CSE 250A HW9

Jiping Lin A15058075

1. We have $\log\gamma \le \gamma-1 \Rightarrow e^{\log\gamma} \le e^{\gamma-1}$
$\Rightarrow \gamma \le e^{\gamma-1}$

then $\sum_{n\ge t}\gamma^n r_n \le \sum_{n\ge t}\gamma^n$ ⌈Since $0\le r_n \le 1, \forall n$⌋

$= \frac{\gamma^t}{1-\gamma}$ ⌈Geo. Series⌋

$\le \frac{e^{t(\gamma-1)}}{1-\gamma}$

$= he^{t(\gamma-1)}$ ⌈$h=\frac{1}{1-\gamma}$⌋

$= he^{-t/h}$

BE: $V^\pi(s) = R(s) + \gamma\sum_{s'} P(s'|s,\pi(s)) \cdot V^\pi(s')$

2.(a) $V^\pi(1) = R(1) + \frac{2}{3}[P(1|1,\uparrow)V^\pi(1) + P(2|1,\uparrow)V^\pi(2) + P(3|1,\uparrow)V^\pi(3)]$

$= -15 + \frac{2}{3}[\frac{3}{4}V^\pi(1) + \frac{1}{4}V^\pi(2)]$

$V^\pi(2) = R(2) + \frac{2}{3}[P(1|2,\uparrow)V^\pi(1) + P(2|2,\uparrow)V^\pi(2) + P(3|2,\uparrow)V^\pi(3)]$

$= 30 + \frac{2}{3}[\frac{1}{2}V^\pi(1) + \frac{1}{2}V^\pi(2)]$

$V^\pi(3) = R(3) + \frac{2}{3}[P(1|3,\downarrow)V^\pi(1) + P(2|3,\downarrow)V^\pi(2) + P(3|3,\downarrow)V^\pi(3)]$

$= -25 + \frac{2}{3}[\frac{1}{4}V^\pi(2) + \frac{3}{4}V^\pi(3)]$

First 2 eq. gives

$V^\pi(1) = -15 + \frac{1}{2}V^\pi(1) + \frac{1}{6}V^\pi(2)$

$V^\pi(2) = 30 + \frac{1}{3}V^\pi(1) + \frac{1}{3}V^\pi(2)$

$\Rightarrow V^\pi(1) = -18,\ V^\pi(2) = 36$. $\Rightarrow V^\pi(3) = -25 + \frac{1}{6}V^\pi(2) + \frac{1}{2}V^\pi(3)$

$\Rightarrow V^\pi(3) = -19$

-18
36
-19

(b) $Q^\pi(s,a)$ are terms in [ ] in pt. (a),

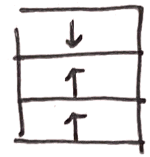
$\pi'(1) = \underset{a}{\operatorname{argmax}}\{P(1|1,\uparrow)V^\pi(1) + P(2|1,\uparrow)V^\pi(2) + P(3|1,\uparrow)V^\pi(3),\ P(1|1,\downarrow)V^\pi(1) + P(2|1,\downarrow)V^\pi(2) + P(3|1,\downarrow V^\pi(3)\}$

$= \underset{a}{\operatorname{argmax}}\{\frac{3}{4}V^\pi(1) + \frac{1}{4}V^\pi(2),\ \frac{1}{4}V^\pi(1) + \frac{3}{4}V^\pi(2)\}$

$= \downarrow$ ⌈Since second term is larger (has more weight on $V^\pi(2)$⌋

Similarly

$$\pi'(2) = \underset{a}{\operatorname{argmax}} \left\{ \tfrac{1}{2}V^{\pi}(1) + \tfrac{1}{2}V^{\pi}(2),\ \tfrac{1}{2}V^{\pi}(2) + \tfrac{1}{2}V^{\pi}(3) \right\}$$



$$= \uparrow \quad \ulcorner \text{Since } V^{\pi}(1) > V^{\pi}(3) \lrcorner$$

$$\pi'(3) = \underset{a}{\operatorname{argmax}} \left\{ \tfrac{3}{4}V^{\pi}(2) + \tfrac{1}{4}V^{\pi}(3),\ \tfrac{1}{4}V^{\pi}(2) + \tfrac{3}{4}V^{\pi}(3) \right\}$$

