Pls edit and improve I have no idea what im doing

1. MC

1.a)

epsilon greedy (D) -> shouldn’t this imply C as well since Epsilon-Greedy is Greedy?

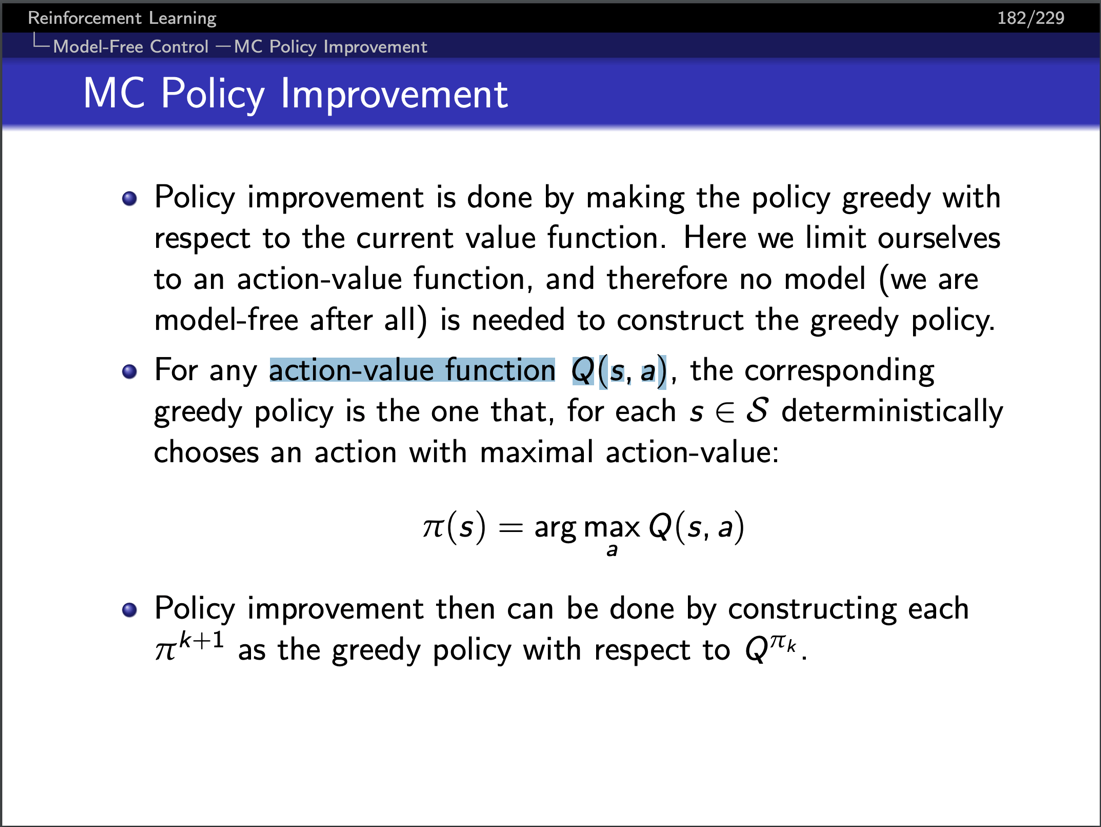
On-policy (A)

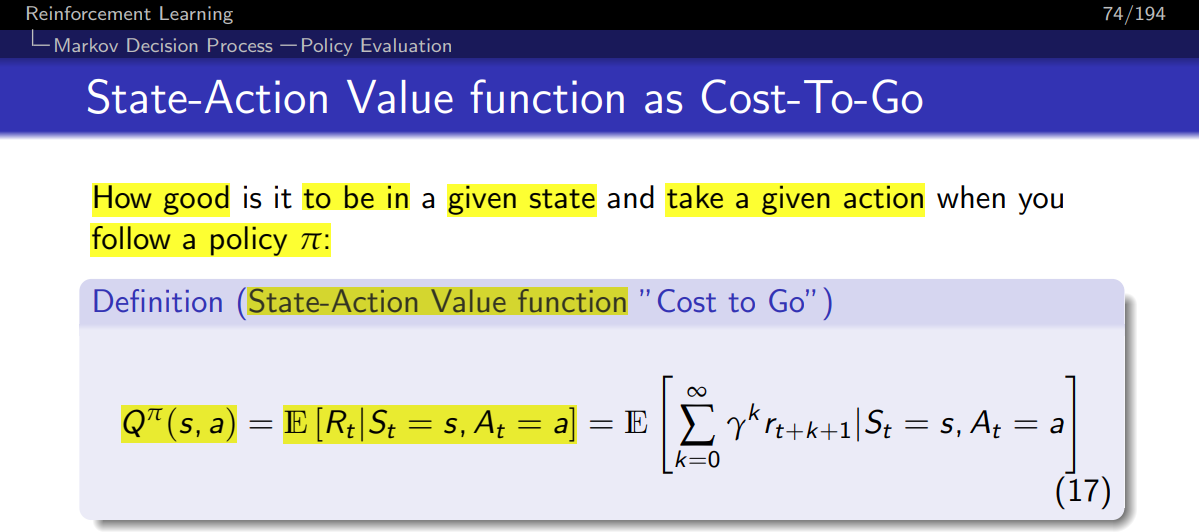
Soft (E) => All epsilon-greedy policies are soft (i.e. probabilistic)

