CS6700 Reinforcement learning
Assignment-I

Arulkumar S (CS15D202)

September 2, 2018

Problem 1

Consider a finite horizon MDP with N stages. Suppose there n possible states in each stage and m actions in each state. Why is the DP algorithm computationally less intensive as compared to an approach that calculates the expected cost J^π for each policy π? Argue using the number of operations required for both algorithms, as a function of m, n and N.

Solution

The naive approach of enumerating all possible actions and their resulting states at every time stage will yield exponential time complexity. Let there be n states, m actions per state and N stages in the given finite horizon MDP. The time complexity for calculating the expected cost J^π for each policy π is given by,

$$\text{Total expected cost} = m^{nN}$$

i.e., At each stage, for each state, there are m possible actions. Determining an optimal policy in such a way by enumerating all possible policies is computationally intensive.

By using Dynamic programming (DP) algorithm, a function/buffer $J_t(x_k)$ is defined for each state x_k at stage t to hold the cost of moving to that particular state from previous stage. The per-stage-cost function is defined as,

$$\begin{aligned} J_N(x_N) &= g_N(x_N) \\ J_t(x_t) &= min_{a_t} E_{x_{t+1}}\{g_t(x_t, a_t, x_{t+1}) + J_{t+1}(x_{t+1})\} \end{aligned}$$

Hence, to evaluate the value functions of all states at every state from backwards, for each state x_t and every action a_t at stage t, we have to aggregate over all the states x_{t+1} at stage $t+1$. The total time complexity is Nmn^2 which is much lesser than the naive version of evaluating policy by enumerating all policies.

Problem 2

Solution:

The expected cost objective $J_\pi(x_0) = E\left[exp\left(g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), x_{k+1})\right)\right]$

i) To prove the DP algorithm, we can use the induct ,ion method of proof. Let,

$$J_N(x_N) = exp(g_N(x_N))$$

Now we assume that $J_{k+1}(x_{k+1}) = J_{k+1}(x_{k+1})$.

$$
\begin{aligned}
J_k(x_k) &= min_{a_k \in A(x_k)} E\left[exp(g_N(x_N)).exp(g_k(x_k,a_k,x_{k+1}))\prod_{j=k+1}^{N-1} exp(g_j(x_j,a_j,x_{j+1}))\right] \\
&= min_{a_k \in A(x_k)} E\left[exp(g_k(x_k,a_k,x_{k+1})).exp\left(g_N(x_N)+\sum_{j=k+1}^{N-1} g_j(x_j,a_j,x_{j+1})\right)\right] \\
&= min_{a_k \in A(x_k)} E\left[exp(g_k(x_k,a_k,x_{k+1})).J_{k+1}(x_{k+1})\right]
\end{aligned}
$$

ii) Consider that the single stage cost g_k is a function of x_k and a_k only. let $V_N(x_N) = logJ_N(x_N) = g_N(x_N)$. The cost of a particular state x_k at stage k is given by,

$$
\begin{aligned}
V_k(x_k) &= min_{a_k \in A(x_k)} \log E_{x_{k+1}}\left[exp(g_k(x_k,a_k)).exp\left(g_N(x_N)+\sum_{j=k+1}^{N-1} g_j(x_j,a_j)\right)\right] \\
&= min_{a_k \in A(x_k)} \log exp(g_k(x_k,a_k)) + \log E_{x_{k+1}}\left[exp\left(g_N(x_N)+\sum_{j=k+1}^{N-1} g_j(x_j,a_j)\right)\right] \\
&= min_{a_k \in A(x_k)} \left(g_k(x_k,a_k)\right) + \log E_{x_{k+1}}\left[exp\left(V_{k+1}(x_{k+1})\right)\right]\Big)
\end{aligned}
$$

Problem 3

Solution:

There are two actions $A = \{buy, notbuy\}$ at every state x_k. Let x_{k+1} be the next state and T be the terminal state.

$$
x_{k+1} = \begin{cases} T, & \text{if } x_k = T \text{ (or) } (x_k \neq T \ \& \ a_k = buy) \\ N-k-1 & \text{otherwise} \end{cases}
$$

The associated cost at every stage is defined as below:

$$
g_N(x_N) = \begin{cases} \frac{1}{1-p} & \text{if } x_N \neq T \\ 0, & \text{otherwise} \end{cases}
$$

$$
g_k(x_k,a_k,x_{k+1}) = \begin{cases} px_k & \text{if } x_k = T \text{ and } a_k = buy \\ 0 & \text{otherwise} \end{cases}
$$