$= \uparrow$ 「First term has "more" $V^{\pi}(2)$, which is larger」

3.(a) $V^{\pi}(s) = R(s) + \gamma \sum_{s'} P(s'|s,a) V^{\pi}(s')$ 「BE」

$$= R(s) + \gamma\left(P(s|s,a)V^{\pi}(s) + P(s+1|s,a)V^{\pi}(s+1)\right)$$

$$= R(s) + \gamma\left(\tfrac{2}{3}V^{\pi}(s) + \tfrac{1}{3}V^{\pi}(s+1)\right)$$

$$= R(s) + \tfrac{2}{3}\gamma V^{\pi}(s) + \tfrac{1}{3}\gamma V^{\pi}(s+1)$$

$$\Rightarrow V^{\pi}(s) = \frac{s}{1-\frac{2}{3}\gamma} + \frac{\frac{1}{3}\gamma}{1-\frac{2}{3}\gamma} V^{\pi}(s+1)$$

(b) Write $\beta = \dfrac{s}{1-\frac{2}{3}\gamma} = \dfrac{s}{\frac{3-2\gamma}{3}} = \dfrac{3s}{3-2\gamma}$

$\lambda = \dfrac{\frac{1}{3}\gamma}{1-\frac{2}{3}\gamma} = \dfrac{\gamma}{3-2\gamma}$, so $V^{\pi}(s) = \beta + \lambda V^{\pi}(s+1)$

If $V^{\pi}(s) = as+b \quad \forall s \in \{0,1,\cdots\}$,

then $V^{\pi}(s+1) = a(s+1)+b$,

and $V^{\pi}(s) = \beta + \lambda V^{\pi}(s+1)$

$$= \beta + \lambda[a(s+1)+b]$$

$$= \beta + \lambda a(s+1) + \lambda b$$

$$\Rightarrow as+b = \frac{3s}{3-2\gamma} + \frac{\gamma a(s+1)}{3-2\gamma} + \frac{b\gamma}{3-2\gamma}$$

$$\Rightarrow (as+b)(3-2\gamma) = 3s + \gamma a(s+1) + b\gamma, \ \forall s$$

Since this is true for all s, then it is true for $s=1,2$,

so $\begin{cases} (a+b)(3-2\gamma) = 3 + 2\gamma a + b\gamma \\ (2a+b)(3-2\gamma) = 6 + 3\gamma a + b\gamma \end{cases}$

solve for a, b we have

$$\begin{cases} a = \dfrac{1}{1-\gamma} \\ b = \dfrac{\gamma}{3(\gamma-1)^2} \end{cases}$$

4. See code

5. $\Delta_k = \max_s |V_k(s) - V^\pi(s)|$

$$= \max_s |(\underset{\sim}{R(s)} + \gamma \sum_{s'} P(s'|s,\pi(s)) V_{k-1}(s'))$$

$$- (\underset{\sim}{R(s)} + \gamma \sum_{s'} P(s'|s,\pi(s)) V^\pi(s'))|$$

$$= \gamma \max_s |\sum_{s'} P(s'|s,\pi(s)) V_{k-1}(s') - \sum_{s'} P(s'|s,\pi(s)) V^\pi(s')|$$

$$= \gamma \max_s |\sum_{s'} [P(s'|s,\pi(s)) (V_{k-1}(s') - V^\pi(s'))]|$$

$$= \gamma \max_{s,s'} |\sum_{s'} (V_{k-1}(s') - V^\pi(s'))|$$ 「Choose the largest weight」

$$= \gamma \Delta_{k-1}$$

Since $\gamma < 1$, we have $\Delta_k < \gamma \Delta_{k-1}$.

