



Reinforcement Learning

Lecture 12: Monte Carlo Control

Sep 2, 2025

Instructor: Raunak Bhattacharyya



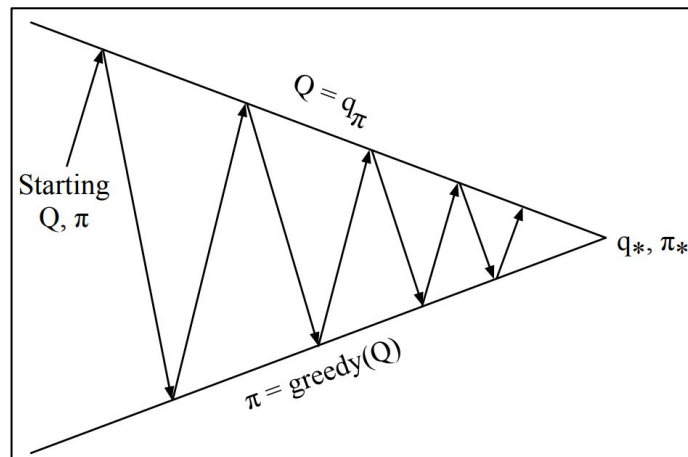
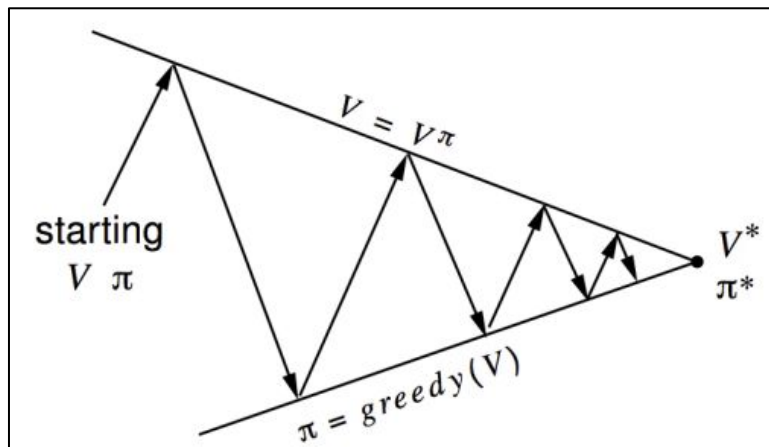
ScAI

YARDI SCHOOL OF ARTIFICIAL INTELLIGENCE
INDIAN INSTITUTE OF TECHNOLOGY DELHI

Recap

- Bias & Variance
- MC and TD in batch settings

GPI with MC evaluation



$$\pi'(s) = \underset{a}{\operatorname{argmax}} \mathbb{E} [R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s, A_t = a]$$

Recall: Monte Carlo Prediction

First-visit MC prediction, for estimating $V \approx v_\pi$

Input: a policy π to be evaluated

Initialize:

$V(s) \in \mathbb{R}$, arbitrarily, for all $s \in \mathcal{S}$

$Returns(s) \leftarrow$ an empty list, for all $s \in \mathcal{S}$

Loop forever (for each episode):

Generate an episode following π : $S_0, A_0, R_1, S_1, A_1, R_2, \dots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \dots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless S_t appears in $S_0, S_1, \dots, S_{t-1}$:

Append G to $Returns(S_t)$

$V(S_t) \leftarrow \text{average}(Returns(S_t))$

Will not work for Q estimation

Plan

- Monte carlo control
- Exploration
- Policy improvement

Monte Carlo Control: Exploring Starts

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize:

$\pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$

$Q(s, a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$

$Returns(s, a) \leftarrow$ empty list, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$

Loop forever (for each episode):

Choose $S_0 \in \mathcal{S}, A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0

Generate an episode from S_0, A_0 , following π : $S_0, A_0, R_1, \dots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \dots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1, \dots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

