Reinforcement Learning

Lecture 6:
Model-free Learning (1)

Yishay Mansour, Tel-Aviv University

Lecture 6: outline

- Q-learning:
  - Off-policy
  - Aprrox average online
  - DDP
  - MDP
- SARSA
  - On-policy
- Monte-Carlo
  - Methodology
  - First vs Every visit
  - Control

Online approx. of mean

- Batch updates:
  - Compute the average
  - $\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} r_t$
  - offline
- Online view
  - $\hat{\mu}_T = \frac{T-1}{T}\hat{\mu}_{T-1} + \frac{1}{T} r_T$
  - $= \hat{\mu}_{T-1} + \frac{1}{T}(r_T - \hat{\mu}_{T-1})$

- Exponential average
  - $\hat{\mu}_T$
  - $= \hat{\mu}_{T-1} + \alpha_T(r_T - \hat{\mu}_{T-1})$
  - $= \sum_{t=1}^{T} \beta_t r_t$
    - Where:
    - $\beta_t = \alpha_t \prod_{i=1}^{t-1}(1-\alpha_i)$

Concentration bounds: McDiarmid's inequality

- Setting
  - Domain X, $f: X^n \to \mathbb{R}$
- Sensitivity:
  - $c_i = \max_{x, x_i, x_i'} |f(x_1, \ldots, x_i, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots, x_n)|$
- *McDirmind's inequality*
  - $\Pr[|f(x) - E[f(x)]| \geq \epsilon] \leq 2\exp(-\frac{2\epsilon^2}{\sum_{i=1}^{n} c_i^2})$

McDiarmid's inequality: average

- $avg(x_1, \ldots, x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i$
  - $x_i \in [0,1]$
  - $c_i = \frac{1}{n}$
- Concentration Bound
  - $\Pr[|avg(x) - E[avg(x)]| \geq \epsilon] \leq 2e^{-2\epsilon^2 n}$
- Different x_i can have different expectations

McDiarmid's : weighted average

- $wavg(x_1, \ldots, x_n) = \sum_{i=1}^{n} \beta_i x_i$
  - $x_i \in [0,1]$
  - $c_i = \beta_i$
  - $\Pr[|wavg(x) - E[wavg(x)]| \geq \epsilon] \leq 2e^{-\frac{2\epsilon^2}{\sum_{i=1}^{n} \beta_i^2}}$
- Example: $\beta_t = \alpha(1-\alpha)^{t-1}$
  - $\sum_{t=1}^{T} \beta_i^2 = \alpha^2 \frac{1-(1-\alpha)^T}{1-(1-\alpha)^2} \approx \frac{\alpha}{2-\alpha}$
  - Set $\alpha \approx 4\,\epsilon^2/(\log 1/\delta)$

Lecture 6: outline

- Q-learning:
  - Off-policy
  - Aprrox average online
  - DDP
  - MDP
- SARSA
  - On-policy
- Monte-Carlo
  - Methodology
  - First vs Every visit
  - Control

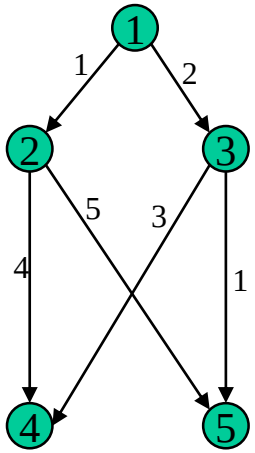
Q-learning: motivation

- Learn the optimal Q-function
  - $Q^*(s,a) = r(s,a) + \gamma E_{s' \sim p(\cdot|s,a)}[\max_a Q^*(s',a)]$
    - Discounted return
  - Q^* defines the optimal policy
- Off-policy
  - Observes some exploring policy
    - Need that each (s,a) performed infinitely often
  - Convergence: under mild conditions

Q-learning: DDP Algorithm

- Initialization: arbitrary
  - $Q_0(s, a) = 0$
- Observe (s_t, a_t, r_t, s_{t+1})
- Update:
  - $Q_{t+1}(s_t, a_t) = r_t + \gamma \max_a \{Q_t(s_{t+1}, a)\}$

