MATH4406: Assignment 5

Ryan Heneghan // 41778535

October 7, 2014

Question 1

A company buys a new bus with cost $R = 50,000$. The company puts bus related expenses on a credit card that must be paid at the end of every month. At the end of each month, the company evaluates the bus for breakages, wear and tear etc. Let $i \in \{0, 1, 2, ...\}$ be the number of breakages found at evaluation. The credit card bill is $C(\cdot)$, which is assumed to be an increasing function in i. That is, the more things that have gone wrong in the month prior, the higher the bus related expenses on the credit card. The company knows that as i increases, that is as the buses' condition deteriorates, the expected number of breakages (and therefore $C(\cdot)$) will increase further in future evaluations. Consequently, the company must decide at each monthly evaluation whether they will run the bus for another month or pay a fixed price R for a brand new bus. Let $C(\cdot)$ and $P_{i,j}$ be defined as follows:

$$C(i) = 1000(i+1), \qquad \text{and} \quad P_{i,j} = \begin{cases} 0.9 & j = i \\ 0.07 & j = i+1 \\ 0.03 & j = i+2 \end{cases}$$

Question 2

Machines naturally deteriorate over time and so will become increasingly costly to run. So, as time increases the probability of the machine being in a state that is worse than some arbitrary state k increases. That is, $\mathbb{P}(T_{i+1} > k) \geq \mathbb{P}(T_i > k)$, where T_i is a random variable representing the next state visited after state i.

Question 3

States: $s \in S = \{0, 1, 2, ...\}$.

Actions: $a \in \{0, 1\}$.

Transition Probabilities:

$$\mathbb{P}(j|s,a) = \begin{cases} P_{s,j} & a = 0,\ \forall j \in S, \\ 1 & a = 1,\ j = 0. \end{cases}$$

Rewards:

$$r(s,a) = \begin{cases} -R - C(s) & a = 1,\ \forall s \in S, \\ -C(s) & a = 0,\ \forall s \in S. \end{cases}$$

Objective Function:
The discounted value function is $v_\lambda^\pi(s)$. The goal in this scenario would be to find the policy $\pi^* \in \pi^{HR}$ that finds the smallest value function $v_\lambda^*(s)$, or equivalently the policy that maximises rewards.

$$v_\pi^\lambda(s) = \lim_{N\to\infty} \mathbb{E}_s^\pi \left\{ \sum_{t=1}^{N} \lambda^{t-1} r(X_t, Y_t) \right\},$$

where,

$$v_\lambda^*(s) = \sup_{\pi\in\pi^{HR}} v_\lambda^\pi(s).$$

Question 4

$$\begin{aligned} v(s) &= \sup_{a\in A_s} \left\{ r(s,a) + \sum_{j\in S} \lambda \mathbb{P}(j|s,a) v(j) \right\}, \\ &= \max \left\{ r(s,1) + \lambda \sum_{j=0}^{\infty} \mathbb{P}(j|s,1) v(j),\ r(s,0) + \lambda \sum_{j=0}^{\infty} \mathbb{P}(j|s,0) v(j) \right\}, \\ &= \max \left\{ -C(s) - R + \lambda v(0),\ -C(s) + \lambda \sum_{j=0}^{\infty} \mathbb{P}(j|s,0) v(j) \right\}. \end{aligned}$$

The value function is always negative with respect to the maximisation operator. That is, we are looking for the largest negative value function, with respect to the maximisation operator. If we let $v(\cdot) = -v(\cdot)$, then this is equivalent to finding the smallest value function with respect to the minimisation operator. Let $v(\cdot) = -v(\cdot)$, then

$$v(s) = C(s) + \min \left\{ R + \lambda v(0),\ \lambda \sum_{j=0}^{\infty} \mathbb{P}(j|s,0) v(j) \right\}. \tag{1}$$

Question 5

Use the value iteration, $v_0(i) = C(i)$ and for $n \geq 1$,

$$v_n(i) = C(i) + \min\{R + \lambda v_{n-1}(0), \alpha \sum_{j=0}^{\infty} P_{i,j} v_{n-1}(j)\}, \tag{2}$$

to prove that $v(i)$ is an increasing function ($v(i)$ is defined by the value function with respect to the minimisation operator).

Proof:
We know that $C(i)$ is an increasing function with respect to i, therefore $v_0(i) = C(i)$ is an increasing function. For $n \geq 1$, the first term in the minimisation operator of (2) is constant

and positive for all i. Regarding the second term in the minimisation operator, for $n = 1$, this is an increasing, positive function with respect to i, due to the stochastic ordering of the MDP. Coupling this with the fact that $C(i)$ is an increasing function and by induction on n, $v_n(i)$ is an increasing function with respect to i for all $n \geq 0$. Furthermore, because

$$\lim_{n\to\infty} v_n(i) = v(i),$$

it can be conluded based on the value iteration that $v(i)$ is an increasing function with respect to i.