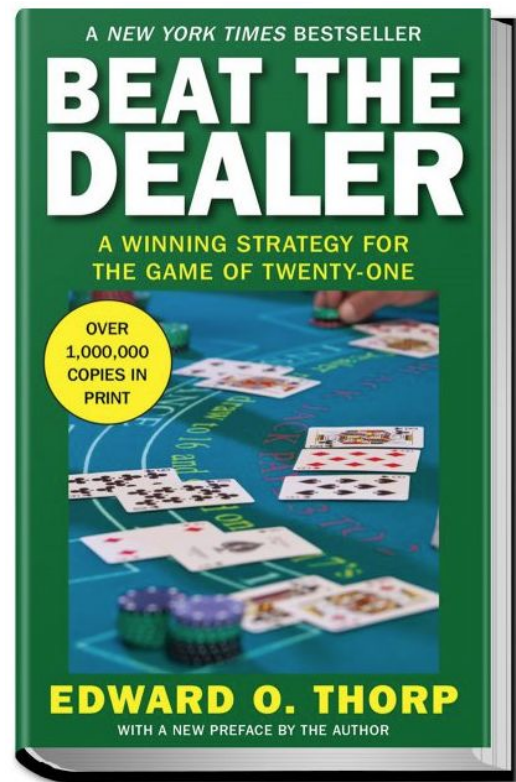
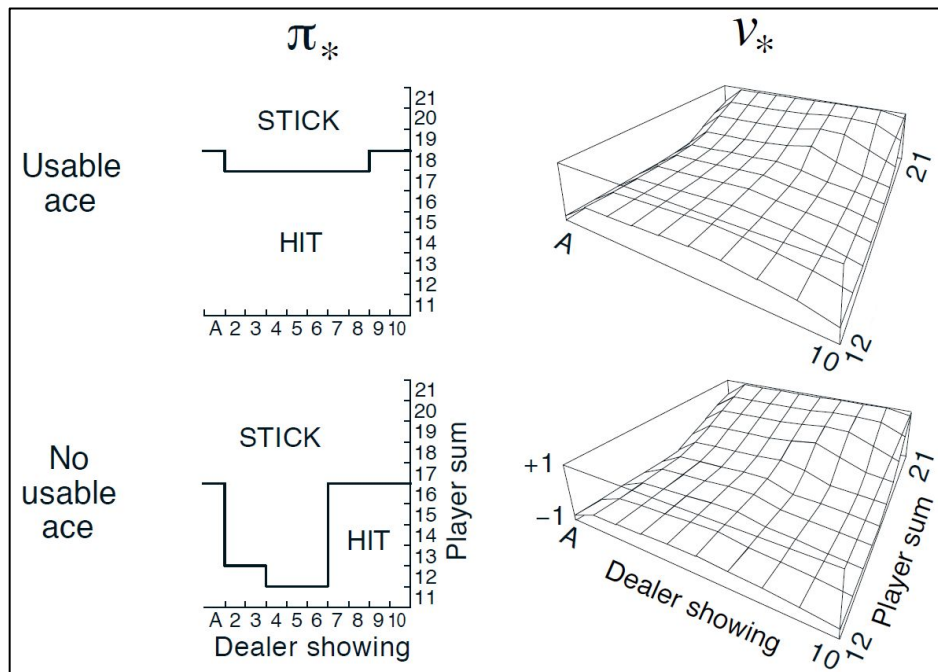
$\pi(S_t) \leftarrow \arg\max_a Q(S_t, a)$

What's the interleaving
between evaluation and
improvement processes?

After every episode

Policy Improvement?

Blackjack: Monte Carlo ES



[Source](#)

Tetris



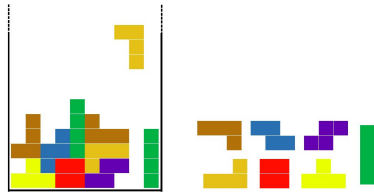
[Source](#)

- Goal: Find a policy that maximises the average (over multiple games) number of lines removed in a game
 - Shown to be a finite number



[Source](#)

Tetris: MDP Components



- State:
 - Board configuration
 - Shape of block (tetromino)
- Action: Placement $|\mathcal{A}| \leq 32$
- Reward: Number of rows eliminated
- Dynamics:
 - Wall change
 - Random next tetromino

- Board is 10x20. And every square could be filled/not filled. Pieces of 7 different shapes

$$\text{about } 2^{200} \simeq 1.6 \times 10^{60}$$

- Even if sequence of pieces is known in advance, finding the optimal strategy is NP hard

Various Approaches

Approximate Dynamic Programming Finally Performs Well in the Game of Tetris

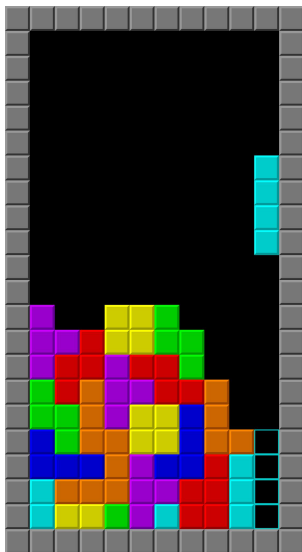
Victor Gabillon
INRIA Lille - Nord Europe,
Team SequeL, FRANCE
victor.gabillon@inria.fr