Q-learning: DDP Example



$$\gamma = \frac{1}{2}$$

	initial	$2 \overset{4}{\rightarrow} 4$	$1 \overset{1}{\rightarrow} 2$	$2 \overset{5}{\rightarrow} 5$	$1 \overset{1}{\rightarrow} 2$
$1 \rightarrow 2$	0	0	3	3	3.5
$1 \rightarrow 3$	0	0	0	0	0
$2 \rightarrow 4$	0	4	4	4	4
$2 \rightarrow 5$	0	0	0	5	5
$3 \rightarrow 4$	0	0	0	0	0
$3 \rightarrow 5$	0	0	0	0	0

Q-learning: DDP

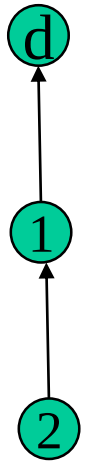
- Theorem:
  - Assume every (s, a) occurs infinitely often
  - Then: $\lim_{t \to \infty} Q_t(s, a) = Q^*(s, a)$
- Proof: Let
  - $\Delta_t = \|Q_t - Q^*\|_\infty = \max_{(s,a)} |Q_t(s, a) - Q^*(s, a)|$

- At time t:
  - $|Q_{t+1}(s_t, a_t) - Q^*(s_t, a_t)|$
  - $= |r_t + \gamma \max_{\bar{a}} Q_t(s_{t+1}, \bar{a}) - r_t - \gamma \max_{a'} Q^*(s_{t+1}, a')|$
  - $= \gamma |\max_{\bar{a}} Q_t(s_{t+1}, \bar{a}) - \max_{a'} Q^*(s_{t+1}, a')|$
  - $\leq \gamma \max_{a} |Q_t(s_{t+1}, a) - Q^*(s_{t+1}, a)|$
  - $\leq \gamma \Delta_t$
- Assume that during $[t, t_1]$each (s, a) appear at least once, then,
  - $\Delta_{t_1} \leq \gamma \Delta_t$
- $\Delta_t \to 0$ Q.E.D.

Q-learning: Shortest paths

- Shortest paths:
  - Assume a single destination
  - The value (cost) is the distance to destination
  - Initialization at very high values
    - Except for destination
- How will convergence occur?
  - By the distance from the destination
  - At any time the Q-cost upper bound real cost.

- Consider the shortest path tree
  - Exists time t_1 when nearest node to destination use the edge to destination
    - From that time its best action is fixed!
  - Exists time $t_2 \geq t_1$ when the second nearest node test the edge on the shortest path
    - From that time its best action is fixed!
  -
  -
  - Exists time t_n where all nodes …



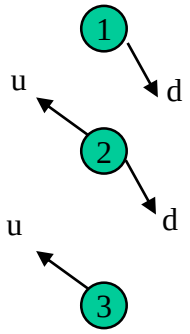
Q-learning: MDP Algorithm

- Initialization: arbitrary
  - $Q_0(s, a) = 0$
- Observe (s_t, a_t, r_t, s_{t+1})
- Update:
  - $Q_{t+1}(s_t, a_t) = Q_t(s, a) + \alpha_t(s, a)\Gamma_t$
  - $\Gamma_t = r_t + \gamma \max_a \{Q_t(s_{t+1}, a)\} - Q_t(s_t, a_t)$

Q-learning: MDP

- Intuition:
  - If $Q_t = Q^*$ then $E[\Gamma_t] = 0$
- Challenge
  - Using Q_t rather than Q^*
    - Not clear that it will converge.
  - Need to handle stochastic noise
    - Similar to stochastic gradient descent.

Q-learning: Example



		(1, d, 1,2)	(2, d, 2,2)	(2, u, 4,2)	(3, u, 3,2)	(2, u, 7,1)
$(1, d)$	0	1	1	1	1	1
$(2, u)$	0	0	0	5	5	6.25
$(2, d)$	0	0	2	2	2	2
$(3, u)$	0	0	0	0	5.5	5.5

$$\gamma = \frac{1}{2}, \alpha_t(s, a) = \frac{1}{\#(s, a)}$$

Q-learning: Convergence

- Theorem: Convergence in the limit
  - If every *(s,a)* performed infinitely often,
  - Step size, for every *(s,a):*

$$\sum_t \alpha_t(s,a) = \infty \quad \text{and} \quad \sum_t \alpha_t^2(s,a) = O(1)$$