Question 6

Using the optimality equation for this MDP, we replace when

$$R + \lambda v(0) \leq \lambda \sum_{j=0}^{\infty} P_{i,j} v(j),$$

and do not replace when

$$R + \lambda v(0) > \lambda \sum_{j=0}^{\infty} P_{i,j} v(j).$$

We need to prove there exists some $k < \infty$ such that we replace when $i \geq k$ and do not replace when $i < k$.

Proof:
We've established that $v(i)$ is an increasing function with respect to i. What's more, we've also established that because of the stochastic order of the MDP, $\sum_{j=0}^{\infty} P_{i,j}$ is an increasing function with respect to i. Therefore, provided $\lambda > 0$, $\lambda \sum_{j=0}^{\infty} P_{i,j} v(j)$ is a monotone increasing function with respect to i. Because $R + \lambda v(0)$ is a constant, there exists a point $k < \infty$ such that for all $i \geq k$,

$$R + \lambda v(0) \leq \lambda \sum_{j=0}^{\infty} P_{i,j} v(j),$$

which means that we replace the machine when $i \geq k$. Similiarly, for $i < k$,

$$R + \lambda v(0) > \lambda \sum_{j=0}^{\infty} P_{i,j} v(j),$$

which means we do not replace when $i < k$.

Question 7

In question 6, $k = \infty$ if $\lambda = 0$, since $R > 0$, the machine would never be replaced. Alternatively, $k = \infty$ if $P_{i,0} = 1$ for all i, that is the machine never wears out and is always in state 0.

Question 8

An algorithm for finding k, as defined in question 6.

1. Run the value iteration from question 5 for i up to some large number h (say $h = 10,000$), to obtain an estimate for $v(i)$ up to $i = h$. If you are sent here from step 6, go to step 3, else go to step 2.
2. Initiate $v(i)$ with $i = 1$. Set α to some small cut-off value (α is the cut-off for all $P_{i,j} \approx 0$). Calculate $R + \lambda v(0)$. Go to state 4.
3. Initiate $v(i)$ with $i = m + 1$, where m is the state you stopped at (the old h) in step 6 immediately before going to step 1 the last time. Calculate $R + \lambda v(0)$.
4. Calculate $\lambda \sum_{j=0}^{\infty} P_{i,j} v(j)$.
5. If $R + \lambda v(0) > \lambda \sum_{j=0}^{\infty} P_{i,j} v(j)$, go to step 6. Else, stop and conclude $k = i$.
6. If $i = h$, return to step 1 and increase the value of h. If $i < h$, set $i = i + 1$ and return to 4.

Question 9

a) Convergence is linear at rate λ:
Linear convergence for $\{v_n\}$ means there exists a λ and a value v^* such that,

$$||v_{n+1} - v_\lambda^*|| \leq K||v_n - v_\lambda^*||^\alpha, \qquad \text{where } \alpha = 1.$$

Where λ is the smallest value K can take such that the inequality above still holds. In terms of application, this means that the smaller λ is, the faster the rate of convergence.

b) The asymptotic average rate of convergence (AARC) is λ:
Mathematically, this is expressed as

$$\limsup_{n\to\infty} \left[\frac{||v_n - v_\lambda^*||}{||v_0 - v_\lambda^*||}\right]^{1/n} = \lambda.$$

The AARC can be used to find the expected number of iterations n necessary to reduce the error $||v_n - v_\lambda^*||$ by a fraction ϕ of the starting error $||v_0 - v_\lambda^*||$. If the AARC is λ, then n is calculated solving $\phi^{1/n} = \lambda$.

c) The value iteration converges $\mathbf{O(\lambda^n)}$:
Mathematically, this is expressed as

$$\limsup_{n\to} \frac{||v_n - v_\lambda^*||}{\lambda^n} \leq ||v_0 - v_\lambda^*||.$$

This means that convergence of the value iteration algorithm is geometric at rate λ. The order $O(\lambda^n)$ measures the 'worst case performance' of the rate of convergence of the value iteration algorithm. The inequality above implies that the value iteration algorithm decreases monotonically at at least rate λ^n.

d) for all n:

$$||v^n - v_\lambda^*|| \leq \frac{\lambda^n}{1-\lambda}||v_1 - v_0||.$$

This inequality shows that the error of the value iteration on the n^{th} iteration is bounded by a monotonically decreasing function. This inequality could be used to find an estimate of the number of additional iterations required to get a 'good' approximation of v_λ^*.

e) for any $\mathbf{d_n} \in \arg\max_{\mathbf{d}\in\mathbf{D}}\{\mathbf{r_d} + \lambda\mathbf{P_d}\mathbf{v^n}\}$:

$$||v_\lambda^{(d_n)^\infty} - v_\lambda^*|| \leq \frac{2\lambda^n}{1-\lambda}||v_1 - v_0||.$$

This inequality shows that the error of the optimal deterministic policy for the n^{th} iteration of the value iteration algorithm is bounded by a monotonically decreasing function. Further, using this formula one can find an estimate of the number of additional estimates necessary to obtain a ϵ-optimal policy.

