Question 1: Understanding of MDPs

1. My CID is 00598177, so as specified the modified CID number to be used is 1598177. Computing $s(t) = (CID(t) + 2)\ mod\ 3$, $r(t) = CID(t)\ mod\ 3$ yields the following trace:

$$\tau = s_0\ 1\ s_1\ 1\ s_2\ 1\ s_1\ 0\ s_0\ 1\ s_0\ 1\ s_0\ 1$$

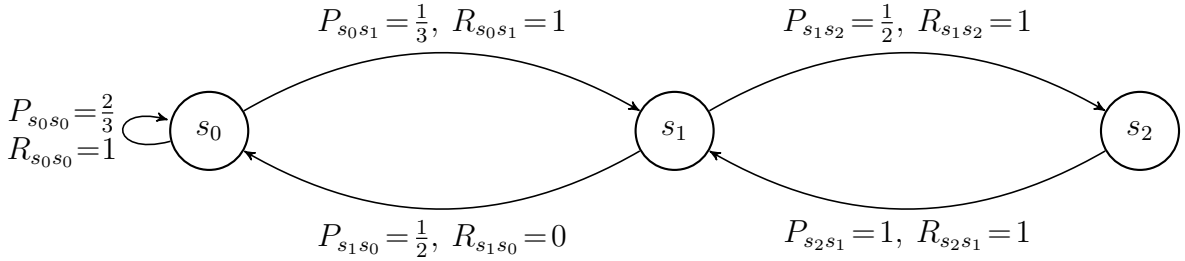
2.a. Given the data that we just observe (trace τ), and under the assumption that the Markov chain is stationary, we can estimate the transition matrix $\mathbf{P}_{ss'}$ by computing the relative frequency of transition from s to s' within our data:

$$\hat{P}(s_{t+1} = s'|s_t = s) = \frac{\text{number of transitions from s to s'}}{\text{total number of transitions out of s}} \Rightarrow \hat{\mathbf{P}}_{ss'} = \begin{bmatrix} 2/3 & 1/3 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \end{bmatrix}$$

Similarly, we can estimate the reward of a transition from s to s', $\hat{\mathbf{R}}_{ss'}$, as well as the expected reward collected upon leaving state s, $\hat{\mathbf{R}}_s$, by using our data (note that the reward for transitions not observed within our data is denoted by *):

$$\hat{\mathbf{R}}_{ss'} = \begin{bmatrix} 1 & 1 & * \\ 0 & * & 1 \\ * & 1 & * \end{bmatrix} \Rightarrow \hat{\mathbf{R}}_s = \begin{bmatrix} 2/3 + 1/3 + 0 \\ 0 + 0 + 1/2 \\ 0 + 1 + 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ 1 \end{bmatrix}$$

The minimal MDP graph consistent with the data is shown below. The state space is $\mathcal{S} = \{s_0,\ s_1,\ s_2\}$ and the transition probabilities and rewards are as specified above. Note that the reward for transitioning out of state s_1 is stochastic (0 with 50% probability and 1 with 50% probability).



2.b. (a) If we consider the above Markov Reward Process as fully described by $\hat{\mathbf{P}}_{ss'}$ and $\hat{\mathbf{R}}_s$, then we could calculate the value of each state using the Bellman equation. However, the MRP does not have a terminating state and the process will loop forever with positive rewards accumulated along the loop. Hence, the value of all states (including s_0) will increase as a function of γ and reach infinity at $\gamma = 1$. This is illustrated by considering that by the Bellman equation: $\mathbf{v} = (\mathcal{I} - \gamma\mathcal{P})^{-1}\mathcal{R}$. But if we compute $\lim_{\gamma \to 1}(\mathcal{I} - \gamma\hat{\mathbf{P}}_{ss'})^{-1}\hat{\mathbf{R}}_s$ we find that it tends to infinity.

(b) An alternative approach is to consider the trace as a complete episode of an unknown MDP that has terminal states (which is particularly dubious as the final state visited, s_0, is also visited within the trace and is hence clearly not a terminating state). Then we can use a Monte Carlo algorithm to estimate the value of states. We choose the Every-visit MC algorithm and estimate the value of $s_0 = \frac{5+2+1}{4} = 2$.

(c) Regardless of whether the trace is a complete episode, we can use the TD algorithm. The values of all states are initialized to 0 and then are successively updated as we move along the trace. The update rule used is: $\hat{V}(s_t) \leftarrow \hat{V}(s_t) + \alpha \left(r_{t+1} + \gamma \hat{V}(s_{t+1}) - \hat{V}(s_t) \right)$. For $\gamma = 1$, $\alpha = 1$ (maximum learning rate chosen since the sample size is so small) the TD method estimates the value of s_0 as 3.

Question 2: Understanding of Grid Worlds

1. $[x, y, z] = [1, 7, 7] \Rightarrow$ reward state $= s_3$, $p = 0.3$, $\gamma = 0.55$

2. The optimal value function and optimal policy were computed using the value iteration algorithm where the estimated optimal value function is obtained by incremental improvement until it converges to the optimal value function. The optimal policy is then simply the policy that maximizes the value of each state, given the optimal value function calculated. This approach has the benefit of no explicit policy until the very last step. Convergence was assumed after a threshold of $\theta = 0.0001$ was crossed. Given the set of actions, $\mathcal{A} = \{N, E, S, W\}$, with stochastic effects, four transition matrices needed to be calculated.

14 iterations were needed for convergence of the algorithm. It was noted that increasing θ to 0.001 yielded the same values to 3 decimal places but converged after 11 iterations. Increasing θ to 0.01 yielded similar results but did not require less iterations to converge (still 11), thus providing no benefit. Figures 1 and 2 show the optimal value function and optimal policy respectively.

Figure 1: Optimal value function. Values for each state rounded to 2 decimal places.

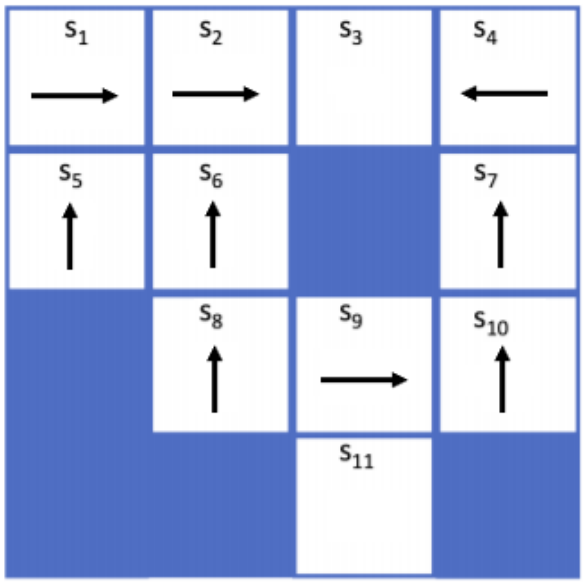


Figure 2: Optimal policy. Arrows indicate optimal action direction for each state (deterministic policy), multiple arrows from one state indicate equiprobable choice between indicated directions (stochastic policy).

3. The optimal policy (π^*) executes action 'East' at state s_9. The optimal policy is deterministic, i.e $P(\text{a=East}|\pi^*, s_9) = 1$. Given that the action is East, the transition probability wiil be:

$$P(s'|East, s_9) = \begin{cases} p & s' = s_{10} \\ \frac{1-p}{3} & s' \in \{s_8,\ s_9,\ s_{11}\} \end{cases}$$

$p > 0.25 \Rightarrow p > \frac{1-p}{3}$ which means that the action is more likely to succeed than fail. As such, the optimal actions correspond to the optimal outcome.

The optimal outcome when the agent is at s_9 is clearly not s_{11} (corresponding to success of action 'South') as that would incur a penalty of -100. Additionally, for $\gamma > 0$ (the agent cares about more than just the immediate reward) staying in s_9 (corresponding to success of action 'North') is not optimal as the agent has not improved its position and has incurred a transition cost of -1. Observing that $V(s_{10}) > V(s_8)$ we can see why transitioning into s_{10} is optimal. However, determining intuitively the optimal choice between s_8 (success in going 'West') and s_{10} (success in going 'East') is not obvious. That is because both states are equidistant from the penalty state and from the reward state. This is captured by the fact that their values are very close.

If $p = 0.25$, then the action chosen by the agent does not affect the outcome (all directions equiprobable with 25% chance) and the optimal policy will consist of equiprobable choice between all directions at all states. If $p < 0.25$ then the probability of going in the chosen direction is less than that of going in any other direction and hence 'East' would be a sub-optimal choice at s_9. In fact, for $p < 0.25$ it is expected that the optimal choice would be 'South' (thus minimizing the likelihood of transitioning into s_{11}).

If $\gamma = 0$, then the optimal action would be the equiprobable choice between North, East and West all with immediate reward -1. Finally, for $\gamma = 1$ and $p = 1$ the optimal action would be equiprobable choice between East and West with values of both s_8 and s_{10} equal to 8.