$Q_t(s,a)$ Converges with probability 1 to $Q^*(s,a)$

Stochastic Approximation

- Iterative Algorithm:
  - $X_{t+1}(s) = \big(1 - \alpha_t(s)\big)X_t(s) + \alpha_t(s)\big((HX_t)(s) + w_t(s)\big)$
- Well behaved (B, γ):
  - Step size: $\sum_{t=1}^{\infty} \alpha(s) = \infty$ and $\sum_{t=1}^{\infty} \alpha^2(s) < \infty$
  - Noise: $E[w_t(s)] = 0$ and $|w_t(s)| \leq B$
  - Contraction: $\exists X^*$: $\|HX - X^*\| \leq \gamma \|X - X^*\|$
    - $HX^* = X^*$

Stochastic Approximation

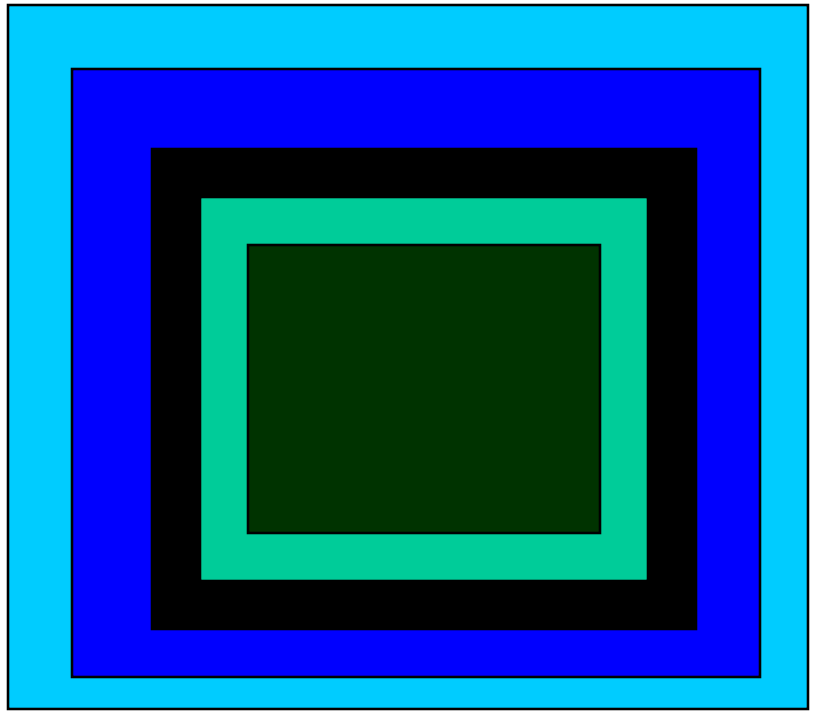
- Theorem:
  - If X_t generated by a well behaved (B, γ)
  - Then $X_t \rightarrow X^*$
    - With probability 1

Stoch Approx: proof methodology

- Partition the error to two parts:
  - One which contracts, HX_t
  - One which is noise, w_t
    - the difference of the expected and sampled values.
- The first drops deterministically (as expected).
- The second has zero expectation,
  - bound it using law of large numbers.
    - needs to be bounded for an **entire** next phase.

SA: Classical Proof Methodology

Let $\Delta_t = Q_t - Q^*$



$$D_0 = V_{\max}$$

$$D_1 = (1-\beta)D_0$$

$$D_2 = (1-\beta)D_1$$

$$D_3 = (1-\beta)D_2$$

$$D_4 = (1-\beta)D_3$$

Q-learning: contraction

- Operator:
  - $(Hq)(s,a) = \sum_{s'} p(s'|s,a)[r(s,a) + \gamma \max_{b \in A} q(s',b)]$
- Contraction:
  - $\|Hq_1 - Hq_2\|_\infty$
  - $= \gamma \max_{s,a} |\sum_{s'} p(s'|s,a)[\max_{b_1 \in A} q_1(s',b_1) + \max_{b_2 \in A} q_2(s',b_2)]$
  - $\leq \gamma \max_{s,a} \max_{b,s'} |q_1(s',b) - q_2(s',b)|$
  - $\leq \gamma \|q_1 - q_2\|_\infty$