DP algorithm:

$$
\begin{aligned}
J_N(x_N) &= g_N(x_N) \\
J_k(x_k) &= \min_{a_k \in \{buy,notbuy\}} E_{x_{k+1}}\left[g(x_k,a_k,x_{k+1}) + J_{k+1}(x_{k+1})\right] \\
&= \begin{cases} min\left\{px_k, E(J_{k+1}(x_{k+1})\right\}, & \text{if } x_k \neq T, \\ 0, & \text{otherwise} \end{cases} \\
&= \begin{cases} min\left\{p(N-k), E(J_{k+1}(x_{k+1})\right\}, & \text{if } x_k \neq T, \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
$$

Policy: Buy if $p(N-k) \leq E(J_{k+1}(x_{k+1})$ else dont buy.

Problem 4

Suppose there are N jobs to schedule on a computer. Let T_i be the time it takes for job i to complete. Here T_i is a positive scalar. When job i is scheduled, with probability p_i a portion β_i (a positive scalar) of its execution time T_i is completed and with probability (1 - p_i), the computer crashes (not allowing any more job runs). Find the optimal schedule for the jobs, so that the total proportion of jobs completed is maximal.

Solution:

Let Z_i be the residual execution time of job $i \in \{1, 2, \dots N\}$. Consider two alternative schedules that job i is executed before job j and job i is executed after job j (interchanging argument). The number of jobs executed in these two policies respectively are:

$$J_{i \to j} = p_1\beta_1 Z_1 + p_2\beta_2 Z_2 + \dots + p_i\beta_i Z_i + p_i p_j \beta_j Z_j + \dots$$

$$J_{j \to i} = p_1\beta_1 Z_1 + p_2\beta_2 Z_2 + \dots + p_j\beta_j Z_j + p_i p_j \beta_i Z_i + \dots$$

comparing $J_{i \to j}$ and $J_{j \to i}$,

$$\begin{aligned}
p_1\beta_1 Z_1 + p_2\beta_2 Z_2 + \dots + p_i\beta_i Z_i + p_i p_j \beta_j Z_j + \dots &= p_1\beta_1 Z_1 + p_2\beta_2 Z_2 + \dots + p_j\beta_j Z_j + p_i p_j \beta_i Z_i + \dots \\
p_i\beta_i Z_i + p_i p_j \beta_j Z_j &= p_j\beta_j Z_j + p_i p_j \beta_i Z_i \\
p_i\beta_i Z_i + p_i p_j \beta_j Z_j &= p_j\beta_j Z_j + p_i p_j \beta_i Z_i \\
p_i(1 - p_j)\beta_i Z_i &= p_j(1 - p_i)\beta_j Z_j \\
\frac{p_i\beta_i Z_i}{(1 - p_i)} &= \frac{p_j\beta_j Z_j}{(1 - p_j)}
\end{aligned}$$

Let $B_k = \frac{p_k\beta_k Z_k}{(1-p_k)}$ and schedule the jobs based on non-decreasing B_k.

Problem 5

Solution:

It is given that the single stage cost is time invariant. i.e., $g_k = g$.

i) if $J_{N-1}(x) \le J_N(x)$ for all $x \in X$. Taking expectation on both sides and take minimum and add minimum of current stage cost,

$$\begin{aligned}
min_a E_a[g(x, a)] + min_x[J_{N-1}(x)] &\le min_a E_a[g(x, a)] + min_x[J_N(x)] \\
J_{N-2}(x) &\le J_(N - 1)(x)
\end{aligned}$$

Tracing back to kth stage, we can infer that $J_k(x) \le J_(k + 1)(x)$

ii) if $J_{N-1}(x) \ge J_N(x)$ for all $x \in X$. Taking expectation on both sides and take minimum and add minimum of current stage cost,

$$\begin{aligned}
min_a E_a[g(x, a)] + min_x[J_{N-1}(x)] &\ge min_a E_a[g(x, a)] + min_x[J_N(x)] \\
J_{N-2}(x) &\ge J_(N - 1)(x)
\end{aligned}$$

Tracing back to kth stage, we can infer that $J_k(x) \ge J_(k + 1)(x)$

Problem 6

Solution:

Let X be number of errors. Let N be the number of students/stages.

$$x_k = p_k E_k$$

p_k is the probability of the student k finding error, E_k is the number of errors found by student k.

per-stage costs:

$$g_N(x_N) = p_N x_N c_1 + (1 - P_N)(X - X_N)c_2 \quad (1)$$

$$g_k(x_k) = p_k x_k c_1 \quad (2)$$