Lecture 8: TD Learning and Eligibility Trace

Course: Reinforcement Learning Theory
Instructor: Lei Ying
Department of EECS
University of Michigan, Ann Arbor

Value/Q-function evaluation

Value Iteration for a given policy (Actor-Critic)

$$J_{k+1}(x_k) = J_k(x_k) + \beta_k(r(x_k, u_k) + \alpha J_k(x_{k+1}) - J_k(x_k))$$

SARSA:

$$Q_{k+1}(x_k, u_k) = Q_k(x_k, u_k) + \beta_k(r(x_k, u_k) + \alpha Q_k(x_{k+1}, u_{k+1}) - Q_k(x_k, u_k))$$

Temporal Difference (TD)

$$d_k = r(x_k, u_k) + \alpha J_k(x_{k+1}) - J_k(x_k)$$

Temporal difference (TD) learning algorithms:

- TD(0): $J_{k+1}(x_k) = J_k(x_k) + \beta_k d_k$
- TD(λ): $J_{k+1}(x_k) = J_k(x_k) + \beta_k \sum_{m=k}^{\infty} \lambda^{m-k} d_m \ (0 \leq \lambda \leq 1)$
- TD(1): Monte Carlo

TD(λ)

TD(λ) from the Bellman equation.

Consider a fixed policy μ,

$$J(i) = r(i, \mu(i)) + \alpha E[J(x_{k+1})|x_k = i],$$

which can be approximated as

$$J_{k+1}(x_k) = J_k(x_k) + \beta_k(r(x_k, \mu(x_k)) + \alpha J_k(x_{k+1}) - J_k(x_k)).$$

Instead of one-step transition, consider l step transition:

$$J(i) = E\left[\sum_{m=0}^{l} \alpha^m r(i_{k+m}, \mu(i_{k+m})) + \alpha^{l+1} J(x_{k+l+1}) \middle| x_k = i\right] \triangleq f_l(i).$$

TD(λ)

$$J(i) = (1-\lambda)\sum_{l=0}^{\infty}\lambda^l f_l(i)$$

$$= (1-\lambda)E\left[\sum_{l=0}^{\infty}\lambda^l\left(\sum_{m=0}^{l}\alpha^m r_{k+m} + \alpha^{l+1}J(x_{k+l+1})\right)\middle| x_k = i\right].$$

Interchanging the order of the two summations, we obtain

$$J(i) = (1-\lambda)E\left[\sum_{m=0}^{\infty}\alpha^m r_{k+m}\sum_{l=m}^{\infty}\lambda^l + \alpha\sum_{l=0}^{\infty}(\lambda\alpha)^l J(x_{k+l+1})\middle| x_k = i\right].$$

Using the fact that $(1-\lambda)\sum_{l=m}^{\infty}\lambda^l = \lambda^m$, we have

TD(λ)

$$
\begin{aligned}
J(i) = E\Bigg[&\sum_{m=0}^{\infty}(\lambda\alpha)^m r_{k+m} + \alpha\sum_{m=0}^{\infty}(\lambda\alpha)^m J(x_{k+m+1}) \\
&- \sum_{m=0}^{\infty}(\lambda\alpha)^{m+1} J(x_{k+m+1})\Big| x_k = i\Bigg] \\
= E\Bigg[&\sum_{m=0}^{\infty}(\lambda\alpha)^m r_{k+m} + \alpha\sum_{m=0}^{\infty}(\lambda\alpha)^m J(x_{k+m+1}) \\
&- \sum_{m=1}^{\infty}(\lambda\alpha)^m J(x_{k+m})\Big| x_k = i\Bigg] \\
= E\Bigg[&\sum_{m=0}^{\infty}(\lambda\alpha)^m \left(r_{k+m} + \alpha J(x_{k+m+1}) - J(x_{k+m})\right)\Big| x_k = i\Bigg] + J(i)
\end{aligned}
$$

TD(λ)

Define $d_{k+m} = r_{k+m} + \alpha J(x_{k+m+1}) - J(x_{k+m})$.