Q-learning: convergence

- Rewrite Q-learning:
  - $Q_{t+1}(s_t, a_t) = (1 - \alpha_t(s_t, a_t))Q_t(s_t, a_t) + \alpha_t(s_t, a_t)\Phi_t$
  - $\Phi_t = r_t + \gamma \max_a \{Q_t(s_{t+1}, a)\}$
  - $E[\Phi_t] = (HQ_t)(s_t, a_t)$
  - $w_t(s_t, a_t) = \Phi_t - (HQ_t)(s_t, a_t)$
  - $E[w_t] = 0$, and $|w_t| \leq \frac{R_{max}}{1-\gamma}$
  - $\Phi_t = (HQ_t)(s_t, a_t) + w_t$

Q-learning: Convergence

- Use *stochastic approximation* convergence
  - Step size: by assumption
  - Noise: $E[w_t] = 0$ & $|w_t| \leq V_{max} = \frac{R_{max}}{1-\gamma}$
  - Contraction:
    - Use Q^* and operator H is γ-contracting
- Result: Q_t converges to Q^* with prob. 1
  - Assumes "exploration"

Q-learning: Step size

- Step size: $\alpha(s,a) = \frac{1}{g(\#(s,a))}$
- Linear: $g(n) = n$
  - $\sum_{n=1}^{N} \frac{1}{n} \approx \ln N$ and $\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$
- Poly: $g(n) = n^{\theta}$ for $\theta \in (\frac{1}{2}, 1)$
  - $\sum_{n=1}^{N} \frac{1}{n^{\theta}} \approx \frac{N^{1-\theta}}{1-\theta}$ and $\sum_{n=1}^{\infty} \frac{1}{n^{2\theta}} \leq \frac{1}{2\theta-1}$

Q-learning: Linear step size

- Bad example for linear step size
- Single state s and action a with $r(s,a) = 0$
- Q-learning initialized $Q_0(s,a) = 1$
  - $Q_t = \left(1 - \frac{1}{t}\right)Q_{t-1} + \frac{1}{t}(0 + \gamma Q_{t-1}) = (1 - \frac{1-\gamma}{t})Q_{t-1}$
  - $Q_t = \Theta(t^{\gamma-1})$
  - $t = c\left(\frac{1}{\epsilon}\right)^{\frac{1}{1-\gamma}}$ we still have $Q_t \geq \epsilon$

Q-learning: Poly step size

- Single state s and action a with $r(s,a) = 0$
- Q-learning initialized $Q_0(s,a) = 1$
  - $Q_t = \left(1 - \frac{1}{t^\theta}\right) Q_{t-1} + \frac{1}{t^\theta}(0 + \gamma Q_{t-1}) = (1 - \frac{1-\gamma}{t^\theta}) Q_{t-1}$
  - $Q_t = \Theta\left(e^{-(1-\gamma)t^{1-\theta}}\right)$
  - For $t = c\frac{1}{1-\gamma}\log^{1/(1-\theta)}\frac{1}{\epsilon}$ we have $Q_t \leq \epsilon$

Lecture 6: outline

- Q-learning:
  - Off-policy
  - Aprrox average online
  - DDP
  - MDP
- SARSA
  - On-policy
- Monte-Carlo
  - Methodology
  - First vs Every visit
  - Control

SARSA: on-policy

- Run Q-learning with on-policy
  - Need to select actions
- Policy Requirement:
  - Need to explore
    - Each (s, a) visited infinitely often
  - Need to be Greedy
    - In the limit

SARSA: Algorithm

- Initialization: arbitrary
  - $Q_0(s, a) = 0$
- Observe $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$
  - $a_{t+1} = \pi(s_{t+1}; Q_t)$ output of the on-policy
- Update:
  - $Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t(s_t, a_t)\Gamma_t$
  - $\Gamma_t = r_t + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)$

SARSA: exploration

- How to select the action $\pi(s; Q_t)$
  - Let $\bar{a} = \arg\max_a Q_t(s, a)$
- ϵ_t-greedy:
  - With prob $1 - \epsilon_t$ we have $\pi(s; Q_t) = \bar{a}$
  - $\forall a \in A$: with prob $\frac{\epsilon_t}{|A|}$ we have $\pi(s; Q_t) = a$
  - Value for ϵ_t: $\epsilon_t = \frac{1}{t}$; $\epsilon_t = \frac{1}{t^\theta}$

