CS 287: Advanced Robotics
Fall 2009

Lecture 12: Reinforcement Learning

Pieter Abbeel
UC Berkeley EECS

Solving an MDP with linear programming

Outline

- LP approach for finding the optimal value function of MDPs
- Model-free approaches

Solving an MDP with linear programming

Solving an MDP with linear programming

The dual LP

The dual LP: interpretation

$$\max_{\lambda \geq 0} \sum_{s,a,s'} T(s,a,s')\lambda(s,a)R(s,a,s')$$
$$\text{s.t.} \quad \forall s \quad \sum_{a} \lambda(s,a) = c(s) + \sum_{s',a} \lambda(s',a)T(s',a,s)$$

- Meaning λ(s,a) ?
- Meaning c(s) ?

- **Value iteration:**
  - Start with V_0(s) = 0 for all s. Iterate until convergence:

$$V_{i+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_i(s')\right]$$

- **Policy iteration:**
  - Policy evaluation: Iterate until values converge

$$V_{i+1}^{\pi_k}(s) \leftarrow \sum_{s'} T(s,\pi_k(s),s') \left[R(s,\pi_k(s),s') + \gamma V_i^{\pi_k}(s')\right]$$

  - Policy improvement:

$$\pi_{k+1}(s) = \arg\max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V^{\pi_k}(s')\right]$$

- **Generalized policy iteration:**
  - Any interleaving of policy evaluation and policy improvement
  - Note: for particular choice of interleaving → value iteration
- **Linear programming:**

$$\min c^\top V \quad \text{s.t. } \forall s,a: \; V(s) \geq \sum_{s'} T(s,a,s') \left(R(s,a,s') + \gamma V(s')\right)$$

LP approach recap

The optimal value function satisfies:

$$\forall s : V(s) = \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V(s')\right].$$

We can relax these non-linear equality constraints to inequality constraints:

$$\forall s : V(s) \geq \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V(s')\right].$$

Equivalently, ($x \geq \max_i y_i$ is equivalent to $\forall i \;\; x \geq y_i$), we have:

$$\forall s, \forall a : V(s) \geq \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V(s')\right]. \quad (1)$$

The relaxation still has the optimal value function as one of its solutions, but we might have introduced new solutions. So we look for an objective function that will favor the optimal value function over other solutions of (1). To this extent, we observed the following monotonicity property of the Bellman operator T:

$$\forall s \;\; V_1(s) \geq V_2(s) \;\; \text{implies} : \;\; \forall s \;\; (TV_1)(s) \geq (TV_2)(s)$$

Any solution to (1) satisfies $V \geq TV$, hence also: $TV \geq T^2V$, hence also: $T^2V \geq T^3V \ldots T^{\infty-1}V \geq T^\infty V = V^*$. Stringing these together, we get for any solution V of (1) that the following holds:

$$V \geq V^*$$

Hence to find V^* as the solution to (1), it suffices to add an objective function which favors the smallest solution:

$$\min_V c^\top V \;\; \text{s.t.} \forall s, \forall a : V(s) \geq \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V(s')\right]. \quad (2)$$

If $c(s) > 0$ for all s, the unique solution to (2) is V^*.

Taking the Lagrange dual of (2), we obtain another interesting LP:

$$\max_{\lambda \geq 0} \sum_{s,a,s'} T(s,a,s')\lambda(s,a)R(s,a,s')$$
$$\text{s.t.} \quad \forall s \quad \sum_{a} \lambda(s,a) = c(s) + \gamma \sum_{s',a} \lambda(s',a)T(s',a,s)$$

What if T and R unknown

- Model-based reinforcement learning
  - Estimate model from experience
  - Solve the MDP as if the model were correct
- Model-free reinforcement learning
  - Adaptations of the exact algorithms which only require (s, a, r, s') traces [some of them use (s, a, r, s', a')]
  - No model is built in the process

Announcements

- PS 1: posted on class website, due Monday October 26.
- Final project abstracts due tomorrow.

Sample Avg to Replace Expectation?

$$V_{i+1}^{\pi}(s) \leftarrow \sum_{s'} T(s,\pi(s),s')[R(s,\pi(s),s') + \gamma V_i^{\pi}(s')]$$

- Who needs T and R? Approximate the expectation with samples (drawn from T!)

$$sample_1 = R(s,a,s'_1) + \gamma V_i^{\pi}(s'_1)$$
$$sample_2 = R(s,a,s'_2) + \gamma V_i^{\pi}(s'_2)$$
$$\ldots$$
$$sample_k = R(s,a,s'_k) + \gamma V_i^{\pi}(s'_k)$$

Problem: We need to estimate these too!

Sample Avg to Replace Expectation?

- We could estimate $V^\pi(s)$ for all states simultaneously:

Sample of V(s): $sample = R(s, \pi(s), s') + \gamma V^\pi(s')$

Update to V(s): $V^\pi(s) \leftarrow (1-\alpha)V^\pi(s) + (\alpha)sample$

Same update: $V^\pi(s) \leftarrow V^\pi(s) + \alpha(sample - V^\pi(s))$

- Old updates will use very poor estimates of $V^\pi(s')$
  - This will surely affect our estimates of $V^\pi(s)$ initially, but will this also affect our final estimate?

TD(0) for estimating V^π

Initialize $V(s)$ arbitrarily, π to the policy to be evaluated
Repeat (for each episode):
  Initialize s
  Repeat (for each step of episode):
    $a \leftarrow$ action given by π for s
    Take action a; observe reward, r, and next state, s'
    $V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$
    $s \leftarrow s'$
  until s is terminal

Note: this is really V^π

Sample Avg to Replace Expectation?

- Big idea: why bother learning T?
  - Update V(s) each time we experience (s,a,s')
  - Likely s' will contribute updates more often
- Temporal difference learning (TD or TD(0))
  - Policy still fixed!
  - Move values toward value of whatever successor occurs: running average!

Sample of V(s): $sample = R(s, \pi(s), s') + \gamma V^\pi(s')$

Update to V(s): $V^\pi(s) \leftarrow (1-\alpha)V^\pi(s) + (\alpha)sample$

Same update: $V^\pi(s) \leftarrow V^\pi(s) + \alpha(sample - V^\pi(s))$

Convergence guarantees for TD(0)

- Convergence with probability 1 for the states which are visited infinitely often if the step-size parameter decreases according to the "usual" stochastic approximation conditions

$$\sum_{k=0}^{\infty} \alpha_k = \infty$$

$$\sum_{k=0}^{\infty} \alpha_k^2 < \infty$$

- Examples:
  - 1/k
  - C/(C+k)

Exponential Moving Average

- Weighted averages emphasize certain samples

$$\frac{\sum_{i=1}^{n} w_i \cdot x_i}{\sum_{i=1}^{n} w_i}$$

- Exponential moving average
  - Makes recent samples more important

$$\bar{x}_n = \frac{x_n + (1-\alpha) \cdot x_{n-1} + (1-\alpha)^2 \cdot x_{n-2} + \dots}{1 + (1-\alpha) + (1-\alpha)^2 + \dots}$$

  - Forgets about the past (which contains mistakes in TD)
  - Easy to compute from the running average

$$\bar{x}_n = (1-\alpha) \cdot \bar{x}_{n-1} + \alpha \cdot x_n$$

- Decreasing learning rate can give converging averages

Experience replay

- If limited number of trials available: could repeatedly go through the data and perform the TD updates again
- Under this procedure, the values will converge to the values under the empirical transition and reward model.