Then,

$$\begin{aligned} J(i) &= E\left[\sum_{m=0}^{\infty} (\alpha\lambda)^m d_{k+m}\right] + J(i) \\ &= E\left[\sum_{m=k}^{\infty} (\alpha\lambda)^{m-k} d_m\right] + J(i), \end{aligned}$$

which can be approximated (with stochastic approximation) by

$$J_{k+1}(x_k) = J_k(x_k) + \beta_k \sum_{m=k}^{\infty} (\alpha\lambda)^{m-k} d_m$$

where β_k is the step-size (or learning rate) at step k.

TD(λ)

- Forward-view of TD(λ) (with a given episode and offline implementation)

$$J_{k+1}(x_k) = J_k(x_k) + \beta_k \sum_{m=k}^{\infty} (\alpha\lambda)^{m-k} d_m.$$

- Backward-view of TD(λ) (online implementation)

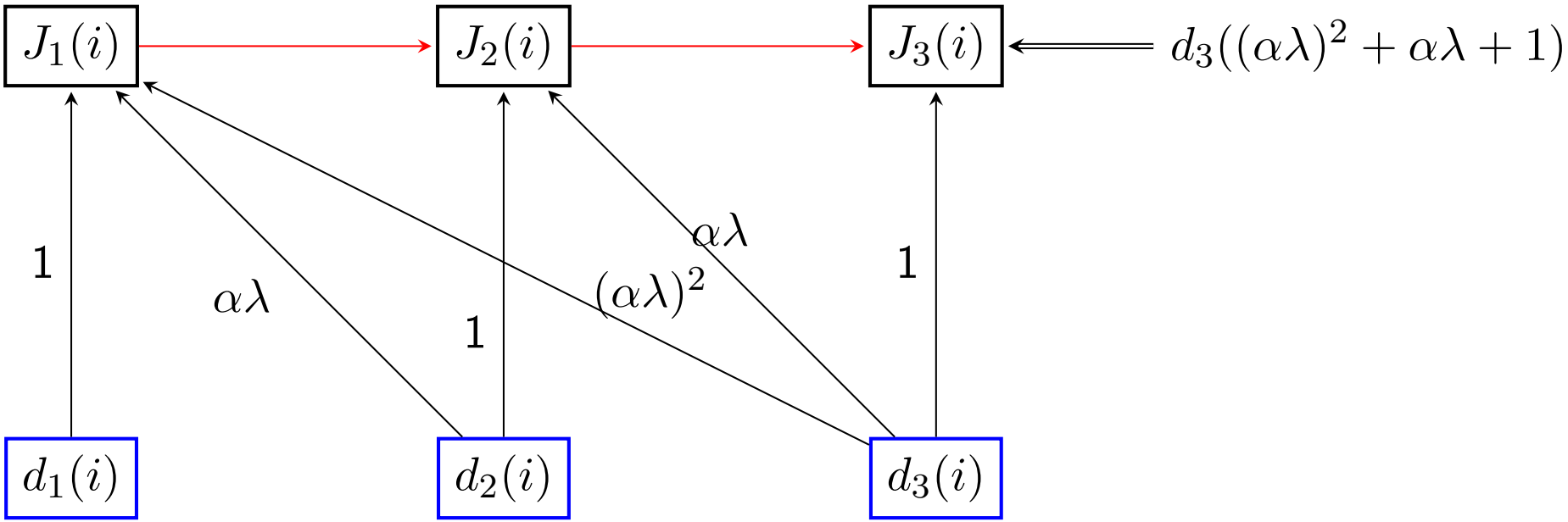
$$J_{k+1}(x_k) = J_k(x_k) + \beta_k e_k(x_k) d_k,$$

where e_k is the eligibility trace such that

$$e_k(i) = \lambda\alpha e_{k-1}(i) + \mathbb{I}_{x_k=i}.$$

Backward-View of TD(λ)