Question 10

a)
For any v_0 in V, the iterates of value iteration satisfy

$$||v_{n+1} - v_\lambda^*|| = ||Lv_n - Lv_\lambda^*|| \leq \lambda||v_n - v_\lambda^*||. \tag{3}$$

Choosing $v_0 = v_\lambda^* + ke$, where k is a nonzero scalar, gives

$$v_1 - v_\lambda^* = \lambda(v_0 - v_\lambda^*).$$

Thus for this sequence (3) holds with equality so the rate convergence of value iteration equals λ.

b)
Iterate (3) and divide both sides by $||v_0 - v_\lambda^*||$, then take the n^{th} root:

$$||v_n - v_\lambda^*|| \leq \lambda||v_{n-1} - v_\lambda^*|| \leq ... \leq \lambda^n||v_0 - v_\lambda^*||,$$
$$||v_n - v_\lambda^*|| \leq \lambda^n||v_0 - v_\lambda^*||,$$
$$\limsup_{n\to\infty}\left[\frac{||v_n - v_\lambda^*||}{||v_0 - v_\lambda||}\right]^{1/n} \leq \lambda.$$

c)
Iterate (3) and divided both sides by λ^n:

$$||v_n - v_\lambda^*|| \leq \lambda^n||v_0 - v_\lambda^*||,$$
$$\limsup_{n\to\infty}\frac{||v_n - v_\lambda^*||}{\lambda^n} \leq ||v_0 - v_\lambda^*||.$$

d)

$$\begin{aligned}||v_n - v_\lambda^*|| &\leq ||v_{n+1} - v_\lambda^*|| + ||v_{n+1} - v_n||, \\ &\leq \lambda||v_n - v_\lambda^*|| + \lambda||v_n - v_{n-1}||,\end{aligned}$$

Rearranging and iterating the second component of the RHS of the inequality gives

$$||v_n - v_\lambda^*|| \leq \frac{\lambda}{1-\lambda}||v_n - v_{n-1}||,$$

$$||v_n - v_\lambda^*|| \leq \frac{\lambda^n}{1-\lambda}||v_1 - v_0||.$$

e)

$$||v_\lambda^{(d_n)\infty} - v_\lambda^*|| \leq ||v_\lambda^{(d_n)\infty} - v_{n+1}|| + ||v_{n+1} - v_\lambda^*||,$$

$$||v_\lambda^{(d_n)\infty} - v_\lambda^*|| \leq ||v_\lambda^{(d_n)\infty} - v_{n+1}|| + \frac{\lambda^n}{1-\lambda}||v_1 - v_0||.$$

Using the argument in the proof of Theorem 6.3.1 c) yields

$$||v_\lambda^{(d_n)\infty} - v_\lambda^*|| \leq \frac{\lambda}{1-\lambda}||v_n - v_{n-1}|| + \frac{\lambda^n}{1-\lambda}||v_1 - v_0||,$$

$$||v_\lambda^{(d_n)\infty} - v_\lambda^*|| \leq \frac{\lambda^n}{1-\lambda}||v_1 - v_0|| + \frac{\lambda^n}{1-\lambda}||v_1 - v_0||,$$

$$||v_\lambda^{(d_n)\infty} - v_\lambda^*|| \leq \frac{2\lambda^n}{1-\lambda}||v_1 - v_0||.$$

Question 11

Corollary 6.4.9 state that sufficient conditions for condition 6.4.15 to hold are that, for each $s \in S$,

1. A_s is compact and convex,
2. $p(j|s,a)$ is affine in a, and
3. $r(s,a)$ is strictly concave and twice continuously differentiable in a.

Condition 3 implies that policy iteration is not necessarily a better choice of algorithm if the reward function is just a constant value for all states and possible actions, since any constant would not be twice differentiable in in a.

A compact set is one that is closed and bounded, so for example if $A_s = [0, \infty)$ then quadratic convergence of the policy iteration algorithm cannot be established, and the algorithm may not be the best choice since it will not converge quadratically.