Mohammad Ghavamzadeh*
INRIA Lille - Team SequeL
& Adobe Research
mohammad.ghavamzadeh@inria.fr

Bruno Scherrer
INRIA Nancy - Grand Est,
Team Maia, FRANCE
bruno.scherrer@inria.fr

- Tsitsiklis and Van Roy (1996):
 - Approximate value iteration 30-40 lines
 - Two features: board height, number of holes in board
- Bertsekas and Ioffe (1996):
 - Lambda policy iteration 4000 lines
 - Elaborate set of 22 features
- Farias and Van Roy (2006):
 - Linear programming
- Kakade (2001):
 - Natural policy gradient 4700, 6800
- Furmston and Barber (2012):
 - Newton method in policy space 14000 lines

Monte Carlo: Without Exploring Starts



How do we do exploring starts in tetris?

In Monte-Carlo ES and control, the policy was set to greedy w.r.t. Q . End of exploration

For all actions to continue to be selected infinitely often, the policy has to continue to select them

ϵ - greedy policy:

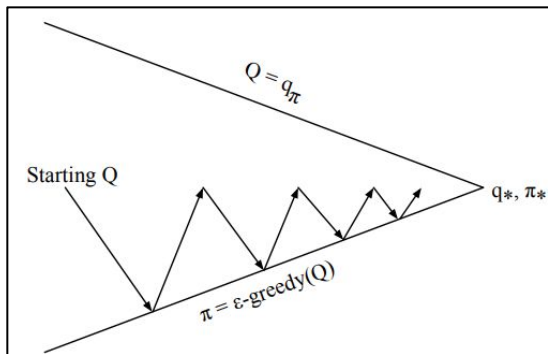
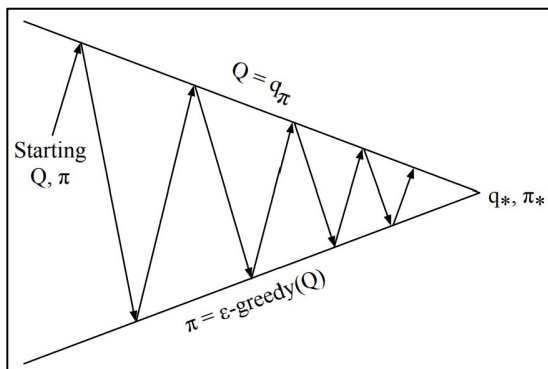
For all actions to continue to be selected infinitely often,
the policy has to continue to select them

$$\pi_{\text{new}} = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}|} & \text{if } a = \arg \max_a Q(s, a) \\ \frac{\epsilon}{|\mathcal{A}|} & \text{otherwise} \end{cases}$$

$$\pi(a|s) \geq \frac{\epsilon}{|A(s)|}$$

More general: epsilon-soft policy

Monte Carlo: Stochastic Exploration Policy



On-policy first-visit MC control (for ϵ -soft policies), estimates $\pi \approx \pi_*$

Algorithm parameter: small $\epsilon > 0$

Initialize:

$\pi \leftarrow$ an arbitrary ϵ -soft policy

$Q(s, a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$

$Returns(s, a) \leftarrow$ empty list, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$

Repeat forever (for each episode):

Generate an episode following π : $S_0, A_0, R_1, \dots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

Loop for each step of episode, $t = T-1, T-2, \dots, 0$:

$G \leftarrow \gamma G + R_{t+1}$

Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1, \dots, S_{t-1}, A_{t-1}$:

Append G to $Returns(S_t, A_t)$

$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$

$A^* \leftarrow \arg\max_a Q(S_t, a)$

(with ties broken arbitrarily)

For all $a \in \mathcal{A}(S_t)$:

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \epsilon + \epsilon/|\mathcal{A}(S_t)| & \text{if } a = A^* \\ \epsilon/|\mathcal{A}(S_t)| & \text{if } a \neq A^* \end{cases}$$

Update the mean return of just those state, action pairs visited in that episode, and then improve policy immediately

Policy Improvement?

GLIE

Conditions to get to the optimal policy?

- Explore: No FOMO
- Be deterministic ... eventually

$$q_*(s, a) = \mathbb{E} \left[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \mid S_t = s, A_t = a \right]$$