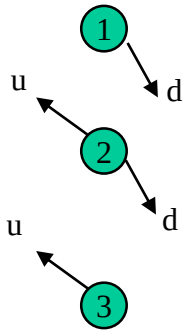
SARSA: exploration

- How to select the action $\pi(s; Q_t)$
  - Let $\bar{a} = \arg\max_a Q_t(s, a)$
- Soft-max:
  - $\forall a \in A$: we have $\pi(s; Q_t) = a$ with prob

$$\frac{e^{\beta_t Q_t(s,a)}}{\sum_{a' \in A} e^{\beta_t Q_t(s,a')}}$$

  - Values for $\beta_t \to \infty$ for $t \to \infty$
    - But slowly

SARSA: Example



		$(1, \mathrm{d}, 1, 2, u)$	$(2, \mathrm{u}, 2, 2, d)$	$(2, \mathrm{d}, 4, 3, u)$	$(3, \mathrm{u}, 3, 2, u)$	$(2, \mathrm{u}, 7, 1, d)$
$(1, d)$	0	1	1	1	1	1
$(2, u)$	0	0	2	2	2	4.75
$(2, d)$	0	0	0	4	4	4
$(3, u)$	0	0	0	0	4	4

$$\gamma = \frac{1}{2}, \alpha_t(s,a) = \frac{1}{\#(s,a)}, \epsilon = \frac{1}{10}$$

SARSA: convergence

- Convergence of Q_t
  - Follows from Q-learning assuming exploration
    - Need enough exploration
- Convergence of return
  - Need to converge to greedy
    - While keeping exploration
  - This is different from Q-learning !

SARSA: sources of errors

- SARSA: sources for errors
  - Sampling
    - Next state,
    - Rewards
  - Policy
    - Stochastic exploration

Expected SARSA: Algorithm

- Initialization: arbitrary
  - $Q_0(s, a) = 0$
- Update:
  - $Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t(s_t, a_t)\Gamma_t$
  - $\Gamma_t = r_t + \gamma E_{a\sim\pi} Q_t(s_{t+1}, a) - Q_t(s_t, a_t)$
- Sample
  - $a_{t+1} = \pi(s_{t+1}; Q_t)$

Lecture 6: outline

- Q-learning:
  - Off-policy
  - Aprrox average online
  - DDP
  - MDP
- SARSA
  - On-policy
- Monte-Carlo
  - Methodology
  - First vs Every visit
  - Control

Monte-Carlo

Monte Carlo: motivation

- Directly learn from experience
- Model free
- Simplest idea:
  - Average the returns
  - Learn a value function
  - No dependencies
    - Both plus and minus
- Mainly for episodic MDP

Monte Carlo: basic idea

- Learn V^{π} from episodes generated by π
  - Episode: $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_k, a_k, r_k$
    - $a_t \sim \pi(\cdot \mid s_t)$
  - Return (of an episode):
    - $V^{\pi}(s) = E^{\pi}[\sum_{t=1}^{k} r_t \mid s_1 = s]$
    - Observation: $G(s) = \sum_{t=1}^{k} r_t$

Monte Carlo: Algorithm

- Given observations
  - $G_1(s), \ldots, G_m(s)$
  - Estimate $\hat{V}^{\pi}(s) = \frac{1}{m}\sum_{i=1}^{m} G_i\,(s)$
- How do we generate the samples:
  - How to define $G_i(s)$ from episodes

MC: Generating samples

- Initial state:
  - Use only the initial state of the trajectory
  - Given Episode: $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_k, a_k, r_k$
  - Update only $\hat{V}^{\pi}(s_1)$
- Weaknesses?
  - Not clear that we will reach all states
  - Very wasteful
    - Can we do better

MC: First visit

- Given the i^{th} episode:
  - $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_k, a_k, r_k$
  - For each state s_j that appears in the episode
  - Update $\hat{V}^{\pi}(s_j)$ once
    - Use only the first occurrence of s_j
      - Assume that this is the m-th state
    - $G_i(s_j) = \sum_{l=m}^{k} r_l$

MC: Every visit

- Given the i^{th} episode:
  - $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_k, a_k, r_k$
  - update **each** appearance of state s_j
  - Update $\hat{V}^{\pi}(s_j)$ using multiple suffixes
    - For each occurrence as the m-th state
    - $G_i(s_j) = \sum_{l=m}^{k} r_l$
- A state updated multiple times in an episode
  - Updates are correlated!

