CSE 573: Artificial Intelligence Winter 2019

Hanna Hajishirzi Markov Decision Processes

slides from Dan Klein, Stuart Russell, Andrew Moore, Dan Weld, Pieter Abbeel, Luke Zettelmoyer

Recap: Defining MDPs

- Markov decision processes:
 - Set of states S
 - Start state s₀
 - Set of actions A
 - Transitions P(s'|s,a) (or T(s,a,s'))
 - Rewards R(s,a,s') (and discount γ)
- MDP quantities so far:
 - Policy = Choice of action for each state
 - Utility = sum of (discounted) rewards



Values of States (Bellman Equations)

- Fundamental operation: compute the (expectimax) value of a state
 - Expected utility under optimal action
 - Average sum of (discounted) rewards
 - This is just what expectimax computed!
- Recursive definition of value:

$$V^{*}(s) = \max_{a} Q^{*}(s, a)$$
$$Q^{*}(s, a) = \sum_{s'} T(s, a, s') \left[R(s, a, s') + \gamma V^{*}(s') \right]$$
$$V^{*}(s) = \max_{a} \sum_{s'} T(s, a, s') \left[R(s, a, s') + \gamma V^{*}(s') \right]$$



Racing Search Tree

- We're doing way too much work with expectimax!
- Problem: States are repeated
 - Idea: Only compute needed quantities once
- Problem: Tree goes on forever
 - Idea: Do a depth-limited computation, but with increasing depths until change is small
 - Note: deep parts of the tree eventually don't matter if γ < 1



Time-Limited Values

- Key idea: time-limited values
- Define V_k(s) to be the optimal value of s if the game ends in k more time steps
 - Equivalently, it's what a depth-k expectimax would give from s



Computing Time-Limited Values



0 0	Gridworl	d Display		
		•		
0.00	0.00	0.00	0.00	
^		^		
0.00		0.00	0.00	
	^		^	
0.00	0.00	0.00	0.00	
VALUES AFTER O TTERATIONS				

0	0	Gridworl	d Display		
	0.00	0.00	0.00 →	1.00	
	0.00		∢ 0.00	-1.00	
		^	^		
	0.00	0.00	0.00	0.00	
				-	
	VALUES AFTER 1 ITERATIONS				

Gridworld Display				
• 0.00	0.00)	0.72)	1.00	
• 0.00		• 0.00	-1.00	
•	• 0.00	•	0.00	
VALUES AFTER 2 ITERATIONS				

Bellman Updates

Example: y=0.9, living reward=0, noise=0.2



Example: Value Iteration



 Information propagates outward from terminal states and eventually all states have correct value estimates

k=3

0	○ ○ Gridworld Display				
	0.00 >	0.52 ▸	0.78 →	1.00	
	• 0.00		• 0.43	-1.00	
	• 0.00	• 0.00	• 0.00	0.00	
	VALUE	S AFTER	3 ITERA	LIONS	

k=4

0 0	Gridworl	d Display		
0.37 ▸	0.66)	0.83)	1.00	
•		• 0.51	-1.00	
•	0.00 →	• 0.31	∢ 0.00	
VALUES AFTER 4 ITERATIONS				

0 0	O O O Gridworld Display					
ſ						
	0.51 →	0.72 >	0.84)	1.00		
			^			
	0.27		0.55	-1.00		
			^			
	0.00	0.22)	0.37	∢ 0.13		
	VALUES AFTER 5 ITERATIONS					

000	0	Gridworl	d Display	
	0.59 →	0.73 →	0.85)	1.00
	• 0.41		• 0.57	-1.00
	• 0.21	0.31 →	• 0.43	∢ 0.19
	VALUE	S AFTER	6 ITERA	FIONS

00	Gridworl	d Display	
0.62 →	0.74 ▸	0.85)	1.00
• 0.50		• 0.57	-1.00
• 0.34	0.36)	• 0.45	∢ 0.24
VALUES AFTER 7 ITERATIONS			

0 0	Gridworl	d Display		
0.63)	0.74)	0.85)	1.00	
• 0.53		• 0.57	-1.00	
• 0.42	0.39 →	• 0.46	∢ 0.26	
VALUES AFTER 8 ITERATIONS				

00	Gridworld Display				
	0.64 →	0.74 ▸	0.85)	1.00	
	• 0.55		• 0.57	-1.00	
	• 0.46	0.40 →	• 0.47	∢ 0.27	
	VALUE	S AFTER	9 ITERA	FIONS	

0 0	Gridworl	d Display	
0.64 →	0.74 ▸	0.85)	1.00
▲ 0.56		• 0.57	-1.00
• 0.48	∢ 0.41	• 0.47	◀ 0.27
VALUES AFTER 10 ITERATIONS			

0 0	0	Gridworl	d Display	-	
ſ					
	0.64 →	0.74 →	0.85 →	1.00	
	• 0.56		• 0.57	-1.00	
	0.48	◀ 0.42	0.47	◀ 0.27	
	VALUES AFTER 11 ITERATIONS				

000		Gridworld Display			
	0.64)	0.74)	0.85)	1.00	
	• 0.57		• 0.57	-1.00	
	0.49	∢ 0.42	• 0.47	∢ 0.28	
VALUES AFTER 12 ITERATIONS					

Gridworld Display					
0.64 ▸	0.74)	0.85)	1.00		
• 0.57		▲ 0.57	-1.00		
▲ 0.49	∢ 0.43	▲ 0.48	∢ 0.28		
VALUES AFTER 100 ITERATIONS					