CS-7642 : TD(wha?)

Week 3 review: TD stuff

TD learning: motivations

Same goal:

predict values (sum of discounted rewards from a state), make policies

Different inputs: NO models! No transition matrix T, no reward matrix R (except on HW2, lucky you!)

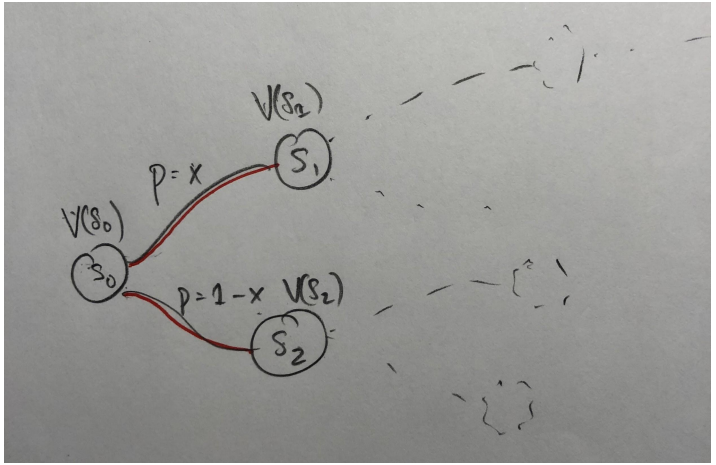
TD learning is “true” RL...we need to figure out how to behave optimally *only* by interacting with the environment

TD learning: main ideas

1. Learning V: TD learning belongs to class of RL algos that tries to estimate and improve V
2. “Bootstrapped”: we require estimations about future states in order to predict values for current state
3. Incremental: we make predictions about the final outcome of a trajectory by utilizing incremental values prior to terminal states; useful for “online” learning
4. “Forward looking” (for now): our decision about how to update each state is based on future rewards and states

TD(0): estimating by taking one step

Idea: estimate the value for a state s_{t-1} by repeatedly sampling the next next states (this may be stochastic!) and using the value of the next state to repeatedly update the value estimate of the current state



TD(0): estimating by taking one step

The “learning” update rule that reflects this *sampled* one step look ahead is:

$$V_T(s_t) = V_T(s_t) + \alpha(r_{t+1} + \gamma V_T(s_{t+1}) - V_T(s_t))$$

Which can also be modeled as an *expected value* obtained over next states:

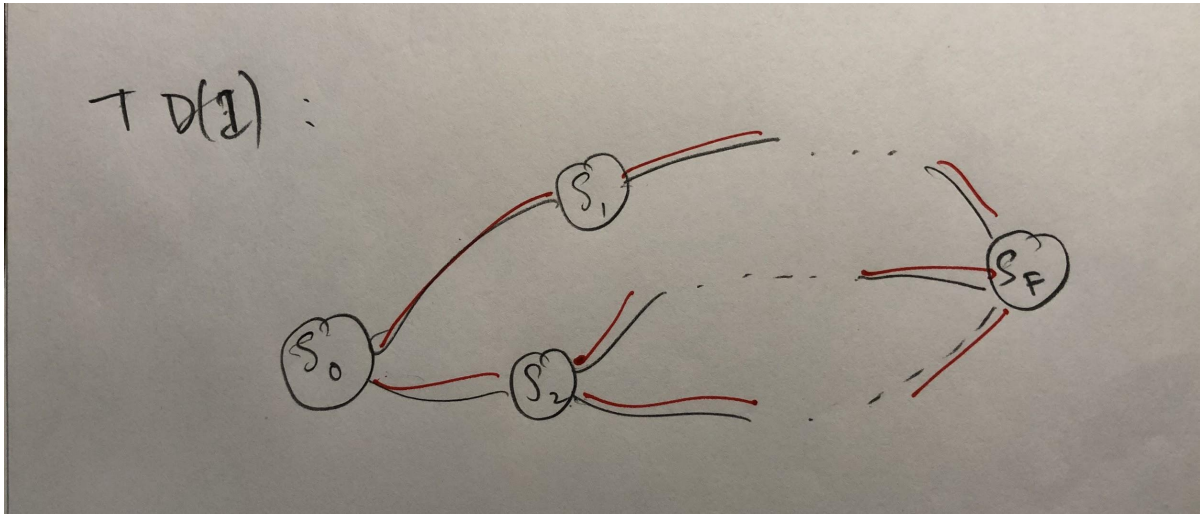
$$V_T(s_t) = E_{s_t}[r + \gamma V_T(s_{t+1})]$$

The update rule is moving the current estimate of V towards the “target” value:

$$G = R_{t+1} + \gamma V_T(s_{t+1})$$

TD(1): estimating by finishing an episode

Whereas TD(0) uses just a one state look ahead to estimate the value of the current state, TD(1) waits until it has traversed a full “trajectory” and reached a terminal state, before it updates the estimate for the value of the current state.



2-step, 3-step, k-step look aheads...

What does the 2-step “target” look like?

$$G = R_{t+1} + \gamma R_{t+2} + \gamma^2 V_T(s_{t+2})$$

What does the 3-step “target” look like?

$$G = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 V_T(s_{t+3})$$

What about the “complete” Monte Carlo, go all the way to terminal state T, target?

$$G = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots + \gamma^{T-t-1} R_T$$

TD(lambda): let's get the best of both TD(0) & TD(1)

TD(lambda) does a *weighted average* of k-step estimators E1, E2, …, Einf, according to the following:

$$TD(\lambda) = \sum_{k=1}^{\infty} \lambda^{k-1}(1-\lambda)E_k$$

Notice that if lambda is 0, then the 1-step estimator (TD(0)) gets all the weight and all the rest are set to 0, whereas if lambda is 1 then we are taking into consideration *only* the TD(1) (Monte Carlo) estimate for the value of a state, and ignoring all the other intermediate (k < inf/terminal) estimators.

References

Sutton and Barto 2nd ed: http://incompleteideas.net/book/RLbook2018trimmed.pdf