UMassAmherst
The Commonwealth's Flagship Campus

Solving MDP using Exhaustive Enumeration

Chaitra Gopalappa

Solution algorithms to solve MDPs?

- Exhaustive enumeration
- Dynamic Programming
- Reinforcement learning
- Deep reinforcement learning

Caution: Unless otherwise specified, here on:

- π
  - is a policy (not steady state distribution like in Markov chain)
- Value function= expected total returns (user choice for discounting):
  - $v_\pi(s) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s]; \forall s \in S$

Cancer screening guidelines

- When to start? How often to screen?
- X_t =health state at age t
- D_t =decision at age t
- $\{X_t, D_t\}_{t=a_1 \dots a_n}$ is a MDP formulated by 4-tuple $\{\Omega, A, P_a, R_a\}$
- $\Omega = \{\text{healthy}, \text{stage } 1, \text{stage } 2, \text{stage } 3, \text{stage } 4\}$
- $A = \{yes, no\}$
- P_a= TPM for action a
- R_a= TRM for action a

Bansal et. al., 20200, Medical Decision Making https://journals.sagepub.com/doi/10.1177/0272989X20910724
Evaluated under resource constraints to inform WHO Global Action Plan for UN SDGs

Exhaustive enumeration of value function using simulation

- $Suppose: S = \{s_1, s_2, s_3\};\ A = \{a, b\}$
- $P_a = \left[\quad\right]; P_b = \left[\quad\right]$
- $R_a = \left[\quad\right]; R_b = \left[\quad\right]$
- suppose, **policy**= $\boldsymbol{\pi} = [\boldsymbol{b}, \boldsymbol{a}, \boldsymbol{a}]$

$v_\pi(s_1)$=? for some fixed policy π

Start in s_1, apply policy π to transition

Episode	k→	0	1	2	3	4………	T	Total reward
1.	State→	s_1	s_1	s_2	s_3	s_1		
	Action (following π)→	b	b	a	a	b		
	Corresponding immediate reward (R)		$\gamma^0\, r(s_1, b, s_1)$	$\gamma^1 r(s_1, b, s_2)$	$\gamma^2 r(s_2, a, s_3)$	$\gamma^3 r(s_3, a, s_1)$		Sum this row
2.	State→	s_1	s_2	s_3	s_3	s_1		
	Action (following π)→	b	a	a	a	b		
	Corresponding immediate reward (R)		$\gamma^0\, r(s_1, b, s_2)$	$\gamma^1\, r(s_2, a, s_3)$	$\gamma^2\, r(s_3, a, s_3)$	$\gamma^3\, r(s_3, a, s_1)$		Sum this row
……								
								$v_\pi(s_1)$ $= expected\ value$ $of\ this\ column$

Repeat for every state, and every policy π

$$\pi^* = arg \max_\pi v_\pi(s)\,, \forall s \in\ S$$

Example

$$v_{\pi=[m,m,m,r]}(G) = ?$$

$P_{maintain} =$

	E	G	F	I
E	0.3	0.2	0.3	0.2
G	0.1	0.2	0.4	0.3
F	0	0.4	0.5	0.1
I	0	0	0	1

$P_{replace} =$

	E	G	F	I
E	1	0	0	0
G	1	0	0	0
F	1	0	0	0
I	1	0	0	0

$r(i, a = maintain, j)$

	E	G	F	I
E	-100	-1000	-1500	-100K
G	-50	-500	-550	-100K
F	-200	-2000	-2500	-100K
I	-300	-3000	-3500	-100K

$r\ (i, a = replace, j)$

	E	G	F	I
E	-6000	-6000	-6000	-6000
G	-6000	-6000	-6000	-6000
F	-6000	-6000	-6000	-6000
I	-6000	-6000	-6000	-6000

Episode	**k→**	**0**	**1**	**2**	**3**	**4………**	**T**	**Total reward**
1.	State→	G	F					
	Action (following π)→	m	**m**					
	Corresponding immediate reward (R)		-550					Sum this row
2.	State→	G						
	Action (following π)→							
	Corresponding immediate reward (R)							Sum this row
……								
								$v_\pi(s_1) = expected\ value\ of\ this\ column$

Exhaustive enumeration is computationally inefficient

- *Suppose:* $S = \{s_1, s_2, s_3\}$; $A = \{a, b\}$
- $P_a = \begin{bmatrix} \; \end{bmatrix}$; $P_b = \begin{bmatrix} \; \end{bmatrix}$
- $R_a = \begin{bmatrix} \; \end{bmatrix}$; $R_b = \begin{bmatrix} \; \end{bmatrix}$
- And suppose, policy= $\pi = [b, a, a]$, and we the following :

Repeat for each episode

Repeat for each state

$v_\pi(s_1)$=?

