MATH4406 – HW4

Madeleine Nargar, 41214538

Question 1

Consider a machine which has a disk which tends to slip out of alignment. Here $i \equiv$ degree of misalignment, with $i = 0 \equiv$ perfectly aligned. Every time the machine is used the disk will either remain at its current level of alignment, or worsen by a random (but usually small) amount. Assume that the alignment cannot improve spontaneously (though this needn't be true in other machine replacement scenarios). At each time step the machine can either be used as normal, or the disk can be re-aligned, incurring a cost of R and returning the machine to state 0. When the machine is misaligned it is less efficient, so has increased running costs.

Example parameters: $R = 100, C(i) = 10 + i, P_{i,j} = \begin{cases} 0 & j < i \\ 0.5 * 0.5^{j-i} & j \geq i \end{cases}$

Question 2

Stochastic ordering describes a scenario in which one random variable 'tends' to be larger than another. Considering the random variable $T_i :=$ state at time $t+1$ given state at time t is i, this is a sensible assumption for systems in which states evolve randomly, but with a tendency to decline (or a tendency to move to nearby states). For example, in the example given in Question 1, consider if at time t we are in state 0 ('perfect') or state 50 ('very bad'). Although it is possible to reach any larger state at time $t+1$, it is far more likely to that $i_{t+1} = 51$ if $i_t = 50$ than if $i_t = 0$ (as misalignment is random, but usually small, next state is likely to be nearby).

Question 3

Using the notation described for Part 1:

State space: $S = \{0, 1, 2, \dots\}$

Action set: $A = \{0, 1\}$, where $0 \equiv$ don't replace, $1 \equiv$ replace.

Transition probabilities: $p(j|s,a) = \begin{cases} P_{s,j} & a = 0 \\ 1 & a = 1, j = 0 \\ 0 & a = 1, j \neq 0 \end{cases}$

Rewards: $r(s,a) = \begin{cases} -C(s) & a = 0 \\ -C(s) - R & a = 1 \end{cases}$
Assuming operating cost is incurred first, regardless of action taken, then replacement may occur. Assumes $C(i) \geq 0$ for all i.

Objective: maximise expected total discounted reward ($v_\lambda^\pi(s),\ \forall s \in S$):
$v_\lambda^\pi(s) = \lim_{N\to\infty} E_s^\pi \sum_{t=1}^{N} \lambda^{t-1} r(s_t, a_t)$

Question 4

Letting $w(i)$ be the value-function (standard-form, i.e. maximisation criteria), and using rewards and transitions from Q3:

$$\begin{aligned} w(i) &= \sup_{a\in A}\left\{ r(s,a) + \sum_{j\in S} \lambda p(j|s,a) w(j) \right\} \\ &= \max\left\{ -C(i) + \sum_{j=0}^{\infty} \lambda P_{i,j} w(j), -C(i) - R + \lambda w(0) \right\} \\ &\quad \text{(replacing } sup \text{ with } max \text{ as finite action set)} \\ &= -C(i) + \max\left\{ \lambda \sum_{j=0}^{\infty} P_{i,j} w(j), -R + \lambda w(0) \right\} \\ &= -C(i) - \min\left\{ -\lambda \sum_{j=0}^{\infty} P_{i,j} w(j), R - \lambda w(0) \right\} \end{aligned}$$

$$max(X) = -min(-X)$$

Letting $v(i) = -w(i),$ i.e. value function wrt minimisation criteria

$$\begin{aligned} v(i) &= C(i) + \min\left\{ -\lambda \sum_{j=0}^{\infty} P_{i,j} w(j), R - \lambda w(0) \right\} \\ &= C(i) + \min\left\{ \lambda \sum_{j=0}^{\infty} P_{i,j} v(j), R + \lambda v(0) \right\} \end{aligned}$$

Question 5

Let $v^n(i)$ be value function given by n^{th} iteration of the value iteration algoritm.

Claim: $v^n(i)$ is an increasing function in i, for all $n \geq 0$.

Proof (by induction):
v^0 is arbitrary, so set $v^0(i) = C(i) \Rightarrow v^0(i)$ increasing.
($C(i)$ increasing as per problem definition).

Assume $v^k(i)$ is an increasing function in i.

$v^{k+1}(i) = C(i) + min\{R + \lambda v^k(0), \lambda \sum_{j=0}^{\infty} P_{i,j} v^k(j)\}$ (value-iteration)

$R + v^k(0)$ is a constant with respect to i.

$Q(i) := \lambda \sum_{j=0}^{\infty} P_{i,j} v^k(j) = \lambda \mathbb{E}[v^k(j)|i] = \lambda \mathbb{E}[v^k(T_i)]$

As $v^k(i)$ is an increasing function (by inductive assumption), we have $\mathbb{E}[v^k(T_i)]$ increasing (by stochastic ordering property, $\mathbb{E}[f(T_{i+1})] \geq \mathbb{E}[f(T_i)]$).

So for $\lambda \in (0,1)$ we have: $v^{k+1}(i) =$ increasing fn $C(i) + min\{$ constant , increasing fn $Q(i)\} \Rightarrow v^{k+1}(i)$ is increasing function.

$\Rightarrow v^n(i)$ increasing function for all $n \geq 0$.

$v^n(i) \rightarrow v(i) \Rightarrow v(i)$ is increasing function.

