Algorithms for Infinite Horizon MDPs

Lecturer: Daniel Russo *Scribe: Kumar Goutam, Apurv Shukla, Raghav Singal*

1 Introduction

We briefly review some material covered in the last lecture.
We characterize an infinite horizon discounted MDP $M = \{X, U, \gamma, g, P\}$, where:

- X represents the state space
- U the control space
- γ is the discounting factor
- $g(x, u)$ the cost function, wherein we assume discounted costs i.e. $g_k(x, u) = \gamma^k g(x, u)$
- $P(x, u, x')$ is the 3-dimensional array denoting the transition probability from x to x', when action u is taken.

We assume the state space, control space and transition probability matrix to be stationary. We define the optimal cost as $\min_{u\in U} \lim_{N\to\infty} \mathbb{E}(\sum_{k=0}^{N} \gamma^k g(x_k, u_k))$.
For a stationary policy $\mu : x \to U(x)$, we define the Bellman operator for the policy μ as: $T_\mu : J \in \mathbb{R}^{|X|}$ as $T_\mu J(x) = \mathbb{E}[g(x, \mu(x)) + \gamma \sum_{x'} P(x, \mu(x), x')J(x')]$ and the Bellman operator $T : J \in \mathbb{R}^{|X|} \to TJ \in \mathbb{R}^{|X|}$ by $TJ(x) = \min_{u\in U(x)} \mathbb{E}(g(x, u) + \gamma \sum_{x'} P(x, u, x')J(x'))$. Some important properties of the Bellman operator are:

- Monotonicity: For any $J \leq J'$ we have $T_\mu J \leq T_\mu J'$
- Contraction: $\|T_\mu J - T_\mu J'\|_\infty \leq \alpha\|J - J'\|$, $\alpha < 1$, $\|J\|_\infty = \max_{x\in X} J(x)$
- $\forall J, \mu$, $TJ \leq T_\mu J$. For any J, there exists μ such that $TJ = T_\mu J$

Using the Bellman operators, we obtain the optimal cost-to-go function and the optimal policy by:

- The optimal cost-to-go function is the unique solution of the fixed point equation $TJ = J$
- Once we know J^*, we obtain the optimal policy μ^* by solving a one-step look ahead problem w.r.t J^* i.e. $T_\mu J^* = TJ^*$

While programming in algorithms for MDP we input tolerance ϵ and $\{X, U, \gamma, g, P\}$, specifying them as :

- $X = \{1, 2, \ldots, n\}$
- $U = \{1, 2, \ldots, m\}$
- $g \in \mathbb{R}^{n\times m}$: where $g(x, u)$ is the expected instantaneous cost when we take action u in state x.
- $P \in \mathbb{R}^{n\times m\times n}$: $P(x, u, x') = P(x_{k+1} = x'|x_k = x, u_k = u)$

In the next step we perform consistency checks on the algorithm, namely dimensional consistency and checking for the fact that the sum of transition probability from every state is 1, i.e. $\sum_{x',u} P(x, u, x') = 1 \forall x, u$. At the end of computation we return a policy μ and perhaps a cost-to-go function. Three popular algorithms for solving MDPs are:

- Value Iteration
- Policy Iteration
- Linear Programming for MDPs

2 Policy evaluation

For a given policy $\mu : X \to U$, how do we find the corresponding J_μ? We define the following terms:

$$\begin{array}{ll} g_\mu \in \mathbb{R}^n, & g_\mu(x) = g(x, \mu(x)) \\ P_\mu \in \mathbb{R}^{n \times n}, & P_\mu(x, x^{'}) = P(x, \mu(x), x^{'}) \end{array}$$

Recall the definition of $T_\mu : \mathbb{R}^n \to \mathbb{R}^n$:

$$\begin{aligned} (T_\mu J)(x) &= g(x, \mu(x)) + \gamma \sum_{x^{'}} P(x, \mu(x), x^{'}) J(x^{'}) \\ &= g_\mu(x) + \gamma \sum_{x^{'}} P_\mu(x, x^{'}) J(x^{'}) \end{aligned}$$

Hence, we can re-write the above equation in matrix form:

$$T_\mu J = g_\mu + \gamma P_\mu J$$

We also know that J_μ solves $J = T_\mu J$. Hence, we solve the above system of linear equations (using any linear system solver of our choice) for J to get J_μ:

$$\begin{aligned} J_\mu &= \sum_{k=0}^{\infty} \gamma^k P_\mu^k g_\mu \\ &= (I - \gamma P_\mu)^{-1} g_\mu \end{aligned}$$