Start in s_1, apply π to transition

$v_\pi(s_2)$=?

Start in s_2, apply π to transition….

.

.

Number of evaluations:

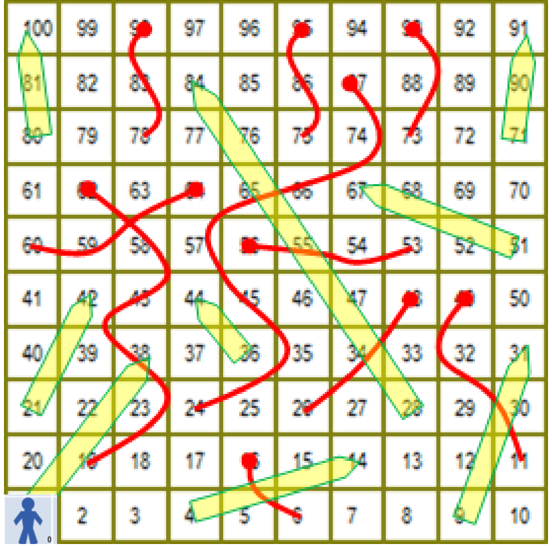
- Number of possible policies = $|A|^{|S|}$
- For each policy
  - $|S|$ number of simulations (each simulation initialized to corresponding state)
    - Each simulation generated for a large number of episodes

Alternative (more efficient) solution methods

- Model-based:
  - Dynamic programming (iteratively solve for the value function)
- Model-free:
  - Reinforcement learning
  - Deep reinforcement learning

Discounting

Move robot on chutes and ladder: Objective is to reach 100



$$S = \{1, \dots, 100\}$$

$$A = \{Left, Right, Up, Down\}$$

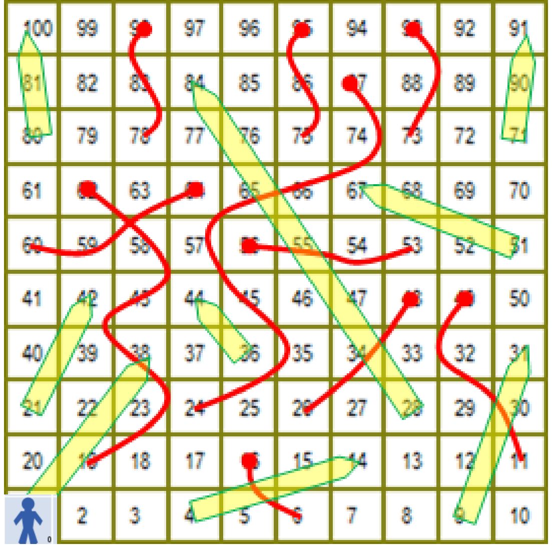
$$r(i,a,j) = \begin{cases} 1000\ is\ j = 100 \\ 0\ otherwise \end{cases}$$

Policy	Value
$\pi = [U, U, \dots\dots, U]$	$v_\pi(1)$=?
$\pi(1) = \cdots = \pi(9) = R;$ $\pi(10) = \cdots = \pi(70) = U;$ $\pi(71) = \cdots = \pi(79) = L$ $\pi(80) = \cdots = \pi(100) = U$	$v_\pi(1)$=?
$\pi(1) = \cdots = \pi(9) = R;$ $\pi(10) = \cdots = \pi(90) = U;$ $\pi(91) = \cdots = \pi(100) = L$	$v_\pi(1)$=?
…….	…….

$$\pi^* = arg \max_{\pi} v_\pi(1)$$

Move robot on chutes and ladder: Objective is to reach 100

Policy	Value
$\pi = [U, U, \ldots\ldots, U]$	$v_\pi(1)$=?
$\pi(1) = \cdots = \pi(9) = R;$ $\pi(10) = \cdots = \pi(70) = U;$ $\pi(71) = \cdots = \pi(79) = L$ $\pi(80) = \cdots = \pi(100) = U$	$v_\pi(1)$=?
$\pi(1) = \cdots = \pi(9) = R;$ $\pi(10) = \cdots = \pi(90) = U;$ $\pi(91) = \cdots = \pi(100) = L$	$v_\pi(1)$=?
…….	…….



