Lecture 16: Average Reward MDP

Course: Reinforcement Learning Theory
Instructor: Lei Ying
Department of EECS
University of Michigan, Ann Arbor

Average Reward MDPs

$$\max \lim_{N\to\infty} \frac{1}{N+1} \sum_{k=0}^{N} E[r(x_k, u_k)]$$

- For simplicity, assume $r(x_k, u_k)$ is deterministic given x_k, u_k.
- The result can be extended to the case where $r(x_k, u_k)$ is a random variable.
- **Intuition:** Consider

$$\max \sum_{k=0}^{N} E[r(x_k, u_k)]$$

The Bellman equation is

$$J_0^N(i) = \max_u r(i,u) + \sum_j P_{ij}(u) J_1^N(j)$$

Average Reward MDPs

Suppose the average reward converges to J^*,

$$J_0^N(i) = (N+1)J^* + h_{N+1}(i) \Leftarrow h_{N+1}(i) = J_0^N(i) - (N+1)J^*$$

and

$$J_1^N(j) = NJ^* + h_N(j).$$

Substituting them into the Bellman equation, we have

$$(N+1)J^* + h_{N+1}(i) = \max_u r(i,u) + \sum_j P_{ij}(u)\left(NJ^* + h_N(j)\right)$$

which implies that

$$J^* + h_{N+1}(i) = \max_u r(i,u) + \sum_j P_{ij}(u) h_N(j)$$

- If as $N \to \infty$, $h_{N+1}(i) \to h(i)$ and $h_N(j) \to h(j)$, we have

$$J^* + h(i) = \max_u r(i,u) + \sum_j P_{ij}(u) h(j)$$

Average Reward MDPs

Theorem

If there exist J^* and h satisfying the above equation, then the obtained policy μ^* from this equation is the optimal stationary policy, and J^* is the optimal cost.

- Proof: Rewrite the Bellman equation as

$$\begin{aligned} J^* &= \max_u r(i,u) + E[h(x_{k+1})|x_k = i] - h(i) \\ &\geq r(i, \mu_k(i)) + E[h(x_{k+1})|x_k = i] - h(i) \quad \forall \mu_k \end{aligned}$$

Then,

$$J^* \geq E[r(x_k, \mu_k(x_k)] + E[h(x_{k+1})] - E[h(x_k)] \quad \forall \mu_k$$

$$NJ^* \geq \sum_{k=0}^{N-1} E[r(x_k, \mu_k(x_k))] + E[h(x_N)] - E[h(x_0)]$$

Average Reward MDPs

$$\frac{1}{N}\sum_{k=0}^{N-1} E[r(x_k, \mu_k(x_k))] \leq J^* - \frac{1}{N}E[h(x_N)] + \frac{1}{N}E[h(x_0)]$$

- If h is bounded or x_k takes values in a finite state space,

$$\lim_{N\to\infty} \frac{1}{N}\sum_{k=0}^{N-1} E[r(x_k, u_k)] \leq J^*$$

where reduces to equality for μ^*.

Average Reward MDPs

Theorem

If there exists a bounded h, J_1 such that

$$J_1 + h(i) \leq r(i,u) + \sum_j P_{ij}(u)h(j) \quad \forall i, u$$

Then $J_1 \leq J^*$.

Similarly, if the inequality is reversed, then $J_1 \geq J^*$.

Proof is similar to the previous proof.

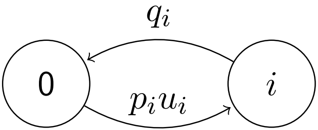
Example

A crowdsourcing worker is presented with type-i job with probability p_i. A job of type i can be completed in a time slot with probability q_i (independent of how long it has been with the worker) to complete.

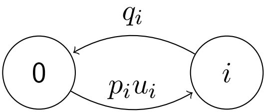
When the worker is working on a job, she cannot take on a new job. Find the optimal strategy to accept jobs to maximize average expected reward.

R_i: reward for completing a job of type i

Assume that if a job is accepted in time slot k, it cannot be completed in the same time slot.



Example



x_k: state of the system at the beginning of time slot k.

$x_k = 0$ means the worker is idle.

$x_k = i$ means the worker is working on a job of type i.

$$u_i = \begin{cases} 1, & \text{if job } i \text{ is accepted} \\ 0, & \text{otherwise} \end{cases}$$

$$\begin{cases} J^* + h(0) = \sum_i P_i \max(\underbrace{h(i)}_{\text{accept}}, \underbrace{h(0)}_{\text{reject}}), & \text{for state 0} \\ J^* + h(i) = q_i(R_i + h(0)) + (1 - q_i)h(i), & \text{otherwise} \end{cases}$$

Example

- Note: adding a constant c to $h(i)$ $\forall i$ does not change the above equations. So take $h(0) = 0$ WLOG.