Note that my personalized value of $\gamma = 0.55$ implies immediate rewards are weighted as approximate twice as important as rewards after just one step and approximately 10 times more important than rewards after just 4 steps ($0.55^4 = 0.09$).

4. The optimal value function that I obtained shows positive values only for states adjacent to the reward state s_3. State 4 has higher value than state 2. This can be explained by the fact that the probabilities of reaching s_3 after exactly one transition is equal for s_2 and s_4:

$$P(s_{t+1} = s_3|s_t = s_2, \pi^*) = P(s_{t+1} = s_3|s_t = s_4, \pi^*) = p$$

while the probabilities of reaching s_3 after exactly two transitions are not (let $q = \frac{1-p}{3}$):

$$\begin{array}{ll} P(s_{t+2} = s_3|s_t = s_2, \pi^*) = & qp \\ P(s_{t+2} = s_4|s_t = s_2, \pi^*) = & 2qp \end{array} \Rightarrow P(s_{t+2} = s_4|s_t = s_2, \pi^*) > P(s_{t+2} = s_4|s_t = s_2, \pi^*)$$

In other words, there is a larger chance of collecting the reward within two steps if we start at s_4 than if we start at s_2. This gives us insight into why the value of s_7 (adjacent to s_4) is higher than that of s_6 (adjacent to s_2) even though they are equidistant to both the reward and penalty state. With this reasoning in mind we can comprehend why the optimal action at state s_9 is to go 'East' along the $s_{10} \rightarrow s_7 \rightarrow s_4 \rightarrow s_3$ path.

At $p = 0.25$ the agent cannot effect its direction by its choice, whereas for $p > 0.25$ it chooses its most likely direction and for $p < 0.25$ the direction can be chosen implicitly (the agent can choose to minimize the odds of going in a particular direction). Hence it stands to reason that the values of all states will be at a minimum for $p = 0.25$. As the agent obtains more and more ability to choose its direction (as p increases) the values will increase, reaching a maximum at $p = 1$.

When $p = 1$ and $\gamma = 1$ the values will correspond to 10 minus the shortest path between the state and the reward: $V(s_1) = 9$, $V(s_2) = 10$ ... $V(s_9) = 7$, $V(s_{10}) = 8$.

For $\gamma = 0$ the immediate rewards is the only thing that counts. Thus all states will have value of -1 except for those adjacent to the reward and penalty states. Then, for $p > 0.25$, $\gamma = 0$: $V(s) = -1$ for $s \notin \{s_2, s_3, s_4, s_9, s_{11}\}$, $V(s_2) = V(s_4) = -3q+10p$, $V(s_9) = -p-2q-100q$.

The optimal policy calculated is deterministic with all optimal actions following the intuitive rule of taking the agent away from the penalty state and towards the reward along the shortest path. The only optimal action that is not obviously intuitive occurs at state s_9 as discussed at length. However, by examining the values of s_2 vs s_4 and s_6 vs s_7 a reasonable explanation has been established. The optimal policy becomes stochastic when for all values of p when $\gamma \in \{0, 1\}$. It also becomes stochastic for all values of γ when $p \in \{[0, 0.25] \cup \{1\}\}$. The optimal policy does not change for $0.2 \leq \gamma \leq 0.65$ and $0.25 \leq p \leq 0.7$ (the values possible for students doing the coursework). But at $\gamma = 0.875$ and $p = 0.3$ we observe that the optimal action at s_9 changes to 'West'. It is interesting to note that for $p > 0.3$ that change does not occur. Nor does it occur for $\gamma < 0.85$ or $\gamma = 1$.

Appendix: MATLAB code

```
clc;close all;clear      %Question 1
g=1;                     %gamma=1
CID=[1 5 9 8 1 7 7];     % modified CID
S=mod((CID+2),3)+1;      %states in trace (note that 1->s0, 2->s1, 3->
    s2
r=mod(CID,2);            %rewards in trace
for ind=1:length(S)      %calc trace in format specified
    if S(ind)==1 tau(2*ind-1)="s0"; end
    if S(ind)==2 tau(2*ind-1)="s1"; end
    if S(ind)==3 tau(2*ind-1)="s2"; end
    tau(2*ind)=r(ind);
end
tau                      %display trace
P=zeros(3);              %initialize transition matrix;
R=NaN(3);                %initialize rewards matrix
%calculate trnasition matrix and reward matrix
for s=1:3
    for ind=1:(length(S)-1)
        if S(ind)==s
            P(s,S(ind+1))=P(s,S(ind+1))+1;
            R(s,S(ind+1))=r(ind);
        end
    end
    P(s,:)=P(s,:)/sum(P(s,:));
end
P                        %display transition matrix
R                        %display reward matrix
Rs=nansum(R.*P,2)        %calc and display reward vector
%every visit monte carlo
v0=0;
n_visits=0;
for ind=1:(length(S)-1)
    if S(ind)==1
        v0=v0+sum(r(ind:(length(S)-1)));
        n_visits=n_visits+1;
    end
end
v0=v0/(n_visits+(S(end)==1))    %result of monte carlo
%TD
V=[0 0 0];
a=1;
for ind=1:(length(S)-1)
    V(S(ind))=V(S(ind))+a*(r(ind+1)+g*V(S(ind+1))-V(S(ind)));
end
V                                %result of TD
```

```
45
46 %%
47 clc;clear;close all;                %Question 2
48 CID=[1 5 9 8 1 7 7];
49 win_st=mod(CID(end)+1,3)+1          %reward state
50 g=0.2+0.5*CID(end-1)/10             %personalized gamma
51 p=0.25+0.5*CID(end-2)/10            %personalized p
52
53 %here we will make the 4 transition matrices initialize prob matrices
54 P1=zeros(11,11); P2=P1;P3=P1; P4=P1;
55
56 %fill out any non zero probability events with 1s. We will put the
       weight in soon.
57 P1_ind=[1 2 3 4 1 2 4 6 9 7 11];
58 P2_ind=[2 3 4 4 6 6 7 9 10 10 11];
59 P3_ind=[5 6 3 7 5 8 10 8 11 10 11];
60 P4_ind=[1 1 2 3 5 5 7 8 8 9 11];
61
62 P1_ind(win_st)=win_st;
63 P2_ind(win_st)=win_st;
64 P3_ind(win_st)=win_st;
65 P4_ind(win_st)=win_st;
66
67 P1(11*(P1_ind-1)+(1:11))=1;
68 P2(11*(P2_ind-1)+(1:11))=1;
69 P3(11*(P3_ind-1)+(1:11))=1;
70 P4(11*(P4_ind-1)+(1:11))=1;
71
72 %weighted sum of the above matrices is the transition matrix for
       different actions (1->choose N, 2->choose E etc)
73 Pn=p*P1+((1-p)/3)*(P2+P3+P4);
74 Pe=p*P2+((1-p)/3)*(P1+P3+P4);
75 Ps=p*P3+((1-p)/3)*(P1+P2+P4);
76 Pw=p*P4+((1-p)/3)*(P1+P2+P3);
77
78 %there might be a more intuitive way to make the transition matrices,
       but this works so i'll leave it as is
79
80 %lets make the reward matri.Note that the reward for imposible
       transitions is irrelevant. However we set it to -1 here (any vale
       would be fine).
81 R=-ones(11,11);
82 R(:,11)=-100;%set the penalty for reaching state 11
83 R(:,win_st)=10;%set the reward for reaching reward state
84 R(win_st,:)=0;%set the absorbing state rewards to 0;
85 R(11,:)=0;
86
```

```
87 %value iteration to find optimal value function and optimal policy
       initialize value
88 V=zeros(1,11);
89
90 thr=0.0001;%number of iterations is not very sensitive to thr
91 d=thr+0.01;% make sure d starts larger than thr
92 ind=0;       % index to denote number of iteration
93 while(d>thr)
94     d=0;
95     ind=ind+1
96     for s=1:11
97         temp=V(s);
98         V(s)=max([Pn(s,:);Pe(s,:);Ps(s,:);Pw(s,:)]*(R(s,:)'+g*V'));
99         d=max(d,abs(temp-V(s)));
100    end
101 end
102 round(V,2) %display rounded value function
103
104 %to find the optimal policy we find the policy that maximizes the
       state values given the optimal value function calculated above
105 pol=["N" "E" "S" "W"];
106 for s=1:11
107     if s~=win_st && s~=11
108         pol_s=zeros(1,4);
109         temp=[Pn(s,:);Pe(s,:);Ps(s,:);Pw(s,:)]*(R(s,:)'+g*V');
110         m=max(temp);
111         pol_s(find(temp==m))=1;
112         pol=[pol;(pol_s/nansum(pol_s))];
113     else
114         pol=[pol;["-" "-" "-" "-"]];
115     end
116 end
117 [[" " "s1","s2","s3","s4","s5","s6","s7","s8","s9","s10","s11"]' pol]
```