$$S = \{1, \ldots, 100\}$$

$$A = \{Left, Right, Up, Down\}$$

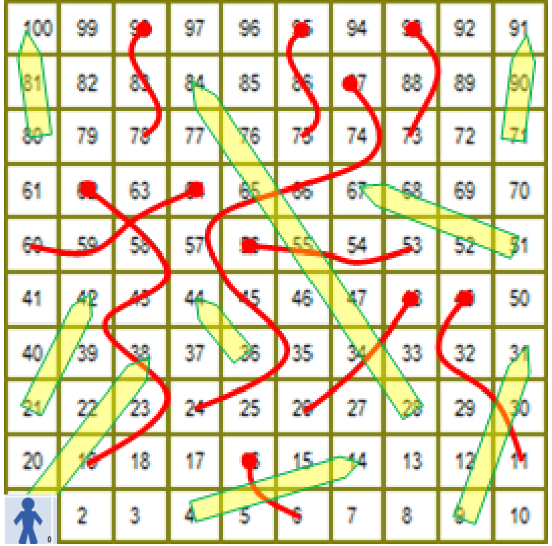
$$r(i, a, j) = \begin{cases} 1000\ is\ j = 100 \\ 0\ otherwise \end{cases}$$

$$\pi^* = arg \max_\pi v_\pi(1)$$

There is a chance that there are multiple optimal policies

Move robot on chutes and ladder: Objective is to reach 100 in shortest distance?

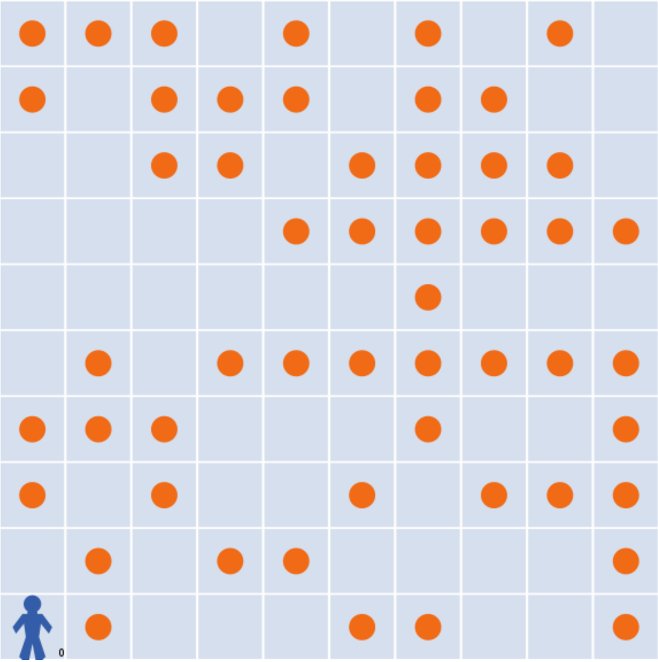
Policy	**Value**
$\pi = [U, U, \dots\dots, U]$	$v_\pi(1)$=?
$\pi(1) = \cdots = \pi(9) = R;$ $\pi(10) = \cdots = \pi(70) = U;$ $\pi(71) = \cdots = \pi(79) = L$ $\pi(80) = \cdots = \pi(100) = U$	$v_\pi(1)$=?
$\pi(1) = \cdots = \pi(9) = R;$ $\pi(10) = \cdots = \pi(90) = U;$ $\pi(91) = \cdots = \pi(100) = L$	$v_\pi(1)$=?
…….	…….



$$S = \{1, \dots, 100\}$$
$$A = \{Left, Right, Up, Down\}$$
$$r(i, a, j) = ?$$
Discounting factor γ =?

