

Monte Carlo RL Review & Preview of TD Methods

Reinforcement Learning and
Planning under Uncertainty

ANU COMP6460/4640, Sem 2, 2008

Scott Sanner

NICTA / RSISE

First.Last@nicta.com.au

Essence of Monte Carlo (MC)

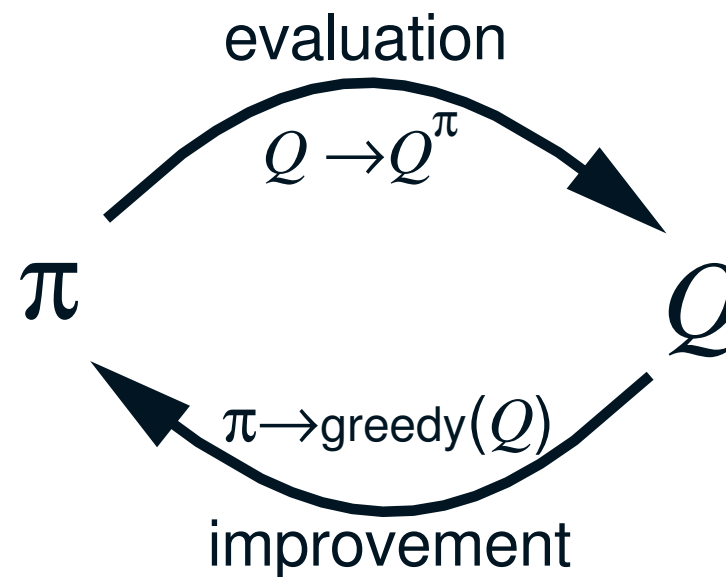
- MC samples *value expectation* directly given π

$$V_{\pi}(s) = E_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t \cdot r_t \mid s_0 = s \right]$$

- Or for *control*, sample *Q-value expectation*

$$Q_{\pi}(s, a) = E_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t \cdot r_t \mid s_0 = s, a_0 = a \right]$$

Monte Carlo Control



Generalized
policy
iteration (GPI)

- **MC policy iteration:** Policy evaluation using MC methods followed by policy improvement
- **Policy improvement step:** greedify with respect to value (or action-value) function

Need for Exploration

Where is deterministic policy in value iteration?

- For *model-based* (known) MDP solutions
 - Get convergence with *deterministic policies*
- But for *model-free* RL...
 - Need *exploration*
 - Usually use *stochastic policies* for this
 - Choose exploration action with small probability
 - Then get *convergence* to optimality

Where does exploration come in for MC control?

Why Explore?

- Why do we *need* exploration in RL?
 - Convergence requires all state/action values updated
 - Easy when model-based
 - Update any state as needed
 - Harder when model-free
 - Must be in a state to take a sample from it
 - How to get to any state in the first place? Exploration.
- Current best policy may not explore all states...
 - Must occasionally *divert from exploiting* best policy
 - Exploration ensures all reachable states/actions *updated with non-zero probability*

Key property, cannot guarantee convergence to π^* otherwise – lab!

Two Types of Exploration (of many)

- ϵ -greedy
 - Select random action ϵ of the time
 - Can decrease ϵ over time for convergence
 - But should we really select *all* actions with same probability?

- Gibbs / Boltzmann Softmax

- Still selects all actions with non-zero probability
- Draw actions from

$$P(a|Q) = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_b e^{\frac{Q(s,b)}{\tau}}}$$

- More uniform as “temperature” $\tau \rightarrow \infty$
- More greedy as $\tau \rightarrow 0$

Exploration vs. Exploitation

- How to tradeoff exploration vs. exploitation?
 - Central problem of reinforcement learning for *control*
 - Different algorithms provide different answers
 - **Generalized policy iteration**
 - a.k.a. **on-policy** algorithms
 - Need to ensure all $Q_{\pi}(s,a)$ sampled
 - » *Not* just Q-values for $Q_{\pi}(s,\pi(s))!$
 - TD Methods (incl. on-policy MC)
 - **Off-policy** algorithms
 - Learn value of policy following *some* other exploratory policy
 - » Off-policy MC
 - » Q-learning

Not an issue for simple policy evaluation... why?

Strong convergence guarantees, but generally not as efficient as on-policy methods in practice.

DP Updates vs. Sample Returns

- How to do updates?
 - Another major dimension of RL methods

- **MC** uses full sample return

$$V_{\pi}(s_t) = E_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid s_t = s \right]$$

- **TD(0)** uses sampled DP backup (bootstrapping)

$$\begin{aligned} V_{\pi}(s_t) &= \sum_{s_{t+1}} P(s_{t+1} | s_t, \pi(s_t)) [R(s_t, \pi(s_t), s_{t+1}) + \gamma V_{\pi}(s_{t+1})] \\ &= E_{s_{t+1} \sim P(s_{t+1} | s_t, \pi(s_t))} \left[r_{t+1} + \gamma V_{\pi}(s_{t+1}) \mid s_t = s \right] \end{aligned}$$

- **TD(λ)** interpolates between TD(0) and MC=TD(1)

The Great MC vs. TD(λ) Debate

- As we will see...
 - TD(λ) methods generally learn faster than MC...
 - Because TD(λ) updates value throughout episode
 - MC has to wait until termination of episode
- But MC methods are robust to MDP violations
 - non-Markovian models:
 - MC methods can also be used to evaluate semi-MDPs
 - Partially observable:
 - MC methods can also be used to evaluate POMDP controllers
 - Technically includes *value function approximation* methods

Why partially observable? B/c FA aliases states to achieve generalization.

Some Useful Distinctions

- If learning V_π (policy evaluation)
 - Just use plain MC or TD(λ); always on-policy!
- For *control* case, where learning Q_π
 - Terminology for off- vs. on-policy...

Sampling Method

	MC	TD(λ)
On-policy	MC On-policy Control (GPI)	SARSA (GPI)
Off-policy	MC Off-policy Control	Q-learning <i>if $\lambda=0$</i>

Summary: Concepts you should know

- Policy evaluation vs. control
- Exploration
 - Where needed for RL.. which of above cases?
 - Why needed for convergence?
 - ϵ -greedy vs. softmax
 - Advantage of latter?
- MC vs. TD(λ) methods
 - Differences in sampling approach?
 - (Dis)advantages of each?
- Control in RL
 - Have to learn Q-values, why?
 - On-policy vs. off-policy exploration methods

This is main web of ideas. From here, it's largely just implementation tricks of the trade.