3 Value iteration

Recall the definition of $T : \mathbb{R}^n \to \mathbb{R}^n$:

$$(TJ)(x) = \min_u \{ g(x, u) + \gamma \sum_{x^{'}} P(x, u, x^{'}) J(x^{'}) \}.$$

Suppose the input policy is J_0. In value iteration, we repeat $J_k = TJ_{k-1}$ over k. The natural question is when do we stop. We know $J^* = TJ^*$. Accordingly, one might propose to stop when $J \approx TJ$ and output policy μ satisfying $T_\mu J = TJ$. Following three propositions prove that this is a “reasonable” stopping criterion.

Proposition 1. *If $||J - TJ||_\infty < \epsilon$, then $||J - J^*||_\infty < \frac{\epsilon}{1-\gamma}$.*

Proof.

$$\begin{aligned} ||J - J^*||_\infty &= ||J - TJ^*||_\infty \\ &= ||J - TJ + TJ - TJ^*||_\infty \\ &\le ||J - TJ||_\infty + ||TJ - TJ^*||_\infty \\ &< \epsilon + \gamma ||J - J^*||_\infty. \end{aligned}$$

Therefore, $||J - J^*||_\infty < \frac{\epsilon}{1-\gamma}$. □

Proposition 2. *Define policy μ as $T_\mu J = TJ$. If $||J - T_\mu J||_\infty < \epsilon$, then $||J - J_\mu||_\infty < \frac{\epsilon}{1-\gamma}$.*

Proof skipped in class.

Proposition 3. *$||J - TJ||_\infty < \epsilon$ and $T_\mu J = TJ$, then $||J^* - J_\mu||_\infty < \frac{2\epsilon}{1-\gamma}$.*

Proof. From proposition 1,

$$||J - J^*||_\infty < \frac{\epsilon}{1-\gamma}.$$

Moreover,

$$T_\mu J = TJ \Rightarrow ||T_\mu J - J||_\infty = ||TJ - J||_\infty < \epsilon.$$

Applying proposition 2, we get

$$||J - J_\mu||_\infty < \frac{\epsilon}{1-\gamma}.$$

Combining the above statements with triangle inequality, we get the required result. □

Algorithm 1 presents the pseudo-code for value iteration.

Algorithm 1 Value iteration

Require: J, ϵ

stop = False
while stop == False **do**
  $J' = TJ$
  if $||J - J'||_\infty \le \epsilon(1-\gamma)/2$ **then**
    stop = True
  end if
  $J = J'$
  $\mu(x) = \arg\min_{\mu}\{g(x,u) + \gamma \sum_{x'} P(x,u,x')J(x')\}$
end while
return (J, μ)

By propositions 1 and 3, $||J - J^*||_\infty < \frac{\epsilon}{2}$ and $||J_\mu - J^*||_\infty < \epsilon$.

4 Policy iteration

The following algorithm defines policy iteration:

- Input μ_0
- For $k = 0, 1, 2, \ldots$
  - solve $J_k = T_{\mu_k} J_k$ (just a linear system of equations)
  - Find μ_{k+1} as the solution to $T_{\mu_{k+1}} J_k = TJ_k$, i.e., solve $\operatorname{argmin}_u \{g(x,u) + \gamma \sum_{x'} P(x,u,x^{'})J_k(x^{'})\}$
  - If $J_k = TJ_k$, STOP and return μ_{k+1}

Proposition 4. *For the above algorithm, we have $J_0 \ge J_1 \ge J_2 \ge \ldots$.*

Proof. We have

$$\begin{aligned} J_{\mu_k} &= T_{\mu_k} J_k && \text{(by definition)} \\ &\ge TJ_{\mu_k} && \text{(T is minimum)} \\ &= T_{\mu_{k+1}} J_{\mu_k} && \text{(by algorithm design)} \end{aligned}$$

Applying $T_{\mu_{k+1}}$ on both sides, we get:

$$T_{\mu_{k+1}} J_{\mu_k} \ge T^2_{\mu_{k+1}} J_{\mu_k}$$

Continuing in a similar fashion, we would get:

$$J_{\mu_k} \geq T_{\mu_{k+1}} J_{\mu_k} \geq T^2_{\mu_{k+1}} J_{\mu_k} \geq \ldots \geq J_{\mu_{k+1}}$$

Since, $J_k = J_{\mu_k}$, we get the desired result. □

Proposition 5. *If $J_{k+1} = J_k$, then μ_{k+1} is optimal.*

Proof. If $J_{k+1} = J_k$, i.e., $J_{\mu_{k+1}} = J_{\mu_k}$, then in the previous proof, we would get equality everywhere. In particular, we would get:

$$J_{\mu_k} = TJ_{\mu_k} \Rightarrow J_{\mu_k} = J^*.$$

□

Corollary: Policy iteration terminates in finite time.

