

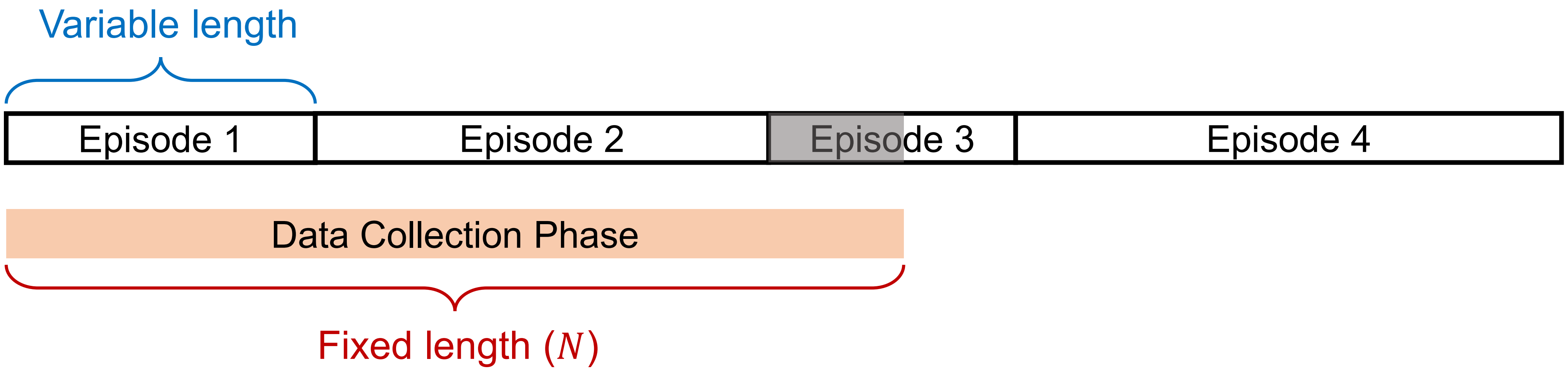
Multi-step TD Estimator

(If the episode ends before m steps, this falls back to the MC estimator)

Recall: A Caveat of MC Estimator



The **episode end** may go beyond the **end of the data collection phase**



Issue: we may not construct $r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^{\tau_i} r_{i+\tau_i}$ for all i

Solution: use $r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^{N-i} r_N + \gamma^{N-i+1} V_\phi(s'_N)$ if i is in an incomplete episode

Recall: A Caveat of MC Estimator

E.g., $N = 16$. Episodes end at steps $5, 11, 19$

For $i = 1, \dots, 5$:

$$Z_k(s_i, a) = \frac{\boldsymbol{r_i + \gamma r_{i+1} + \cdots + \gamma^{5-i} r_5} - b_k(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\}$$

For $i = 6, \dots, 11$:

$$Z_k(s_i, a) = \frac{\boldsymbol{r_i + \gamma r_{i+1} + \cdots + \gamma^{11-i} r_{11}} - b_k(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\}$$

For $i = 12, \dots, 16$:

$$Z_k(s_i, a) = \frac{\boldsymbol{r_i + \gamma r_{i+1} + \cdots + \gamma^{16-i} r_{16} + \gamma^{17-i} V_\phi(s'_{16})} - b_k(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\}$$

How to Calculate Z_k Efficiently for MC Estimator

Assume we have collected $(s_1, a_1, r_1, s_1'), (s_2, a_2, r_2, s_2'), \ldots, (s_N, a_N, r_N, s_N')$, where $s_{i+1} = s_i'$ if s_i' is not a terminal state, and s_{i+1} is redrawn from initial state otherwise.

How to calculate

$$r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^{\tau_i} r_{i+\tau_i}$$
$$r_i + \gamma r_{i+1} + \gamma^2 r_{i+2} + \cdots + \gamma^{N-i} r_N + \gamma^{N-i+1} V_\phi(s_N')$$

efficiently for all $i = 1, \ldots, N$?

Naïve implementation requires $\mathrm{O}(N \times \mathrm{episode\_length})$ time

How to Calculate Z_k Efficiently for MC Estimator

For an incomplete episode

For a complete episode that ends at τ

How to Calculate Z_k Efficiently for MC Estimator

$G_{N+1} = V_\phi(s'_N)$

For $i = N, N-1, \dots, 1$:

If s'_i is a terminal state:

$G_i = r_i$

Else:

$G_i = r_i + \gamma G_{i+1}$

For $i = 1, \dots, N$:

Define $Z_k(s_i, a) = \frac{G_i - b_k(s_i)}{\pi_{\theta_k}(a|s_i)} \mathbb{I}\{a_i = a\}$

Remark: For TD estimator, G_i is simply calculated as $r_i + \gamma V_\phi(s'_i)$

For general (m-step) estimator, we can also design efficient algorithm (omitted)

Policy Iteration with Samples

For $k = 1,\ 2, \ldots$

For $i = 1, 2, \ldots, N$:

