Dynamic Programming Quiz

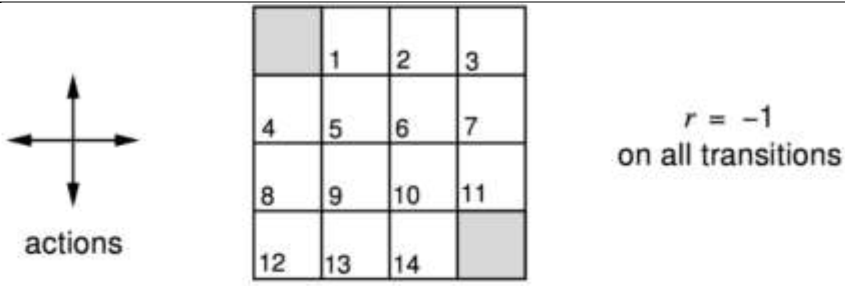
Reinforcement Learning – Move 37 [Week2]

Quiz on Policy Iteration, Policy Improvement, Policy Evaluation, Value Iteration
(5 Questions, 4 Possible Answers)

Collected by **Kurt** (Dean, Seoul School of AI, Korea)

V 1.4
9/16/2018

#Policy Evaluation



Undiscounted episodic MDP (γ = 1)

Nonterminal states 1, ..., 14

One terminal state (shown twice as shaded squares)

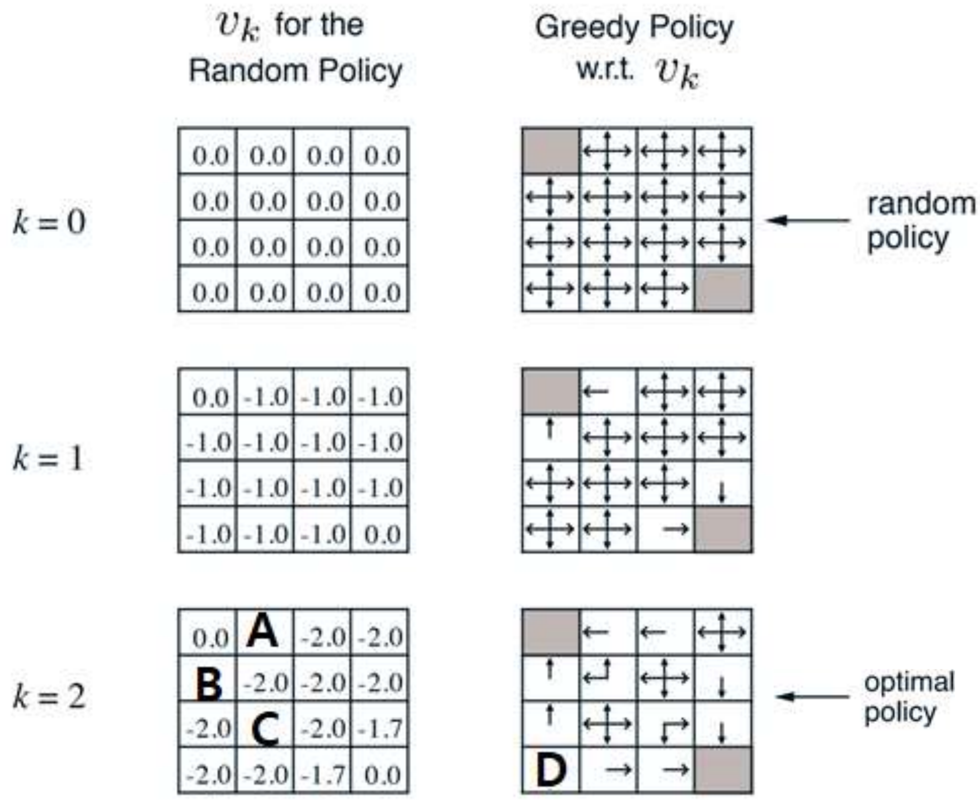
Actions leading out of the grid leave state unchanged