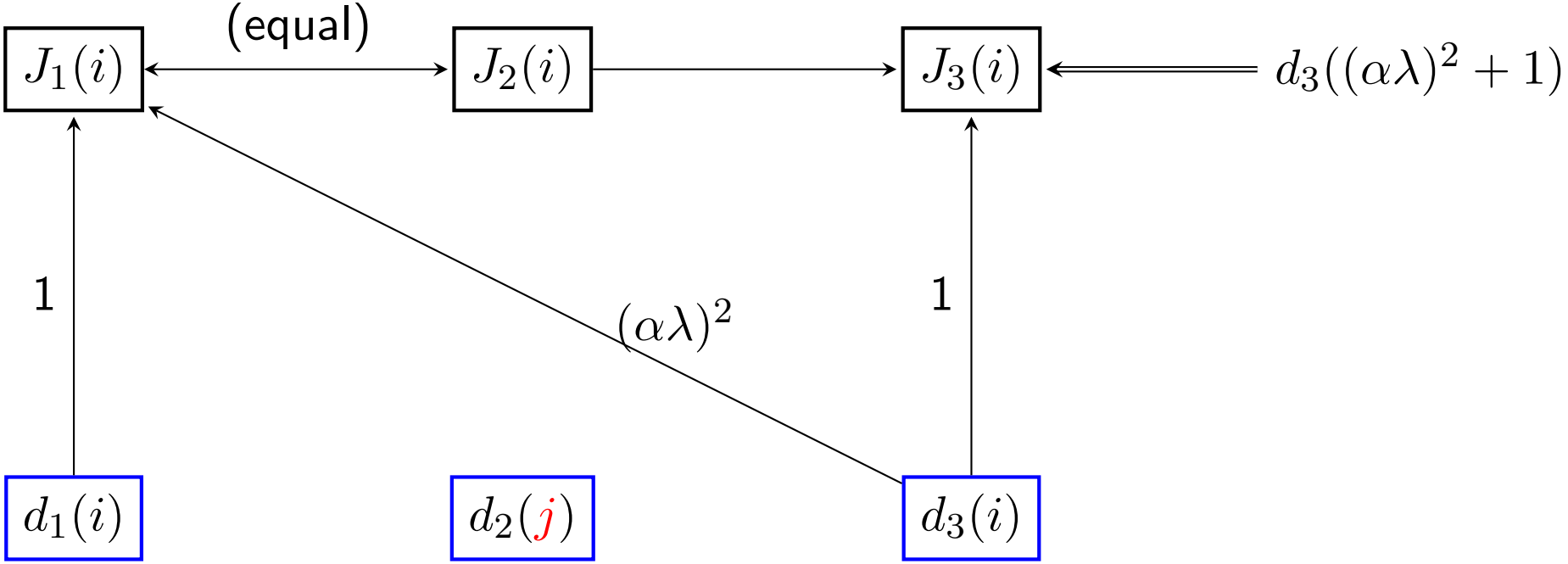
Backward view of TD(λ): Suppose $x_1 = x_2 = x_3 = i$



- $e_1(i) = 1$
- $e_2(i) = \lambda\alpha e_1(i) + 1 = 1 + \lambda\alpha$
- $e_3(i) = \lambda\alpha e_2(i) + 1 = 1 + \lambda\alpha + (\lambda\alpha)^2$

Backward-View of TD(λ)

Backward view of TD(λ): Suppose $x_1 = i,\ x_2 = j,\ x_3 = i$



- $e_1(i) = 1$
- $e_2(i) = \lambda\alpha e_1(i) = \lambda\alpha$
- $e_3(i) = \lambda\alpha e_2(i) + 1 = 1 + (\lambda\alpha)^2$

Reference

- Chapter 5.3 of Dimitri P. Bertsekas and John Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, 1996.

Acknowledgements: I would like to thank Alex Zhao for helping prepare the slides, and Honghao Wei and Zixian Yang for correcting typos/mistakes.