So $k \to \infty \Rightarrow \Delta_k \to 0$, i.e. $\lim_{k\to\infty} V_k(s) = V^\pi(s)$

6. (a) $\sum_{k=1}^{\infty} \alpha_k = 1 + \frac{1}{2} + \frac{1}{3} + \cdots$,

this is the harmonic series $\sum_{n=1}^{\infty} \frac{1}{n^p}$.

and $\sum \frac{1}{n^p}$ converges if $p > 1$,

diverges if $p \le 1$

「This proof can be found in page 62 of principles of mathematical analysis by Rudin」

So $\sum_{k=1}^{\infty} \alpha_k = \infty$ diverges

$\sum_{k=1}^{\infty} \alpha_k^2 < \infty$ converges

(b) Base: $\mu_1 = \mu_0 + \alpha_1 (X_1 - \mu_0)$

$$= \alpha_1 X_1 = X_1$$

If $\mu_k = \frac{1}{k}(X_1 + \cdots + X_k)$

$$\mu_{k+1} = \mu_k + \alpha_{k+1}(X_{k+1} - \mu_k)$$

$$= \frac{1}{k}\sum_{i=1}^{k} X_i + \frac{X_{k+1} - \frac{1}{k}\sum_{i=1}^{k} X_i}{k+1}$$

$$= \frac{X_{k+1}}{k+1} + \left[\frac{1}{k} - \frac{1}{k(k+1)}\right] \sum_{i=1}^{k} X_i$$

$$= \frac{X_{k+1}}{k+1} + \frac{1}{k+1}\sum_{i=1}^{k} X_i = \frac{1}{k+1}\sum_{i=1}^{k+1} X_i$$