Reward is −1 until the terminal state is reached

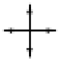
Agent follows uniform random policy

$\pi(n|\cdot) = \pi(e|\cdot) = \pi(s|\cdot) = \pi(w|\cdot) = 0.25$

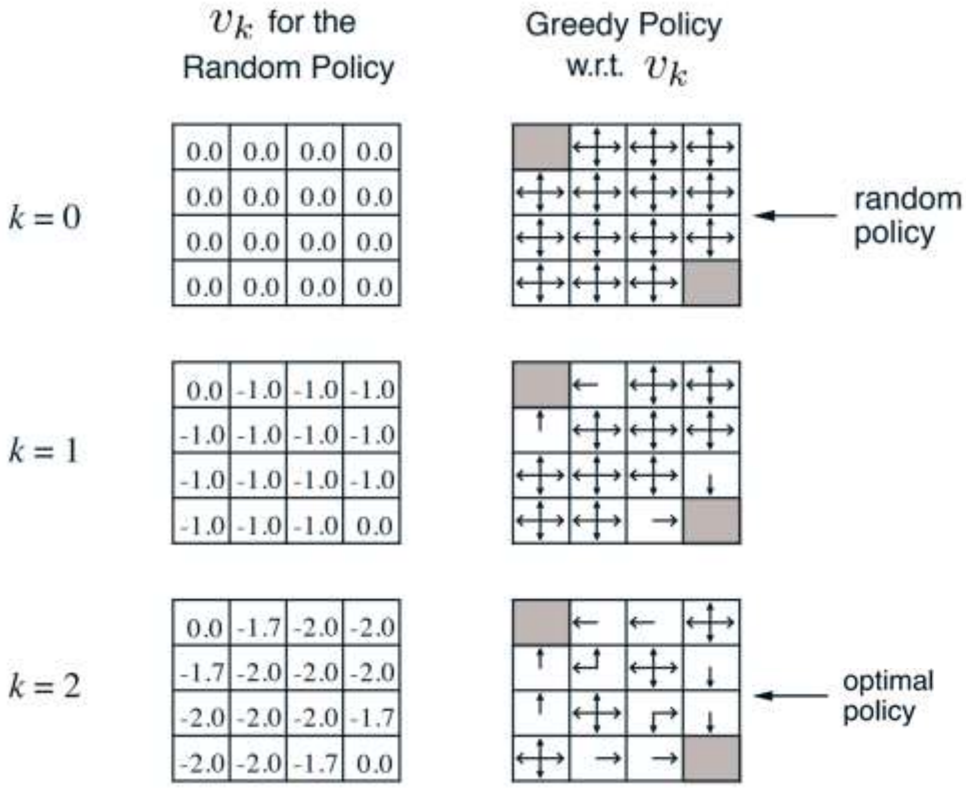
truncate to 1 decimal place



Q1. Choose all that apply. (4 possible answers)

1. A=-1.0
2. B=-2.0
3. C=-2.0
4. D= 

Q1 Explanation>



1. [x] A= -1.7 (truncated to one decimal place)

   v = 1 X 0.25 X (-1 + 0) + 3 X 0.25 X (-1 + -1)

2. [x] B= -1.7 (truncated to one decimal place)

   v = 1 X 0.25 X (-1 + 0) + 3 X 0.25 X (-1 + -1)

3. [o] C= -2.0

   v = 4 X 0.25 X (-1 + -1)

4. [o] same values of -2.0

Reference> Policy Evaluation

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/DP.pdf

#Policy Improvement #Policy Evaluation

Q2. Choose all that apply. (4 possible answers)

1. Policy π can be evaluated by $v_\pi(s) = \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \ldots | S_t = s\right]$
2. Process of policy iteration always converges.
3. Bellman expectation equation is used for Policy Improvement.
4. Policy evaluation $\mathbf{v}_{k+1} = \max_{a \in \mathcal{A}} \mathcal{R}^{a} + \gamma \mathcal{P}^{a} \mathbf{v}_k$

Q2 Explanation>

1. **[o]**

2. **[o]**

3. **[x]** **Bellman optimality equation** is used for Policy Improvement.

   Bellman expectation equation is $v_\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) q_\pi(s, a)$

   Bellman optimality equation is $v_\pi(s) = \max_{a \in \mathcal{A}} q_\pi(s, a)$

4. **[x]** value iteration

   Policy iteration is $v_{k+1}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_k(s') \right)$

$$\mathbf{v}^{k+1} = \boldsymbol{\mathcal{R}}^{\boldsymbol{\pi}} + \gamma \boldsymbol{\mathcal{P}}^{\boldsymbol{\pi}} \mathbf{v}^k$$

.

