Lecture 4 - *Value and Policy Iteration*

OBJECTIVE: In this lecture, we will show that the Bellman operator is a contraction operator. Hence, it will cause the initial value function to contract toward the optimal value function. We then use these mathematical results to derive the most popular algorithms for MDPs: value iteration, policy iteration and backward induction.

◇ CONTRACTION OPERATORS

The Bellman operator (also known as the DP operator) is a **contraction mapping** with respect to the **weighted maximum norm**. That is, there exists a vector $\xi = (\xi(1), \ldots, \xi(n)) \in \mathbb{R}^n$, with positive components, and a

scalar $\lambda \in [0, 1)$, such that:

$$\|TV - T\overline{V}\|_\xi \leq \lambda \|V - \overline{V}\|_\xi$$

for all vectors V and $\overline{V}$, where the weighted maximum norm is defined by:

$$\|V\|_\xi = \max_{i=1:n} \frac{|V(i)|}{\xi(i)}$$

This is formalized by the following theorem.

Theorem 3 *Assume that the terminal state can be reached with a stationary policy. There exists a vector $\xi \in \mathbb{R}^n$ with positive entries such that the Bellman operators T and T_d, for all stationary policies d, are contraction operators with respect to the maximum norm $\|\cdot\|_\xi$ with $\lambda \in [0, 1)$.*

⋆ Proof:

⋆ Proof:

◇ VALUE ITERATION

The existence and uniqueness of the solution to the value fixed point equation $V^\star = TV^\star$ is the basis for the **value iteration algorithm**:

```
function [pi,V] = valueIteration(P, R,gamma)
% Given the (s x s' x a) matrices P and R and discount factor gamma, use value
% iteration to find a deterministic policy pi and value function V.

theta = 10e-6; delta = theta+1; NS = size(P,2); NA = size(P,3);
V = zeros(1,NS);        % INITIALIZATION

while delta >= theta    % VALUE ITERATION
    delta = 0;
    for s = 1:NS
        for a = 1:NA
            a_val(a) = sum(P(s,:,a) .* (R(s,:,a) + gamma * V));
        end
        newv(s) = max(a_val);
    end
    delta = max(abs(newv-V));
    V = newv;
end

for s=1:NS              % FINDING A POLICY
    for a = 1:NA
        a_val(a) = sum(P(s,:,a) .* (R(s,:,a) + gamma * V));
    end
    [val, pi(s)] = max(a_val);
end
```


◇ BACKWARD INDUCTION

Backward induction is a form of value iteration when there is a final reward $V^\star(\mathbf{x}_N) = r(\mathbf{x}_N)$ for all values of $\mathbf{x}_N$. The algorithm proceeds as follows:

1. At $t = N$, $V^\star(\mathbf{x}_N) = r(\mathbf{x}_N)$ for all $\mathbf{x}_N$.
2. For $t = N - 1 : 1$
   (a) Compute the optimal value function:

$$V_t^\star = \max_{\mathbf{a}_t} \sum_{\mathbf{x}_{t+1}} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{a}_t) \left[r(\mathbf{x}_t, \mathbf{a}_t, \mathbf{x}_{t+1}) + \gamma V_{t+1}^\star(\mathbf{x}_{t+1}) \right]$$

   (b) Compute the optimal policy:

$$\boldsymbol{\pi}^\star = \arg\max_{\mathbf{a}_t} \sum_{\mathbf{x}_{t+1}} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{a}_t) \left[r(\mathbf{x}_t, \mathbf{a}_t, \mathbf{x}_{t+1}) + \gamma V_{t+1}^\star(\mathbf{x}_{t+1}) \right]$$

This algorithm is often used to solve problems such as network routing, deterministic resource allocation and the secretary problem.

◇ POLICY ITERATION

The **policy iteration algorithm** is an alternative to value iteration that terminates finitely.

```
function [pi,V] = policyIteration(P, R, gamma)
NS=size(P,2);NA=size(P,3); theta = 10e-6;
V = zeros(1,NS); pi = ones(1,NS); policyStable = false; % INITIALIZATION

while policyStable == false   % POLICY EVALUATION: SOLVE LINEAR SYSTEM.
    delta = theta+1;
    while delta >= theta
        delta = 0;
        for s = 1:NS
            newv(s) = sum(P(s,:,pi(s)) .* (R(s,:,pi(s)) + gamma * V));
        end
        delta = max(abs(newv-V));
        V = newv;
    end
    policyStable = true;      % POLICY IMPROVEMENT
    for s=1:NS
        b = pi(s);
        a_val = zeros(1,NA);
        for a = 1:NA
            a_val(a) = sum(P(s,:,a) .* (R(s,:,a) + gamma * V));
        end
        [val, pi(s)] = max(a_val);
        if b ~= pi(s)
            policyStable = false;
        end
    end
end
```


The following theorem establishes the convergence of policy iteration in a finite number of steps.

Theorem 4 *There exists a natural number N_0 such that for all $t > N_0$ we have $V_{d_t} = V^\star$.*

⋆ Proof:

⋆ Proof: