Master MVA
Dynamic Programming & Reinforcement Learning
TP1 - 13/11/2016
Achari Berrada Youssef

1 The On-Site Tree Cutting Problem :

Q1: The state X and the action A space are :

$$
\begin{aligned}
X &= \{1, ..., \text{max\_height}, \text{sick\_state}\} & \\
A &= \{1, 2\} & \text{1 for keep and 2 for cut} \\
(P)_{i,j,a} &= p_{i,j,a} = \mathbb{P}(x_{t+1} = j | x_t = i, a_t = a) & \forall (i,j) \in X^2, a \in A \\
(R)_{i,a} &= r_{i,a} = r(i,a) & \forall (i,a) \in X \times A
\end{aligned}
$$

The matrix P of size $|X| \times |X| \times |A|$ models the random effects.

Q2: **Policy Evaluation:** For a stationary policy π :

$$V^\pi(x) = \mathbb{E}_{x_0=x}\left[\sum_{t=0}^{\infty} \gamma^t r(x_t, \pi(x_t))\right]$$

- Dynamic Programming:

$$
\begin{aligned}
&V^\pi(x) = r(x, \pi(x)) + \gamma \sum_{y \in X} p(y|x, \pi(x)) V^\pi(y) \\
&\Rightarrow \\
&V^\pi = R^\pi + \gamma P^\pi V^\pi \\
&\Rightarrow \\
&V^\pi = (I - \gamma P^\pi)^{-1} R^\pi
\end{aligned}
$$

- Reinforcement Learning with Monté-Carlo: In this method, we approximate the value with N trajectories, and each one has T steps.

$$\hat{V}^\pi = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T} \gamma^t r(x_t^i, \pi(x_t^i))$$

Q3: **Optimal Policy:**

- Value Iteration:
  1. For a given initial policy π_i, we compute the Value function V_0 using DP or RL methods.
  2. For $k = 1, .., K$, $V_{k+1} = \mathcal{T} V_k$ with $\mathcal{T}$ is the Bellman operator.
  3. Find the Greedy policy π_f with $\pi_f(x) = \underset{a \in A}{\text{argmax}}\, Q_K(x,a)$

     with :
     $$Q_k(x,a) = r(x,a) + \gamma \sum_{y \in X} p(y|x,a) V^k(y)$$

- Policy Iteration:
  1. Let π_0 be the initial stationary policy,

2. For $k = 0, .., K-1$, we evalute the policy π_k and compute V^{π_k}. Then we improve the policy by finding the greedy policy π_{k+1} with $\pi_{k+1}(x) = \underset{a \in A}{\operatorname{argmax}} Q_k(x, a)$
3. Return π_K.

Q4: Optimal policy:

- Value iteration is computationally efficient but the convergence is asymptotic.
- Policy iteration converge is a small number of iterations but each iteration requires a full policy evaluation which might be expensive.

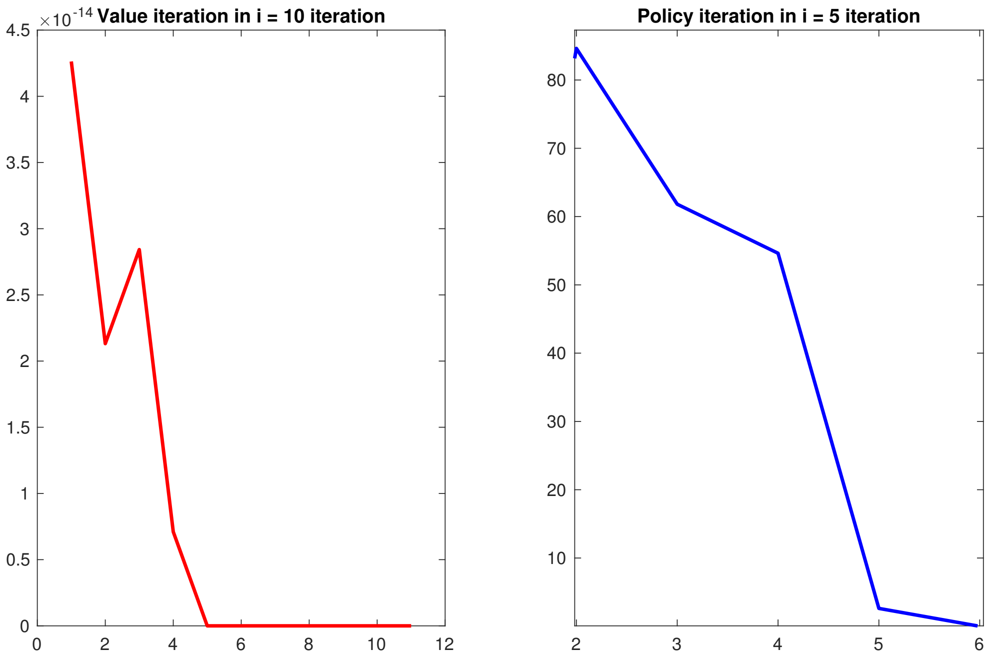

Figure 1: Value iteration error vs Policy iteration error