Episode length

Robot: pick cans

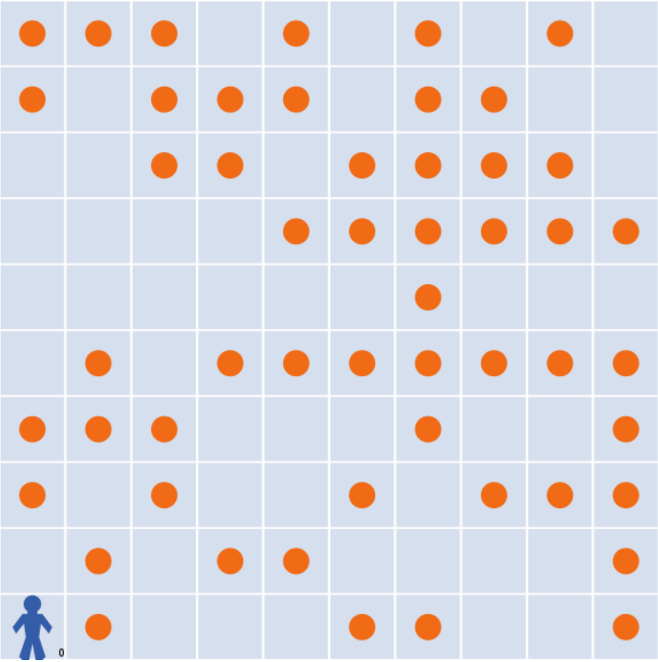


Actions:
1 [move-north]
2 [move-east]
3 [move-south]
4 [move-west]
5 [move-random]
6 [stay-put]
7 [pick-up-can]

State: the can-status in 4 adjacent grids and current grid

Reward:
Wall_penalty: if hits wall
can_reward: if can in current grid and action = 7
Can_penalty: if no can in current grid and action=7

Robot: pick cans



Actions:
1 [move-north]
2 [move-east]
3 [move-south]
4 [move-west]
5 [move-random]
6 [stay-put]
7 [pick-up-can]

State: the can-status in 4 adjacent grids and current grid

Reward:
Wall_penalty: if hits wall
can_reward: if can in current grid and action = 7
Can_penalty: if no can in current grid and action=7

Episode ends when cans have been picked up

Episode length in continuous systems

- $Suppose: S = \{s_1, s_2, s_3\};\ A = \{a, b\}$
- $P_a = \begin{bmatrix} & & \end{bmatrix}; P_b = \begin{bmatrix} & & \end{bmatrix}$
- $R_a = \begin{bmatrix} & & \end{bmatrix}; R_b = \begin{bmatrix} & & \end{bmatrix}$
- And suppose, policy= $\pi = [b, a, a]$

$v_\pi(s_1)$=?

Start in s_1, apply π to transition

Episode	k→	0	1	2	3	4………	T	Total reward
1.	State→	s_1	s_1	s_2	s_3	s_1		
	Action (following π)→	b	b	a	a	b		
	Corresponding immediate reward (R)		$\gamma^0\, r(s_1, b, s_1)$	$\gamma^1 r(s_1, b, s_2)$	$\gamma^2 r(s_2, a, s_3)$	$\gamma^3 r(s_3, a, s_1)$		Sum this row
2.	State→	s_1	s_2	s_3	s_3	s_1		
	Action (following π)→	b	a	a	a	b		
	Corresponding immediate reward (R)		$\gamma^0\, r(s_1, b, s_2)$	$\gamma^1\, r(s_2, a, s_3)$	$\gamma^2\, r(s_3, a, s_3)$	$\gamma^3\, r(s_3, a, s_1)$		Sum this row
……								
								$v_\pi(s_1) = expected\ value\ of\ this\ column$

If system is a continuous system (no terminating state), set episode length as T, and when k=T, reset k=0, reset system to start in s and continue.

Episodic length in terminating systems

- $Suppose: S = \{s_1, s_2, s_3\};\ A = \{a, b\}$
- $P_a = \left[\quad\right]; P_b = \left[\quad\right]$
- $R_a = \left[\quad\right]; R_b = \left[\quad\right]$
- And suppose, policy= $\pi = [b, a, a]$, and we the following :

$v_\pi(s_1)$=?

Start in s_1, apply π to transition

Episode	k→	0	1	2	3	4………	Terminating state	Total reward
1.	State→	s_1	s_1	s_2	s_3	s_1		
	Action (following π)→	b	b	a	a	b		
	Corresponding immediate reward (R)		$\gamma^0\, r(s_1, b, s_1)$	$\gamma^1 r(s_1, b, s_2)$	$\gamma^2 r(s_2, a, s_3)$	$\gamma^3 r(s_3, a, s_1)$		Sum this row
2.	State→	s_1	s_2	s_3	s_3	s_1		
	Action (following π)→	b	a	a	a	b		
	Corresponding immediate reward (R)		$\gamma^0\, r(s_1, b, s_2)$	$\gamma^1\, r(s_2, a, s_3)$	$\gamma^2\, r(s_3, a, s_3)$	$\gamma^3\, r(s_3, a, s_1)$		Sum this row
……								
								$v_\pi(s_1) = expected\ value\ of\ this\ column$

If system is episodic (has terminating state), episode ends when system reaches termination state (so T is not fixed); When terminating state is reached, reset k=0, reset system to start in s and continue.

UMassAmherst

The Commonwealth's Flagship Campus