Aalto University
School of Science

Decision recommendations with help of simulation-based algorithms

Tommi Summanen
Presentation *17*

MS-E2191 Graduate Seminar on Operations Research
Fall 2020

Table of contents

Example 1: Blackjack card game 1/2

- **Dealer and player play against each other and both are dealt two cards. Player's cards are face up and one of dealer's card is face up.**
- **Goal is to have cards' sum as close to 21 as possible but not above. Both can pick new cards. If sum goes above 21 player or dealer loses automatically.**
- **Face card have value 10 and other cards have value according to their number.**
- **To simplify let's assume card deck with replacement and fix ace to have only value 1.**

Example 1: Blackjack card game 2/2

- **What is winning probability if dealer's visible card is 5, and your policy is to stop taking new cards after your sum is 15, and dealer's policy is to stop after sum is 17?**
- **Monte Carlo simulation with N = 1 000 000 gives**
  - P(win) ≈ 49,4%
  - P(draw) ≈ 6,6%
  - P(loss) ≈ 44,0 %

Monte Carlo Prediction algorithm

- **Answers question how to evaluate value function V(s) (expected return of state) under given policy π**
- **We need sample sequences $S_0, A_0, R_0, \ldots, S_T, A_T, R_T$ from measurements or simulations**
- **We assume that sample is divided into episodes and each episode terminates**

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: $S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_T$

$G \leftarrow 0$

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

$G \leftarrow \gamma G + R_{t+1}$ ← This looks quite familiar…

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

Append G to $Returns(S_t)$

$V(S_t) \leftarrow \text{average}(Returns(S_t))$ ← Update value of state!

Sutton, R. S., & Barto, A. G. (2018).

Monte Carlo ES (Exploring starts) algorithm

- **But how to find optimal policy?**
- **Values of states are not enough, we have to also estimate reward for each state-transition pair to be able to optimize policy on each step**
- **Problem: how to make sure that each state-transition pair is visited?**
  - Assume that episode starts in state-action pair and there is nonzero probability for each to occur

Monte Carlo ES

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 ← **Exploring starts!**

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

$G \leftarrow 0$

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

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

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

Append G to $Returns(S_t, A_t)$

$Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$ ← **Update value of state-transition!**

$\pi(S_t) \leftarrow \text{argmax}_a\, Q(S_t, a)$ ← **Improve policy!**

Sutton, R. S., & Barto, A. G. (2018).

Monte Carlo ES

- **Assuming exploring starts can be problematic considering real live systems**
- **We can get rid of this assumption by finding policy among ε-soft policies**
  - When policy is improved give probability **ε/|A(s)|** for each non optimal action
  - …and rest of the probability for optimal action

Monte Carlo using ε-soft policies

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

Algorithm parameter: small $\varepsilon > 0$

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: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$ ← **No exploring starts!**

$G \leftarrow 0$

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

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

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

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

$$\pi(a|S_t) \leftarrow \begin{cases} 1 - \varepsilon + \varepsilon/|\mathcal{A}(S_t)| & \text{if } a = A^* \\ \varepsilon/|\mathcal{A}(S_t)| & \text{if } a \neq A^* \end{cases}$$ ← **Update ε-soft policy**

Note that…

- **Probability distributions of transitions don't need to be known when using Monte Carlo methods!**
- **Computational cost for single state is independent from number of states!**
- **Slight difference between *first visit* method and *every visit method***
- **Here we considered only *on-policy* methods where policy that is used to generate states is simultaneously improved. There are also *off-policy* methods where different policy is improved than the one that is used to generate data.**

References

Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press. From: http://incompleteideas.net/book/RLbook2018.pdf. Accessed 9.11.2020.

Homework: Blackjack revisited 1/2

I. **Recall our Blackjack example. Calculate winning probabilities when dealer's threshold is 17 and dealer's visible card is in range [1,10] and player's threshold belongs to range [12,21].**

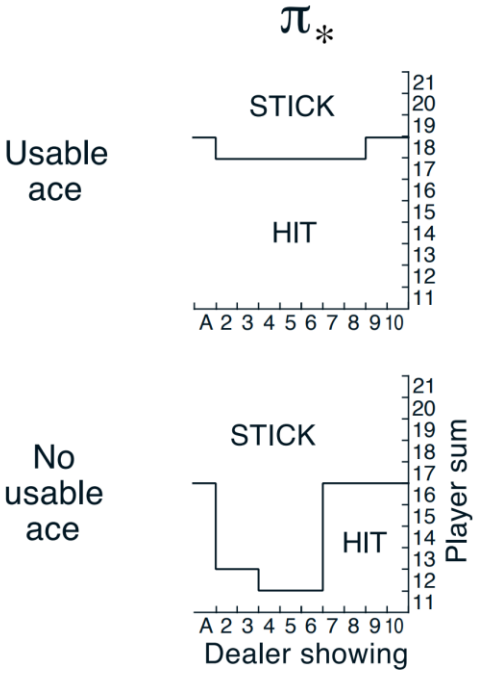
II. **Formulate optimal policy based on simulations i.e. give optimal threshold for player for each possible dealer's visible card.**

Homework: Blackjack revisited 2/2

III. Graph on the right shows optimal policy that Sutton and Barto obtained with Monte Carlo ES simulation where player could choose whether ace in player's hand had value 11 or 1. Graph on the top shows situation where player has usable ace and one on the bottom where player doesn't have ace or can't give ace value 11 because player would go above 21. Comment on the difference of your results and Sutton's and Barto's results.

Submit your answers to tommi.summanen@aalto.fi

Please send any comments and questions to Telegram group :)



Sutton, R. S., & Barto, A. G. (2018).