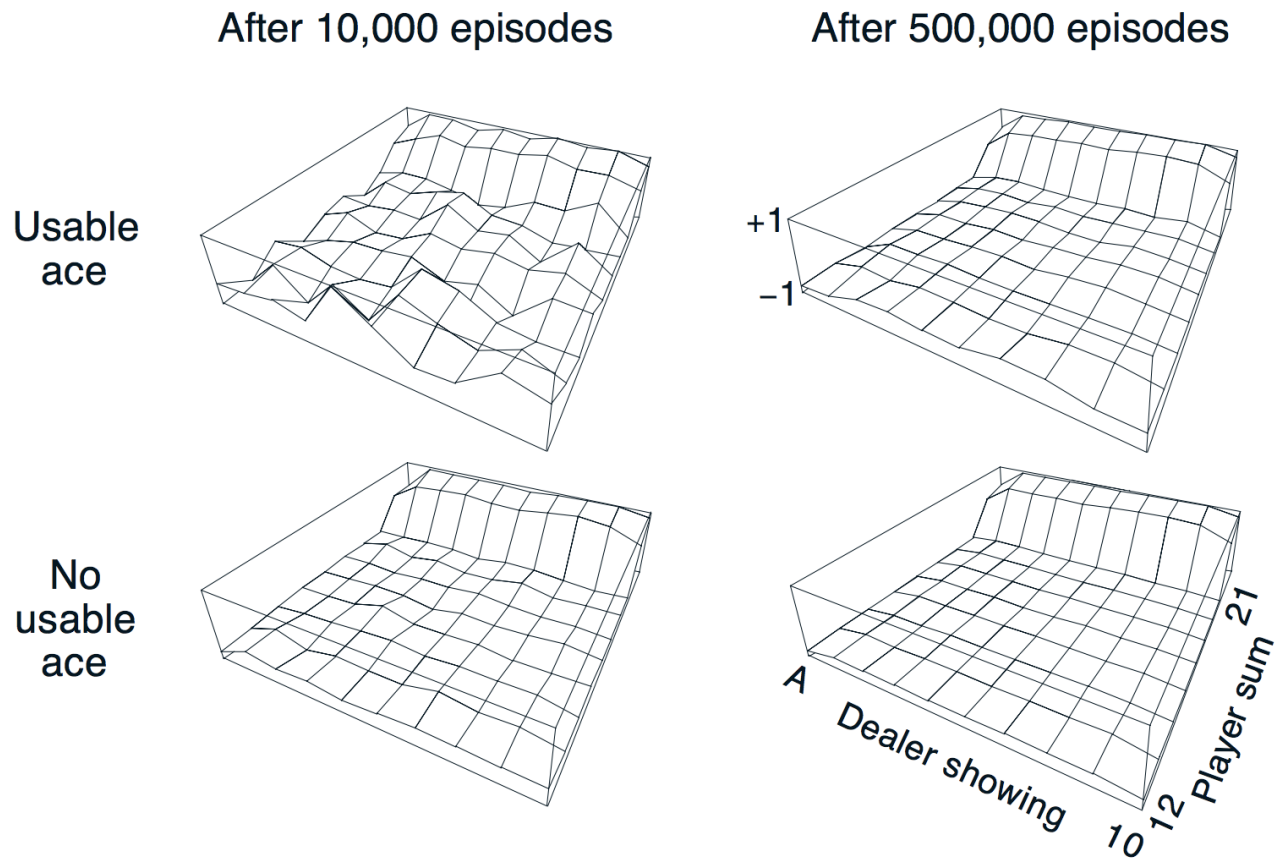
Example: blackjack

- States (about 200)
  - Current sum (12-21)
  - Dealer card (ace, 2-10)
  - Do I have “usable ace”
- Actions
  - stick: stop
  - Twist: add a card
- House: stick on 16

- Reward
  - Stick:
    - +1 winner
    - 0 tie
    - -1 loser
  - Twist
    - -1 over 21
    - 0 otherwise