Reference> Policy Evaluation, Policy Iteration

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/DP.pdf

#Value Iteration

Gray g is the goal. start with final rewards and work backwards using value iteration.
Basic rule is same as Q1

g			

Problem

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

V_1

0	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1

V_2

0	-1	-2	-2
-1	-2	-2	-2
-2	-2	-2	-2
-2	-2	-2	-2

V_3

0	-1	-2	-3
-1	-2	-3	-3
-2	-3	-3	-3
-3	-3	-3	-3

V_4

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-4
-3	-4	-4	-4

V_5

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-5

V_6

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	**A**	**B**
-3	-4	**C**	**D**

V_7

Q3. Choose all that apply. (4 possible answers)

1. A = -5
2. B = -6
3. C =-5
4. D = -6

Q3 Explanation>

g			

Problem

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

V_1

0	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1
-1	-1	-1	-1

V_2

0	-1	-2	-2
-1	-2	-2	-2
-2	-2	-2	-2
-2	-2	-2	-2

V_3

0	-1	-2	-3
-1	-2	-3	-3
-2	-3	-3	-3
-3	-3	-3	-3

V_4

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-4
-3	-4	-4	-4

V_5

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-5

V_6

0	-1	-2	-3
-1	-2	-3	-4
-2	-3	-4	-5
-3	-4	-5	-6

V_7

Max(-1+Up, -1+Left, -1+Down, -1+Right) (using own value when it blocked to wall)

1. **[x] A = -4**

   Max(-3-1, -3-1 -5-1, -5-1)

2. **[x] B = -5**

   Max(-4-1, -4-1 -5-1, -5-1)

3. **[o] C =-5**

   Max(-4-1, -4-1 -5-1, -5-1)

4. **[o] D = -6**

   Max(-5-1, -5-1 -5-1, -5-1)

Reference> Value Iteration

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/DP.pdf

Q4. Choose all that apply. (4 possible answers)

1. If we know the solution to subproblems $v*(s')$ then solution $v*(s)$ can be found by **just one-step lookahead**
2. To find optimal policy π, use iterative application of Bellman Expectation Equation.
3. there is no explicit policy in the value iteration.
4. Value iteration is $\mathbf{v}_{k+1} = \max_{a \in \mathcal{A}} \mathcal{R}^{a} + \gamma \mathcal{P}^{a} \mathbf{v}_k$

Q4 Explanation>

1. **[o]** $v_*(s) \leftarrow \max_{a \in \mathcal{A}} \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s')$

2. **[x] Bellman optimality equation** is used **[see Q2]**

3. **[o]**

4. **[o] Value iteration** $v_{k+1}(s) = \max_{a \in \mathcal{A}} \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_k(s') \right)$

$$\mathbf{v}_{k+1} = \max_{a \in \mathcal{A}} \boldsymbol{\mathcal{R}}^{\mathbf{a}} + \gamma \boldsymbol{\mathcal{P}}^{\mathbf{a}} \mathbf{v}_k$$

Policy Iteration $v_{k+1}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_k(s') \right)$

$$\mathbf{v}^{k+1} = \boldsymbol{\mathcal{R}}^{\boldsymbol{\pi}} + \gamma \boldsymbol{\mathcal{P}}^{\boldsymbol{\pi}} \mathbf{v}^k$$

Reference> Value Iteration

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/DP.pdf

Q5. Choose all that apply. (4 possible answers)

Algorithm	Bellman Equation
Policy Evaluation	**A**
Policy Iteration	**B + C**
Value Iteration	**D**

1. A = Bellman Expectation Equation

2. B = Bellman Expectation Equation

3. C = Greedy Policy Improvement

4. D = Bellman Optimality Equation

Q5 Explanation >

Problem	Bellman Equation	Algorithm
Prediction	Bellman Expectation Equation	Iterative Policy Evaluation
Control	Bellman Expectation Equation + Greedy Policy Improvement	Policy Iteration
Control	Bellman Optimality Equation	Value Iteration

- Algorithms are based on state-value function $v_\pi(s)$ or $v_*(s)$
- Complexity $O(mn^2)$ per iteration, for m actions and n states
- Could also apply to action-value function $q_\pi(s, a)$ or $q_*(s, a)$
- Complexity $O(m^2n^2)$ per iteration

1. [o]
2. [o]
3. [o]
4. [o]

Reference> Value Iteration

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/DP.pdf