1.b)

Monte carlo (E)

Action-value (B)





1.c)

Every visit (E) - I don t think it s every visit because every visit averages the returns at the end of the episode

Sampling (B)

1.d)

Model free (A)

Policy iteration (F) meim

.e)

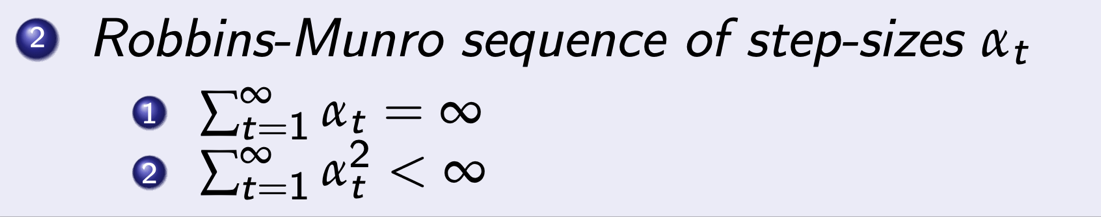
State – action pairs visited infinity times (B)

Policies converge to optimal greedy policy (C)

Sum alpha-I -> infinity (I)

E F (Robbins-Munro condition 2 implies F implies E)? (+1 vote)

I don’t think E is correct, sum of alpha goes to infinity doesn’t imply alpha goes to 0 (+1 vote)



1.f) 6 (F)-;

The feedback seems to indicate that the wording is ambiguous, but only 5 outputs may be required: 3 angular speeds + 1 linear speed + 1 choice of axis

1.g) or Parameters ψ experience a smaller range (B)?

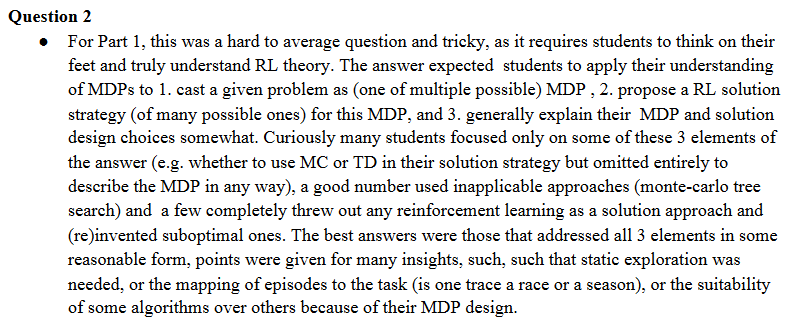
1.h) 10 (F)

1.i) H (Not examined anymore 2021+)F?

1.j) 1.5 (G) (not examined)

anymore (I don’t think)

2a.



1. Cast into MDP

Idea for state space: Need some way to track the progress through a season, so use the number of wins each horse has accumulated. Each race advances the state space by incrementing the horse that won (according to p\_i). Terminal states: total number of wins across

horses = R.

Action space: H (Pick a horse for each race)



Returns: Either H – 1 or –1.

Guys I have an alternative casting, any comment would be appreciated:

My idea is each race is an episode, I think when the horses are replaced after a season, our 'environment' is changed, and we should learn again.

MDP:

S: start, win, lose

A: bet on one horse H\_i

P: the probability of winning or losing by taking action/choose a horse, (which we don't know)

R: r(start, a=bet on horse, lose) = -1, r(start, a=bet on horse, win) = H-1

Pi: policy is just probability of betting on a horse, or a deterministic betting on one horse.

Gamma: This doesn’t matter since only one step will be taken.

1. Propose RL solution

Let’s go with Q-Learning assuming the state space is tractable (if not, DQN). Each season is a trace/episode since the horses are reset. Pick horses according to the epsilon greedy action and observe the outcome of the race. Sum the immediate reward (H-1 or –1) and the undiscounted future return (using the greedy action), and set Q.

1. “Generally explain somewhat”

We don’t know the probabilities, hence have to go with Model-Free Control. Between Monte-Carlo and TD, TD is preferred as it can exploit the Markov property that each state is independent of the past and can improve rewards online during a season. “Static exploration” => does this just mean epsilon-soft? to explore the large state space.

We want the optimal policy, so it would be good to go with off-policy methods rather than on-policy methods that derive the best exploratory policy.

2b.

Not examinable