Proof. As there are at most a finite number of policies (as the state space and action space are both finite) and each time we get to see a new policy, the algorithm terminates in finite time. □

Proposition 6. *Policy iteration requires no more iterations than value iteration.*

Proof. Write $J_{\mu_k} = J_k$. We have, $J_k \geq TJ_k \geq J_{k+1}$. Hence, we get $J_1 \leq TJ_0, J_2 \leq TJ_1 \leq T^2 J_0$ and so on, which would lead to $J_K \leq T^k J_0$. So, $J^* \leq J_N \leq T^N J_0 \leq J_0$. □

5 Linear programming

To obtain the solution of MDP by linear programming, we introduce $\alpha \in \mathbb{R}^n, \sum_i \alpha_i = 1, \alpha_i > 0$. The alpha's can be interpreted as state relevant weights. Then, the optimal cost-to-go function of the MDP can be found by solving the following system:

$$\begin{aligned} & arg \max_J \alpha^T J \\ s.t. \quad & J \leq TJ \end{aligned} \tag{1}$$

Proof. Let J^* be the optimal cost-to-go function of the MDP. J^* is feasible ($J^* = TJ^*$) with cost $\alpha^T J$. Consider any other feasible J, then $J \leq TJ \leq T^2 J \ldots \leq T^N J \ldots \leq J^*$ where the first equality follows by definition of the constraint set, the subsequent inequalities from monotonicity and the final inequality by the taking $N \to \infty$ and using the property of Bellman operators. So J^* is the required optimal solution since $\alpha_i > 0$. □

However, the (1) is a non-linear system of equations, with $|X|$ non-linear inequalities and $|X||U|$ linear inequalities. (1) can be restated as:

$$\begin{aligned} \max_J \quad & \alpha^T J \\ & \text{s.t.} J(x) \leq min_{u \in U} g(x,u) + \gamma \sum_{x'} P(x,u,x')J(x'), \forall x \end{aligned} \tag{2}$$

Therefore we can introduce the all the constraints and write the LP as:

$$\max \quad \alpha^T J \tag{3}$$

$$\text{s.t.} J(x) \leq g(x,u) + \gamma \sum_{x'} P(x,u,x')J(x') \forall x, u \tag{4}$$

The Dual of the LP becomes:

$$\min_{\lambda} \sum_{x,u} \lambda(x,u) g(x,u) \tag{5}$$
$$\text{s.t.} \sum_{u} \lambda(x,u) = \alpha(x) + \gamma \sum_{x',u} P(x,u,x')\lambda(x',u) \forall x$$

Further noting that $\min \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k g(x_k, u_k)] \iff min \sum_{x,u}(\mathbb{E}[\sum_{k=0}^{\infty} \gamma^k \mathbb{1}\{x_k = x, u_k = u\}])g(x,u)$ we see that the cost function of the dual LP (5) corresponds to the optimal cost-to-go function of the MDP. A constrained MDP is a MDP where the cost is also a function of the amount of time spent in a state or might depend on the state-action pair.
The LP approach to approximately solving a DP was first presented by [Schweitzer and Seidmann, 1985] and a fuller theory was developed in [De Farias and Van Roy, 2003].
It was recently shown in [Ye, 2011] that the simplex method for LP form of MDPs with fixed discounting factor is strongly polynomial time with policy iteration requiring at most $\mathcal{O}(\frac{|X||U|}{(1-\gamma)} \log(\frac{|X|^2}{1-\gamma})$

References

[De Farias and Van Roy, 2003] De Farias, D. P. and Van Roy, B. (2003). The linear programming approach to approximate dynamic programming. *Operations research*, 51(6):850–865.

[Schweitzer and Seidmann, 1985] Schweitzer, P. J. and Seidmann, A. (1985). Generalized polynomial approximations in markovian decision processes. *Journal of mathematical analysis and applications*, 110(2):568–582.

[Ye, 2011] Ye, Y. (2011). The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603.