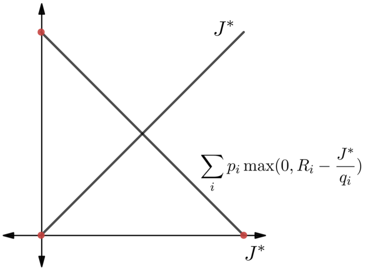
$$J^* = \sum_i p_i \max(0, h(i))$$

$$J^* + h(i) = q_i R_i + (1 - q_i) h(i)$$
$$J^* = q_i (R_i - h(i))$$

Note that $h(i) = R_i - \frac{J^*}{q_i}$,

$$J^* = \sum_i p_i \max(0, R_i - \frac{J^*}{q_i})$$

Example



An optimal J^* exists since the LHS is $\uparrow$ and RHS is $\downarrow$ in J^*.

$$h(i) = R_i - \frac{J^*}{q_i}$$

Thus the optimal policy is:

$$\begin{cases} \text{accept}, & \text{if } R_i \geq \frac{J^*}{q_i} \\ \text{reject}, & \text{otherwise} \end{cases}$$

Relative Value Iteration

Recall the value iteration algorithm:

- Set $J_0(i) = 0 \quad \forall i, \quad k = 0$
- $J_{k+1}(i) = \max_u r(i,u) + \sum_j P_{ij}(u) J_k(j)$

In value iteration, J_k can be thought of as the k-step reward. Thus, J_k will keep increasing and the procedure can be numerically unstable.

Relative Value Iteration

Let x' be an arbitrary state and define

$$\tilde{J}_k(i) = J_k(i) - J_k(x')$$

the relative cost w.r.t state x'. As k increases, if $\frac{J_k(i)}{k} \to J^*$ independent of i, the following procedure will be numerically stable:

- $\tilde{J}_0(i) = 0 \quad \forall i, \quad k = 0$
- $J_{k+1}(i) = \max_u r(i,u) + \sum_j P_{ij}(u)\tilde{J}_k(i)$
  $\tilde{J}_{k+1}(i) = J_{k+1}(i) - J_{k+1}(x')$
- repeat

Relative Value Iteration

- A variant of relative value iteration:

$$J_{k+1}(i) = \max_u r(i,u) + \sum_j P_{ij}(u) J_k(i) - J_k(x')$$

Policy iteration for average reward MDPs

- Fix policy μ_k
- Find J_k and h_k from

$$J_k + h_k = T_{\mu_k} h_k$$

$$h_k(0) = 0$$

- Obtain μ_{k+1} from

$$T_{\mu_{k+1}} h_k = T h_k$$

- If $J_{k+1} = J_k$ and $h_{k+1} = h_k$, stop.

Linear programming for average reward MDPs

- The LP formulation for average reward MDPs is given as:

$$\min_{J,h} J$$

$$\text{subject to } J + h(i) \geq r(i,u) + \sum_j P_{ij}(u)h(j) \quad \forall i, u$$

Average Reward Q-learning

- Note that

$$h(i) = \max_u \underbrace{r(i,u) + \sum_j P_{ij} h(j) - J^*}_{Q(i,u) \quad \text{(definition)}} = \max_u Q(i,u)$$

$$\begin{aligned} Q(i,u) &= r(i,u) + \sum_j P_{ij}(u) h(j) - J^* \\ &= r(i,u) + \sum_j P_{ij}(u) \max_v Q(j,v) - J^* \end{aligned}$$

- We have

$$Q(i,u) + J^* = r(i,u) + \sum_j P_{ij} \max_v Q(j,v)$$

Relative Value Iteration for Q

- Recall value iteration for h:

$$
\begin{aligned}
h_{k+1}(i) &= \max_u r(i,u) + \sum_j P_{ij}(u) h_k(j) - h_k(0) \\
&= \max_u r(i,u) + \sum_j P_{ij}(u) \max_v Q_k(j,v) - \max_v Q_k(0,v)
\end{aligned}
$$

- Thus we have the following value iteration algorithm for Q:

$$
Q_{k+1}(i,u) = r(i,u) + \sum_j P_{ij}(u) \max_v Q_k(j,v) - \max_v Q_k(0,v)
$$

Relative Value Iteration for Q

- Relative value iteration Q-learning:

$$
\begin{aligned}
Q_{k+1}(i,u) = &(1-\epsilon_k)Q_k(i,u) + \\
&\epsilon_k \left(r(i,u) + \sum_j P_{ij}(u) \max_v Q_k(j,v) - \max_v Q_k(0,v) \right).
\end{aligned}
$$

Reference

- This lecture is based on R. Srikant's lecture notes on *Average Cost MDPs* available at `https://sites.google.com/illinois.edu/mdps-and-rl/lectures?authuser=1`

Acknowledgements: I would like to thank Alex Zhao for helping prepare the slides, and Honghao Wei and Zixian Yang for correcting typos/mistakes.