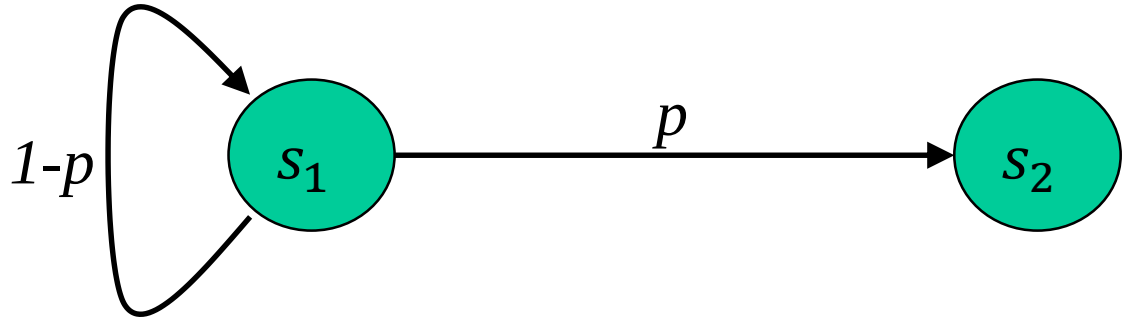
MC: Blackjack simulation

Policy:
Stick on
20 or 21



First versus Every visit

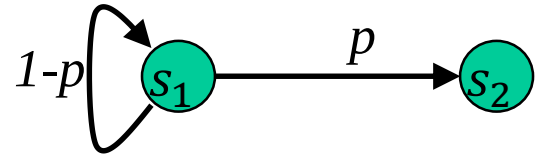
Test case:



- Rewards are +1
- $V(s_1) = \frac{1}{p}$

First versus Every visit

- We observe a trajectory:
  - s_1, s_1, s_1, s_1, s_2
- What would be your estimate for $V(s_1)$
  - What about p?
- First visit: $\hat{V}(s_1) = 4$
- Every visit: $\hat{V}(s_1) = \frac{4+3+2+1}{4} = 2.5$



Maximum Likelihood model

- Maximum Likelihood model:
  - $p^* = \arg\max(1-p)^3 p$
    - $(1-p^*)^3 - 3(1-p^*)^2 p^* = 0$
  - $p^* = \frac{1}{4}$
- Expected value:
  - $V(s_1) = \frac{1}{p^*} = 4$
  - Coincides here with first visit
    - but not always!

Max Likelihood and First visit

- Consider a state *s*
- First visit when updating *s* ignores:
  - Trajectories that not include *s*
  - All states up to first occurance of *s*
- Reduced sample: ignore those parts
- Theorem:

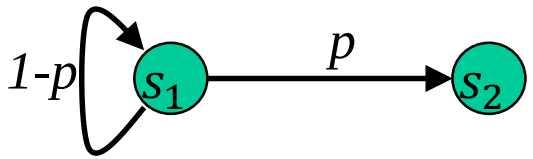
First Visit identical to MaxLikelihood on reduced sample.

Every visit minimizes MSE

- Recall:
  - SE: $\sum_{s_{i,j}} \left(\hat{V}(s_{i,j}) - R(s_{i,j})\right)^2$
  - $\sum_s \sum_{i,j:s_{i,j}=s} \left(\hat{V}(s) - R(s_{i,j})\right)^2$
  - The minimizer $V'(s) = \frac{\sum_{i,j:s_{i,j}=s} R(s_{i,j})}{\{(i,j):s_{i,j}=s\}|}$
    - Exactly MC Every visit.

First versus Every visit

- First visit: Unbiased
  - episode n times s_1
  - $FV = n$
  - $E[FV] = \frac{1}{p}$