HW9 CODE

December 2, 2021

[1]:

```
import numpy as np
import random
# 9.4.a
a1 = np.loadtxt('prob_a1.txt')
a2 = np.loadtxt('prob_a2.txt')
a3 = np.loadtxt('prob_a3.txt')
a4 = np.loadtxt('prob_a4.txt')

gamma = 0.9925

def construct(matrix):
    S = 81
    res = np.zeros((S, S))
    for i in range(matrix.shape[0]):
        res[int(matrix[i][0] - 1)][int(matrix[i][1] - 1)] = matrix[i][2]
    return res

trans1 = construct(a1)
trans2 = construct(a2)
trans3 = construct(a3)
trans4 = construct(a4)

state = list(range(1, 5))

trans = {}
for i in range(4):
    trans[i + 1] = eval('trans' + str(i + 1))

reward = np.loadtxt('rewards.txt')


def p_matrix(pol):
    S = 81
    res = np.zeros((S, S))
    for i in range(res.shape[0]):
        d = pol[i]
        prob = trans[d]
        res[i] = prob[i]
```

```
    return res

def v_matrix(p):
    I = np.eye(81)
    return np.matmul(np.linalg.inv(I - gamma * p), reward)

def q_matrix(v, state, action):
    sum = 0
    prob = trans[action]
    for i in range(81):
        sum += prob[state][i] * v[i]
    return reward[state] + gamma * sum

def update(v):
    res = np.zeros(81)
    for i in range(81):
        choice = []
        for j in range(1, 5):
            choice.append(q_matrix(v, i, j))
        res[i] = np.argmax(np.array(choice)) + 1
    return res


policy = np.zeros(81)
# 1 left 2 up 3 right 4 down
for i in range(len(policy)):
    direction = random.randint(1, 4)
    policy[i] = direction
for i in range(100):
    prev = np.copy(policy)
    P = p_matrix(policy)
    V = v_matrix(P)
    policy = update(V)
    if np.allclose(prev, policy): break

dir_res = ['\u25A0'] * 81
for i in range(len(policy)):
    if V[i] == 0:
        dir_res[i] = '\u25A0'
        continue
    if policy[i] == 1:
        dir_res[i] = '\u2190'
    elif policy[i] == 2:
        dir_res[i] = '\u2191'
    elif policy[i] == 3:
        dir_res[i] = '\u2192'
    else:
```

```
        dir_res[i] = '\u2193'
dragon = [46, 48, 50, 64, 66, 68]
for i in dragon:
    dir_res[i] = '\u2573'
dir_res = np.array(dir_res)
dir_res1 = dir_res.reshape((9, 9)).T
print(dir_res1)
V1 = np.around(V, 2)
print(V1.reshape((9, 9)).T)

# 9.4.b

def update_v(prev):
    res = np.zeros(81)
    for s in range(len(prev)):
        choice = []
        for i in range(1, 5):
            sum = 0
            for j in range(len(prev)):
                prob = trans[i]
                sum += prob[s][j] * prev[j]
            choice.append(reward[s] + gamma * sum)
        maximum = np.max(np.array(choice))
        res[s] = maximum
    return res

V_value = np.zeros(81)
counter = 0
while True:
    prev = np.copy(V_value)
    V_value = update_v(V_value)
    if counter > 50 and 0.001 > prev[2] - V_value[2] > -0.001:
        break
    counter += 1
policy_value = update(V_value) # same as part a
V_value = np.around(V_value, 2)
print(V_value.reshape((9, 9)).T)
dir_res2 = ['\u25A0'] * 81
for i in range(len(policy_value)):
    if V[i] == 0:
        dir_res2[i] = '\u25A0'
        continue
    if policy_value[i] == 1:
        dir_res2[i] = '\u2190'
    elif policy_value[i] == 2:
        dir_res2[i] = '\u2191'
    elif policy_value[i] == 3:
```

```
        dir_res2[i] = '\u2192'
    else:
        dir_res2[i] = '\u2193'
dragon = [46, 48, 50, 64, 66, 68]
for i in dragon:
    dir_res2[i] = '\u2573'
dir_res2 = np.array(dir_res2)
dir_res3 = dir_res2.reshape((9, 9)).T
print(dir_res3)
```

```
[['' '' '' '' '' '' '' '' '']
 ['' '→' '→' '↓' '' '' '↓' '' '']
 ['→' '↑' '' '↓' '←' '←' '↓' '←' '']
 ['' '' '↓' '←' '' '' '↓' '' '']
 ['' '' '↓' '' '' '' '↓' '' '']
 ['' '↓' '←' '' '' '' '↓' '' '']
 ['' '↓' '' '→' '→' '→' '→' '→' '←']
 ['' '→' '→' '↑' '' '→' '→' '↑' '']
 ['' '' '' '' '' '' '' '' '']]
[[   0.      0.      0.      0.      0.      0.      0.      0.      0.  ]
 [   0.    102.38  103.23  104.1     0.   -133.33   81.4  -133.33    0.  ]
 [ 100.7   101.52    0.    104.98  103.78   90.99   93.67   81.4     0.  ]
 [   0.      0.    106.78  105.89    0.   -133.33   95.17 -133.33    0.  ]
 [   0.      0.    107.67    0.      0.      0.    108.34    0.      0.  ]
 [   0.    109.49  108.58    0.      0.   -133.33  109.58 -133.33    0.  ]
 [   0.    110.41    0.    114.16  115.12  116.09  123.64  125.25  133.33]
 [   0.    111.34  112.27  113.21    0.    122.02  123.18  124.21    0.  ]
 [   0.      0.      0.      0.      0.      0.      0.      0.      0.  ]]
[[   0.      0.      0.      0.      0.      0.      0.      0.      0.  ]
 [   0.    102.24  103.1   103.97    0.   -133.19   81.3  -133.19    0.  ]
 [ 100.57  101.39    0.    104.84  103.65   90.87   93.56   81.3     0.  ]
 [   0.      0.    106.65  105.76    0.   -133.19   95.06 -133.19    0.  ]
 [   0.      0.    107.54    0.      0.      0.    108.22    0.      0.  ]
 [   0.    109.36  108.45    0.      0.   -133.19  109.46 -133.19    0.  ]
 [   0.    110.28    0.    114.03  114.99  115.96  123.5   125.11  133.19]
 [   0.    111.2   112.14  113.08    0.    121.89  123.04  124.07    0.  ]
 [   0.      0.      0.      0.      0.      0.      0.      0.      0.  ]]
[['' '' '' '' '' '' '' '' '']
 ['' '→' '→' '↓' '' '' '↓' '' '']
 ['→' '↑' '' '↓' '←' '←' '↓' '←' '']
 ['' '' '↓' '←' '' '' '↓' '' '']
 ['' '' '↓' '' '' '' '↓' '' '']
 ['' '↓' '←' '' '' '' '↓' '' '']
 ['' '↓' '' '→' '→' '→' '→' '→' '←']
 ['' '→' '→' '↑' '' '→' '→' '↑' '']
 ['' '' '' '' '' '' '' '' '']]
```

```
[ ]:
```