Question 6

Want to show $\exists\, i' \leq \infty : d^*(i) = \begin{cases} 0 \text{ (don't replace)} & i < i' \\ 1 \text{ (replace)} & i \geq i' \end{cases} \qquad (*)$

Case 1: $d^*(i) = 0\ \forall i, \Rightarrow (*)$ holds, $i' = \infty$

Case 2: For some $i, d^*(i) \neq 0$. Let $k = min\{i : d^*(i) = 1\}$

For state k, $\qquad R + \lambda v(0) \leq \lambda \sum_{j=0}^{\infty} P_{k,j} v(j) = \lambda \mathbb{E}[v(T_k)]$

As $v(i)$ is an increasing function (see Q5), by stochastic ordering property, $\mathbb{E}[v(T_k)] \leq \mathbb{E}[v(T_{k+1})]$ for all $k \Rightarrow$

For all $i > k$, $\qquad R + \lambda v(0) \leq \lambda \mathbb{E}[v(T_k)] \leq \lambda \mathbb{E}[v(T_i)] = \lambda \sum_{j=0}^{\infty} P_{i,j} v(j) \Rightarrow d^*(i) = 1$.

$\Rightarrow (*)$ holds, $i' = k$

Question 7

For $i' = \infty$ we need a case where replacement cost is sufficiently high compared to expected operating cost that replacement is never the optimal action, i.e. $R+\lambda v(0) > \lambda \sum_{j=0}^{\infty} P_{i,j} v(j)$ for all i.

For example, choose:
$R = 1000,\ C(i) = \begin{cases} 0 & i = 0 \\ 1 & i > 0 \end{cases},\quad P_{i,j} = \begin{cases} 0.9 & j = 0 \\ 0.1 & i = 0, j = 1 \textit{ or } i > 0, j = i \end{cases}$
In this case $v(i)$ is constant for all $i > 0$ as rewards and transitions symmetrical - call this $v(k)$, giving:
$v(0) = min\{1000 + \lambda v(0), \lambda(0.9v(0) + 0.1v(k))\}$,
$v(k) = min\{1000 + \lambda v(0), 1 + \lambda(0.9v(0) + 0.1v(k))\}$
Working this through gives $d^*(i) = 0$ for all i (i.e. never replace).

Question 8

Based on the properties above, we can consider the process instead as a series of cycles, each consisting the machine starting in state 0, changing condition until reaching a state $i > k$, at which point the machine is replaced (returning to 0 to begin a new cycle). By simulating the cycle cost incurred with varying k values, we should be able to find i' by seeking the k which minimises the expected (discounted) per step cycle cost. This approach ignores the starting state (starting all cycles from 0), but as this is an infinite horizon model this should not affect the optimal policy.

The following starts from $k = 0$ and simulates per step cycle cost for a cycle where replace is chosen for all states $i > k$. This cost should decrease with increasing k until $k = i'$, then increase thereafter. i' can be estimated by running the following many times and finding average of i'_{est}.

```
1.   $k = 0, AC_{old} = R + C(0)$
2.   $i = 0, t = 0, v = C(0)$
3.   while $i < k$ :
         $t = t + 1$
         $i = Y(i)$                          where $Y \equiv$ r.v. $T_i$, drawn from distribution $p(.|i, 0)$
         $v = v + \lambda^t C(i)$
     $AC_{new} = (v + \lambda^t R)/(t + 1)$        replace on last step
4.   if $AC_{new} > AC_{old}$ : $i'_{est} = k$, stop.
     else: $k = k + 1, AC_{old} = AC_{new}$, go to Step 2.
```

Question 9

Theorem 6.3.3. Let $v^0 \in V$, let $\{v^n\}$ be iterates of value iteration algorithm. Then the algorithm has the following properties:

a. it converges linearly at rate λ
i.e. v^n converges in norm to v^*_λ AND $\lambda = min\{K : \forall n,\ \|v^{n+1} - v^*_\lambda\| \leq K\|v^n - v^*_\lambda\|\}$

b. its asymptotic average rate of convergence (AARC) is λ
i.e. v^n converges in norm to v^*_λ AND

$$\limsup_{n\to\infty} \left(\frac{\|v^n - v^*_\lambda\|}{\|v^0 - v^*_\lambda\|} \right)^{1/n} = \lambda$$

c. its convergence is $O(\lambda^n)$, i.e.

$$\exists K < \infty : \limsup_{n\to\infty} \frac{\|v^n - v^*_\lambda\|}{\lambda^n} \leq K$$

d. for all n,

$$\|v^n - v^*_\lambda\| \leq \frac{\lambda^n}{1-\lambda}\|v^1 - v^0\|$$

e. for $d_n \in argmax_{d\in D}\{r_d + \lambda P_d v^n\}$,

$$\|v^{d_n}_\lambda - v^*_\lambda\| \leq \frac{2\lambda^n}{1-\lambda}\|v^1 - v^0\|$$

d. and e. provide bounds on the quality of the estimate obtained after a certain number of iterations of the algorithm. For example, this may be used to estimate the number of iterations required to get within ϵ of v^*_λ.

a. and b. describe the rate of convergence, with AARC describing the long-run convergence behaviour (potentially faster than rate of linear convergence, as a) must hold for all n). In this case alogorithm convergence monotonicly, linear rate of convergence and AARC are the same.