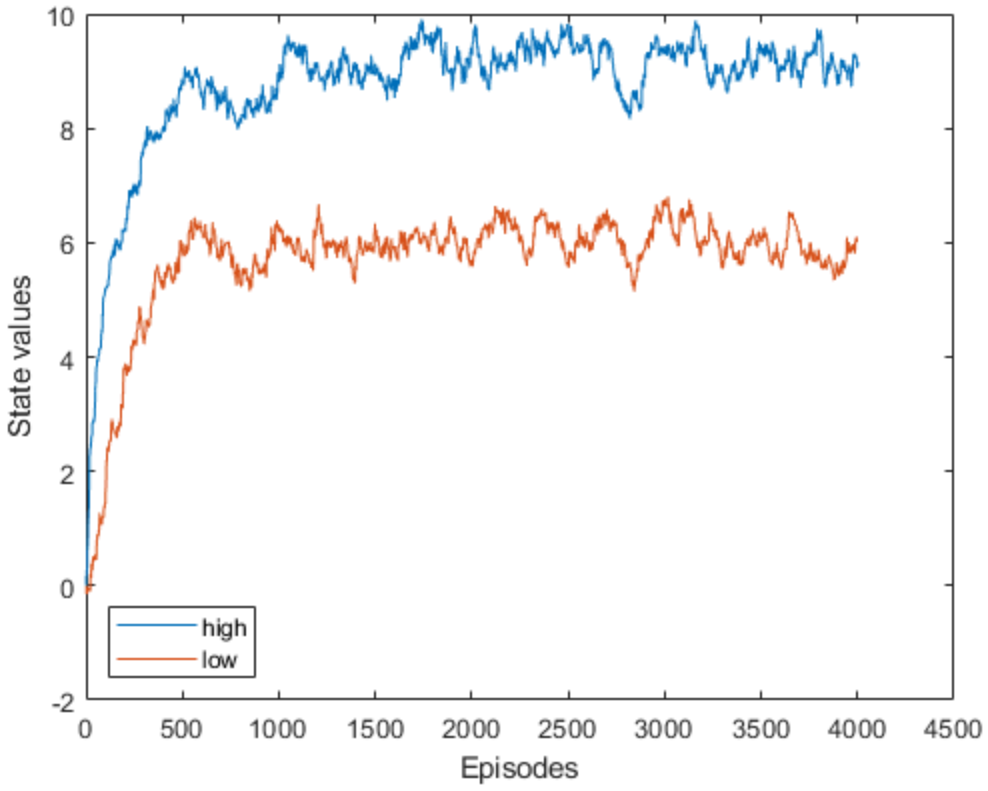
```
% Achyuth Nandikotkur
% V00975928
% ECE-559B
% October 30, 2021

% Question 3

clear;
clc;

global returns qvector searchrewards waitrewards startState
 actionsAtHigh actionsAtLow stepsize;

% To store average returns
% high - low
valuestatehigh = [0];
valuestatelow = [0];

stepsize = 0.05;

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 < 2)
    %since a four step episodic task,
    % example: high -> search -> high -> search ->
    %low -> wait -> low -> recharge -> high

    G = 0;
    % sequence = generateEpisode(prob);
    % selecting initial state as high = 1 or low = 2 with equal
 probability
    sequence = cell(1, 4000);
    for k1 = 1:4000
        sequence{k1}.state = 0;
        sequence{k1}.action = 0;
        sequence{k1}.reward = 0;
    end

    initialstate = randsample([1, 2], 1, true, [0.5, 0.5]);
```

```
   sequence{1}.state = initialstate;

   for i = 1: 4000
       % 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, [prob.high.search,
prob.high.wait]);
           if action == 1
             sequence{i+1}.state = randsample([1, 2], 1, true, [0.25,
0.75]);
             reward =  randsample(searchrewards,1);
           else
             reward =  randsample(waitrewards,1);
             sequence{i+1}.state = 1;
           end

           if(sequence{i+1}.state == 1)
               temphigh = valuestatehigh(end) + stepsize * (reward +
(0.8 * (valuestatehigh(end))) - valuestatehigh(end));
           else
               temphigh = valuestatehigh(end) + stepsize * (reward +
(0.8 * (valuestatelow(end))) - valuestatehigh(end));
           end

           valuestatehigh = [valuestatehigh; temphigh];
           valuestatelow = [valuestatelow; valuestatelow(end)];
       else
           % state is low

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

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

           if(sequence{i+1}.state == 1)
               templow = valuestatelow(end) + stepsize * (reward +
(0.8 * (valuestatehigh(end))) - valuestatelow(end));
           else
```

```
                templow = valuestatelow(end) + stepsize * (reward +
 (0.8 * (valuestatelow(end))) - valuestatelow(end));
            end

            valuestatelow = [valuestatelow; templow];
            valuestatehigh = [valuestatehigh; valuestatehigh(end)];
        end
    end

    loop = loop + 1;
end

% celldisp(sequence);

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


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

    for k1 = 1:4000
        sequence{k1}.state = 0;
        sequence{k1}.action = 0;
        sequence{k1}.reward = 0;
    end


    % 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=1:4000
        % 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
```



Published with MATLAB® R2021a