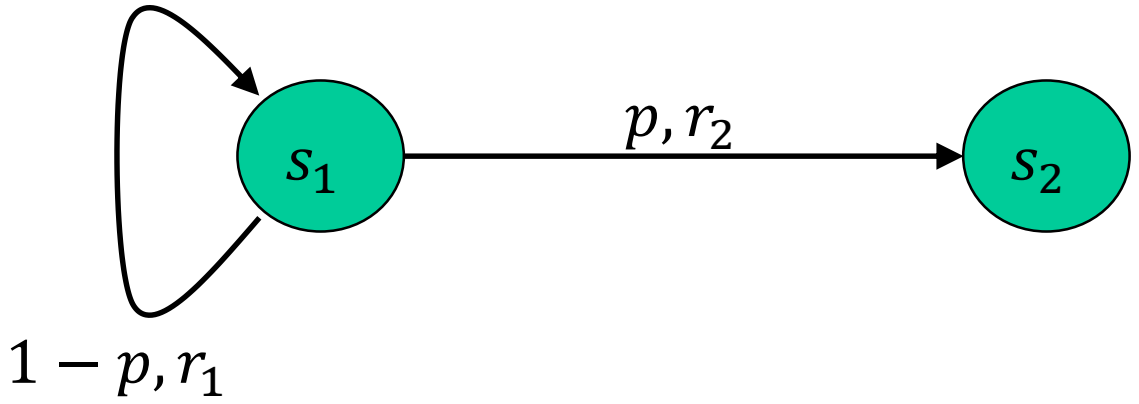


- Every visit: Biased
  - episode n times s_1
  - $EV = \frac{n(n+1)}{2}\frac{1}{n} = \frac{n+1}{2}$
  - $E[EV] = \frac{1}{2p} + \frac{1}{2}$
- Multiple episodes
  - $EV = \frac{sum\ episodes\ s_1}{\#(s_1)}$
  - $E[EV] = \frac{E[\frac{n(n+1)}{2}]}{E[n]} = \frac{1}{p}$

MC: Analysis using 2-state

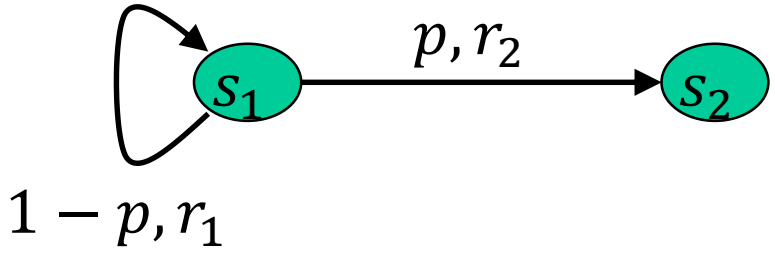
- Fix a state s
  - Partition trajectory with s

  - Build two-state MDP



MC Every Visit: Analysis

- True value
  - $V(s) = \frac{1-p}{p} r_1 + r_2$



- Every visit single episode
  - $E\left[\hat{V}(s)\right] = \frac{1-p}{2p} r_1 + r_2$
- Every visit m episodes
  - $E\left[\hat{V}(s)\right] = \frac{m}{m+1} \frac{1-p}{p} r_1 + r_2$

Monte Carlo: control

- Learn Q-function
- For every (s, a):
  - Update $\hat{Q}(s, a)$ using episodes
    - Either first or every visit
- Control:
  - Update policy π to be near-greedy
    - For example, ϵ-greedy
  - Note: always using the updates Q!

Policy improvement: Epsilon-greedy

- Recall:

$$\pi(a|s) = \begin{cases} \frac{\epsilon}{|A|} + (1-\epsilon) & for\ a = \bar{a} \\ \frac{\epsilon}{|A|} & otherwise \end{cases}$$

- Policy improvement:
  - $\pi'(\cdot\,|s)$ sets $\bar{a} = \arg\max_a Q^{\pi}(s,a)$

Policy improvement: Epsilon-greedy

- Theorem: For any ϵ-greedy π the ϵ-greedy improvement policy π' has $V^{\pi'} \geq V^{\pi}$
- Proof:
  - $E_{a\sim\pi'}[Q^{\pi}(s,a)] = \sum_{a\in A} \pi'(a|s) Q^{\pi}(s,a)$
  - $= \frac{\epsilon}{A} \sum_{a\in A} Q^{\pi}(s,a) + (1-\epsilon) Q^{\pi}(s,\bar{a})$
  - $\geq \frac{\epsilon}{A} \sum_{a\in A} Q^{\pi}(s,a) + (1-\epsilon) \sum_{a} \frac{\pi(a|s) - \epsilon/|A|}{1-\epsilon} Q^{\pi}(s,a)$
  - $= \sum_{a\in A} \pi(a|s) Q^{\pi}(s,a) = V^{\pi}(s)$

Monte Carlo: pros and cons

- Pros:
  - Does not assume Markovian environment
  - Extends naturally for function approximation
  - Unbiased (first visit)
- Cons:
  - Update only at the end of a complete episode
  - Biased (every visit)

Lecture 6: outline

- Q-learning:
  - Off-policy
  - Aprrox average online
  - DDP
  - MDP
- SARSA
  - On-policy
- Monte-Carlo
  - Methodology
  - First vs Every visit
  - Control