Dynamic Programming Continued

& Recitation: MuJoCo setup

CSE599G: Deep Reinforcement Learning

Aravind Rajeswaran and Kendall Lowrey

University of Washington Seattle

April 4, 2018

This lecture was primarily done on the white board.

These slides are intended as a concise summary of key equations and ideas.

Overview

Last Class:

- Small and finite MDPs in the tabular case
- Efficient policy evaluation
- Policy iteration algorithm

Today:

- Monotonic improvement in policy iteration
- Convergence to globally optimal solution
- Recitation: setting up MuJoCo and code for Homework 1

Policy Iteration

Algorithm (review)

Initialize π_0 for all states (arbitrarily)

For i = 1,2,3, … (till convergence)

- Policy evaluation: compute the value of π_i (i.e. V^{π_i})
- Generate corresponding Q function
- Policy improvement: $\pi_{i+1} = \arg\max_a Q^{\pi_i}(s, a)$

Stop when policy does not change for any state

Policy Iteration

Each policy improvement step leads to monotonic improvement in the value

$V^{\pi_i}(s) \leq \max_a Q^{\pi_i}(s,a)$ Each equality/inequality written wrt previous expression

$= \max_a R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^{\pi_i}(s')$ Simply expanding out the Q function

$= R\left(s, \pi_{i+1}(s)\right) + \gamma \sum_{s'} P(s'|s,\pi_{i+1}(s)) \boldsymbol{V^{\pi_i}(s')}$ By definition of π_{i+1}, which takes the max action

$\leq R\left(s, \pi_{i+1}(s)\right) + \gamma \sum_{s'} P(s'|s,\pi_{i+1}(s)) \left(\boldsymbol{\max_{a'} Q^{\pi_i}(s',a')}\right)$ Max better than average (V)

$= R\left(s, \pi_{i+1}(s)\right) + \gamma \sum_{s'} P(s'|s,\pi_{i+1}(s)) \left(R\left(s', \pi_{i+1}(s')\right) + \gamma \sum_{s''} P(s''|s', \pi_{i+1}(s')) V^{\pi_i}(s'')\right)$

$= V^{\pi_{i+1}}(s)$

This holds for all states. So policy iteration leads to monotonic improvement.

Convergence of policy iteration

Most sensible MDPs have an upper bound on the reward (otherwise the value function can be infinite and thus ill defined).

If $R(s,a) \leq r_{max}\ \forall(s,a)$ then, $V^*(s) \leq \frac{1}{1-\gamma} r_{max}$

Since each step results in a monotonic improvement and since there is an upper bound, the algorithm has to converge at some point.

The rate of convergence of policy iteration is unknown. There is a linear lower bound (by a carefully constructed example), and there is a very loose exponential upper bound ($|S|^{|A|}$ - total number of policies for a finite MDP).

In practice, policy iteration is very fast. It is possible that the linear lower bound is in fact the upper bound and worst case.

Convergence to optimal solution

We will first consider a few theorems that will help the proof.

Theorem: The optimal policy satisfies the following recursive relationship

$$V^*(s) = \max_a Q^*(s,a) = \max_a R(s,a) + \gamma \sum_{s'} P(s'|s,a)V^*(s') \quad \forall s \in S$$

Proof: The optimal policy by definition is one that picks the best action according to the long term performance. If V^* and Q^* correctly measures the long term performance according to the optimal policy, the the correct first action for the optimal policy is one that maximizes the local Q function. The second equation simply follows from the recursive definition of Q.

Convergence to optimal solution

Theorem: A value function satisfies the following recursive relationship if and only if it is optimal.

$$V(s) = \max_a R(s,a) + \gamma \sum_{s'} P(s'|s,a)V(s')$$

Proof: We already showed that the optimal policy satisfies this recursion. To show the other part of the statement, we define the following operator for convenience:

$$Bf = \max_a R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, f$$

Hence, the above expression says that V is a fixed point of B: i.e. $BV = V$

Convergence to optimal solution

Theorem: B is a contraction mapping.

Proof: Let $|.|$ be the infinity norm.

$$|BV_i - BV_j| = \left|\left(\max_a R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V_i(s')\right) - \left(\max_{a^\sim} R(s,a^\sim) + \gamma \sum_{s'} P(s'|s,a^\sim)\, V_j(s')\right)\right|$$

$$\leq \left|\max_a R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V_i(s') - R(s,a) - \gamma \sum_{s'} P(s'|s,a^\sim)\, V_j(s')\right|$$

$$= \left|\max_a \gamma \sum_{s'} P(s'|s,a)\left(V_i(s') - V_j(s')\right)\right|$$

$$\leq \left|\max_a \gamma \sum_{s'} P(s'|s,a)\left|V_i - V_j\right|\right|$$

$$\leq \left|\gamma\left|V_i - V_j\right| \max_a \sum_{s'} P(s'|s,a)\right| = \gamma\left|V_i - V_j\right|$$

Convergence to optimal solution

We have showed that B is a contraction mapping, i.e.

$$\left|BV_i - BV_j\right| \leq \gamma |V_i - V_j|$$

Let us pick $V_j = V^*$ so that $BV_j = BV^* = V^*$

If V_i is another value function such that $BV_i = V_i$, then we have:

$$|BV_i - BV^*| = |V_i - V^*| \leq \gamma |V_i - V^*|$$

For $\gamma < 1$, this is possible only if $|V_i - V^*| = 0$

Hence, the fixed point of B is unique and corresponds to the optimal value function.

Convergence to optimal solution

Recall that we showed policy iteration converges to a place where the value function has the property:

$$V^{\pi}(s) = \max_{a} Q^{\pi}(s, a) = BV^{\pi}(s)$$

Hence, from the previous argument, we directly have that this value function is optimal. Since we pick the policy to be the greedy optimizer of this value function, the corresponding policy is optimal.

MuJoCo setup

- Download the MuJoCo binaries appropriate for your OS from:

  https://www.roboti.us/index.html
- Download the mujoco key (mjkey.txt) from canvas
- Clone this repository:
- Follow setup instructions from here:
- To check if things are working correctly, run the following file:

  $ python tests/visualize_test.py