Choose action $a_i \sim \pi_{\theta_k}(\cdot\,|s_i)$

Receive reward $r_i \sim R(s_i, a_i)$ and $s_i' \sim P(\cdot\,|s_i, a_i)$

$s_{i+1} = s_i'$ if episode continues, $s_{i+1} \sim \rho$ if episode ends

Data collection

Define for $i = 1, 2, \ldots, N$ and for all a:

$$Z_k(s_i, a) = \frac{\boldsymbol{G_i} - b_k(s_i)}{\pi_{\theta_k}(a|s_i)}\,\mathbb{I}\{a_i = a\}$$

with $b_k(s_i) = V_\phi(s_i)$

Policy Evaluation

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} \left\{ \frac{1}{N}\sum_{i=1}^{N} \left(\sum_{a} \pi_\theta(a|s_i)\, Z_k(s_i, a) - \frac{1}{\eta}\mathrm{KL}\big(\pi_\theta(\cdot\,|s_i), \pi_{\theta_k}(\cdot\,|s_i)\big) \right) \right\}$$

Perform several times: $\phi \leftarrow \phi - \alpha \nabla_\phi \frac{1}{N}\sum_{i=1}^{N}\left(V_\phi(s_i) - r_i - \gamma V_{\phi_k}(s_{i+1})\right)^2$

Policy Improvement

+ V_ϕ update

Generalized Policy Iteration

$N = \infty \Rightarrow$ Policy Iteration

$N = 1 \Rightarrow$ Value Iteration for policy optimization

For $i = 1, 2, \ldots$

$$\pi_i(s) = \arg\max_a Q_i(s, a)$$

Policy update

$Q \leftarrow Q_i$

Repeat for N times:

$$Q(s, a) \leftarrow R(s, a) + \gamma \sum_{s', a'} P(s' | s, a)\, \pi_i(a' | s') Q(s', a')$$

Value update

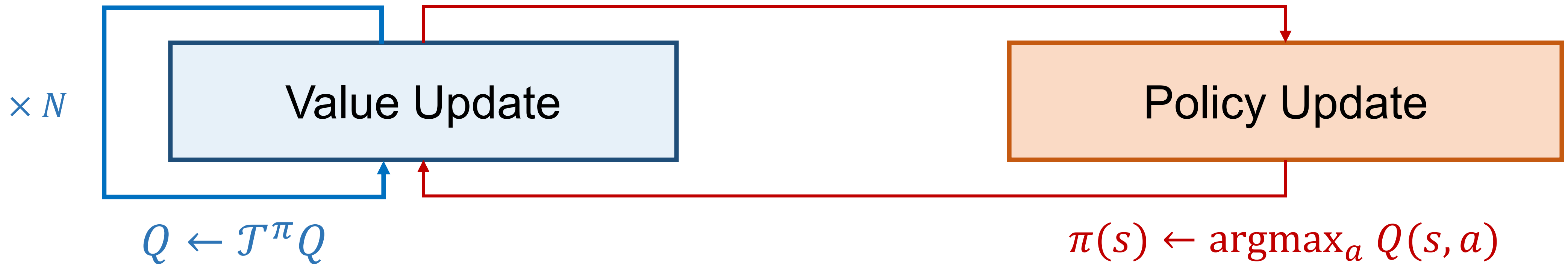
$Q_{i+1} \leftarrow Q$

Notice: in value iteration, there may not exist a policy π such that $Q_i = Q^{\pi}$

In contrast, in policy iteration we have $Q_i = Q^{\pi_{i-1}}$

VI can be viewed as PI **with incomplete policy evaluation**

Generalized Policy Iteration



$Q \leftarrow \mathcal{T}^\pi Q$ means $\quad Q(s,a) \leftarrow R(s,a) + \gamma \sum_{s',a'} P(s'|s,a)\,\pi(a'|s') Q(s',a') \quad$ for all s,a

- Provide a unified view for algorithms approximating optimal policy with known P,R
- Provide a unified view for “value-based” and “policy-based” algorithms
- Compared with bandits:
  - The R is replaced by Q (could be Q^π or $Q^\star$) to capture the long-term goal of the learner

Summary for Policy-Based RL Algorithms

- We introduce Policy Iteration, an algorithm that iterates between **policy evaluation** and **policy improvement**, assuming access to true transition and reward
- We introduce the PPO algorithm for MDPs
  - Almos the same as its special case in contextual bandits, except that we replace $\hat{r}_k(x, a)$ (an estimator for $r(x, a)$) by $Z_k(s, a)$ (an estimator for $Q^{\pi_{\theta_k}}(s, a)$)
  - There are a family of estimators you may choose from, depending on how much the algorithm relies on V_ϕ or real reward collected from environments
  - V_ϕ also serves the baseline
- We introduce Generalized Policy Iteration as a way to unify Policy Iteration and Value Iteration