
```
% Achyuth Nandikotkur
% V00975928
% ECE-559B
% October 30, 2021

% Question 1

clear;
clc;

global returns qvector searchrewards waitrewards startState
actionsAtHigh actionsAtLow;

% To store average returns
% high - low
meanhighvalue = [0];
meanlowvalue = [0];

searchrewards = [3, 4, 5, 6];
waitrewards = [0, 1, 2];

prob.high.search = 1/2;
prob.high.wait = 1/2;
prob.low.search = 1/2;
prob.low.wait = 1/4;
prob.low.recharge = 1/4;

loop = 1;
counterhigh = 0;
counterlow = 0;
% returns{0} high
% returns{1} low
returns = [0 0];
while(loop < 2000)
    %since a four step episodic task,
    % example: high -> search -> high -> search ->
    %low -> wait -> low -> recharge -> high

    G = 0;
    sequence = generateEpisode(prob);

    for i = (length(sequence)-1): -1: 1
        G = G + sequence{i}.reward;

        skip = 0;
        for k = i-1: -1: 1
            if((sequence{i}.state == sequence{k}.state))
                skip = 1;
                break;
            end
        end
    end

    loop = loop + 1;
end
```

```

        % if the pair isn't seen in the values
        if(~skip)
            if(sequence{i}.state == 1)
                counterhigh = counterhigh + 1;
                returns = [returns; [G 0]];
                meanhighvalue = [meanhighvalue; sum(returns(:, 1))/
counterhigh];
                meanlowvalue = [meanlowvalue; meanlowvalue(end)];
            else
                counterlow = counterlow + 1;
                returns = [returns; [0 G]];
                meanlowvalue = [meanlowvalue; sum(returns(:, 2))/
counterlow];
                meanhighvalue = [meanhighvalue; meanhighvalue(end)];
            end
        end
    end

    loop = loop + 1;
end

celldisp(sequence);

t=1:2000;
plot(1:length(meanhighvalue), meanhighvalue, 1:length(meanlowvalue),
    meanlowvalue)
xlabel('Episodes')
ylabel('State values')
legend({'high', 'low'}, 'Location', 'southwest')

function resp = generateEpisode(probability)
    global waitrewards searchrewards;
    sequence = cell(1, 4);

    sequence{1}.state = 0;
    sequence{1}.action = 0;
    sequence{1}.reward = 0;

    sequence{2}.state = 0;
    sequence{2}.action = 0;
    sequence{2}.reward = 0;

    sequence{3}.state = 0;
    sequence{3}.action = 0;
    sequence{3}.reward = 0;

    sequence{4}.state = 0;
    sequence{4}.action = 0;
    sequence{4}.reward = 0;

    sequence{5}.state = 0;
    sequence{5}.action = 0;
    sequence{5}.reward = 0;

```

```

    % selecting initial state as high = 1 or low = 2 with equal
probability
    initialstate = randsample([1, 2], 1, true, [0.5, 0.5]);
    sequence{1}.state = initialstate;

    for i=1:4
        % check if state is high or low
        if(sequence{i}.state == 1)
            % action can be search = 1, wait = 2;
            action = randsample([1, 2], 1, true,
[probability.high.search, probability.high.wait]);
            if action == 1
                nextstate = randsample([1, 2], 1, true, [0.25, 0.75]);
                reward = randsample(searchrewards,1);
            else
                reward = randsample(waitrewards,1);
                nextstate = 1;
            end
        else
            % state is low

            % generate action with input probabilities
            action = randsample([1, 2, 3], 1, true,
[probability.low.search, probability.low.wait,
probability.low.recharge]);

            if action == 1
                nextstate = randsample([2, 1], 1, true, [0.25, 0.75]);
                if(nextstate == 2)
                    reward = randsample(searchrewards,1);
                else
                    reward = -3;
                end
            elseif(action == 2)
                reward = randsample(waitrewards,1);
                nextstate = 2;
            else
                reward = 0;
                nextstate = 1;
            end
        end
        sequence{i}.action = action;
        sequence{i}.reward = reward;
        sequence{i+1}.state = nextstate;
    end
    resp = sequence;
end

sequence{1} =

    state: 1
    action: 2

```

reward: 0

sequence{2} =

state: 1
action: 1
reward: 4

sequence{3} =

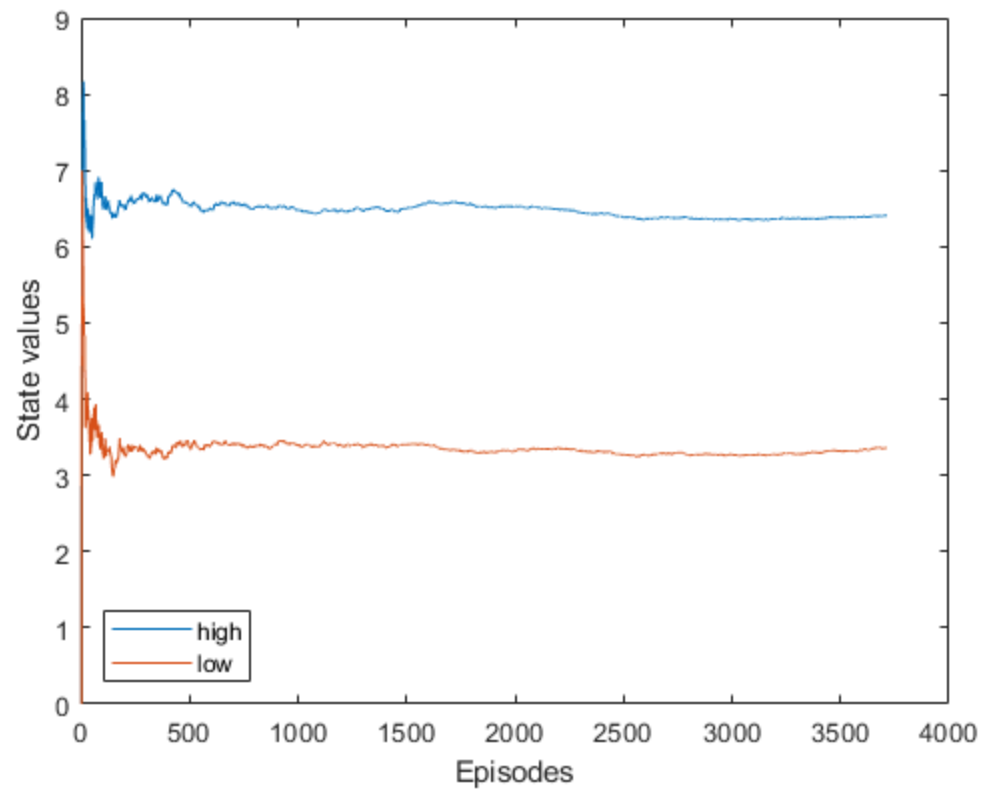
state: 2
action: 1
reward: 5

sequence{4} =

state: 2
action: 1
reward: -3

sequence{5} =

state: 1
action: 0
reward: 0



Published with MATLAB® R2021a