RLearning: Short guides to reinforcement learning

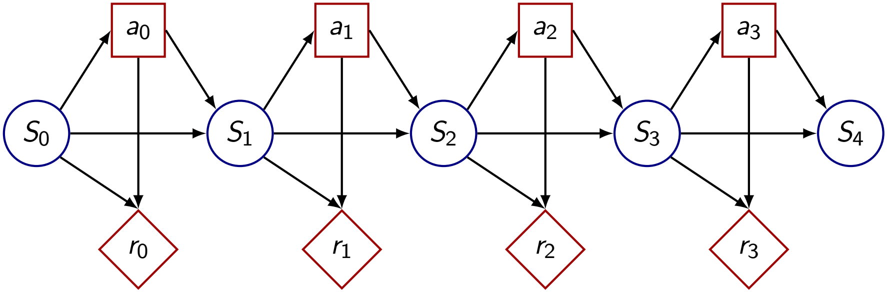
Unit 2-3: Intro to Value Iteration with Bellman Equation

Davud Rostam-Afschar (Uni Mannheim)

Get the best out of now +
what you expect to be best

Value Iteration

- Performs dynamic programming
- Optimizes decisions in reverse order



Value Iteration (?)

- Value when no time left:
  $V(s_h) = \max_{a_h} R(s_h, a_h)$

Value Iteration (?)

- Value when no time left:
  $V(s_h) = \max_{a_h} R(s_h, a_h)$
- Value with one time step left:
  $V(s_{h-1}) = \max_{a_{h-1}} R(s_{h-1}, a_{h-1}) + \gamma \sum_{s_h} \mathbb{P}(s_h \mid s_{h-1}, a_{h-1}) V(s_h)$

Value Iteration (?)

- Value when no time left:
  $V(s_h) = \max_{a_h} R(s_h, a_h)$
- Value with one time step left:
  $V(s_{h-1}) = \max_{a_{h-1}} R(s_{h-1}, a_{h-1}) + \gamma \sum_{s_h} \mathbb{P}(s_h \mid s_{h-1}, a_{h-1}) V(s_h)$
- Value with two time steps left:
  $V(s_{h-2}) = \max_{a_{h-2}} R(s_{h-2}, a_{h-2}) + \gamma \sum_{s_{h-1}} \mathbb{P}(s_h \mid s_{h-2}, a_{h-2}) V(s_{h-1})$

Value Iteration (?)

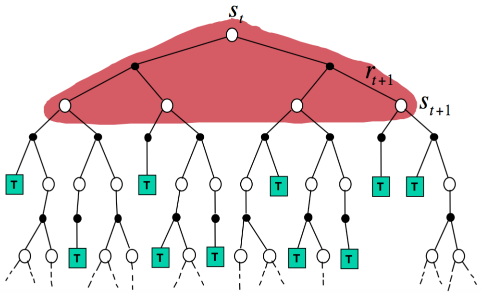
- Value when no time left:
  $V(s_h) = \max_{a_h} R(s_h, a_h)$
- Value with one time step left:
  $V(s_{h-1}) = \max_{a_{h-1}} R(s_{h-1}, a_{h-1}) + \gamma \sum_{s_h} \mathbb{P}(s_h \mid s_{h-1}, a_{h-1}) V(s_h)$
- Value with two time steps left:
  $V(s_{h-2}) = \max_{a_{h-2}} R(s_{h-2}, a_{h-2}) + \gamma \sum_{s_{h-1}} \mathbb{P}(s_h \mid s_{h-2}, a_{h-2}) V(s_{h-1})$
- Bellman's equation:

$$V(s_t) = \max_{a_t} R(s_t, a_t) + \gamma \sum_{s_{t+1}} \mathbb{P}(s_{t+1} \mid s_t, a_t) V(s_{t+1})$$

$$a_t^* = \underset{a_t}{\operatorname{argmax}} R(s_t, a_t) + \gamma \sum_{s_{t+1}} \mathbb{P}(s_{t+1} \mid s_t, a_t) V(s_{t+1})$$

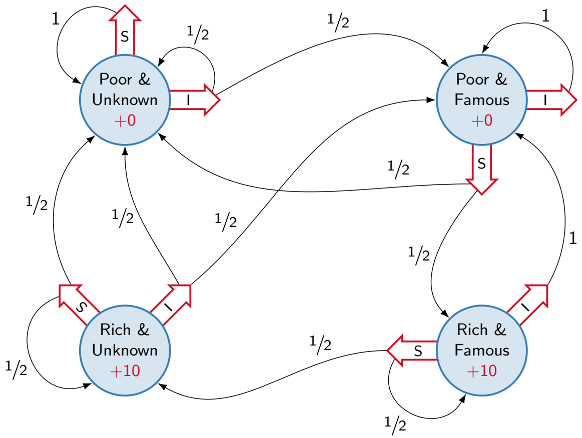
Dynamic Programming Backup

$$V(S_t) \leftarrow \mathbb{E}_\pi \left[R_{t+1} + \gamma V(S_{t+1})\right]$$



Example: Invest or Save?

A Markov Decision Process

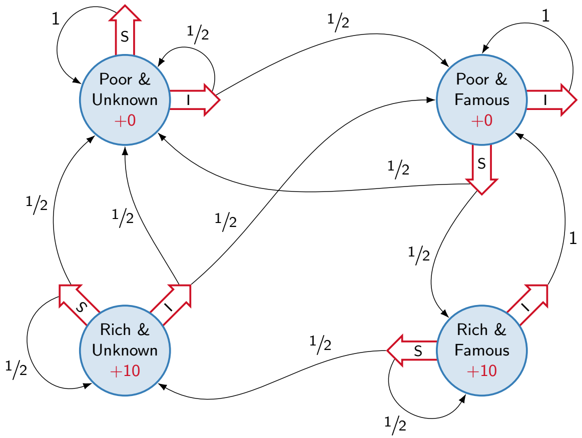


You own a company

In every state you must choose between **I**nvesting or **S**aving.

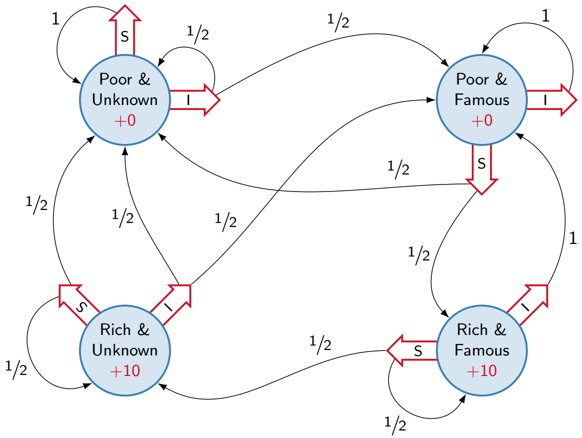
$\gamma = 0.9$

Transition Model for Invest



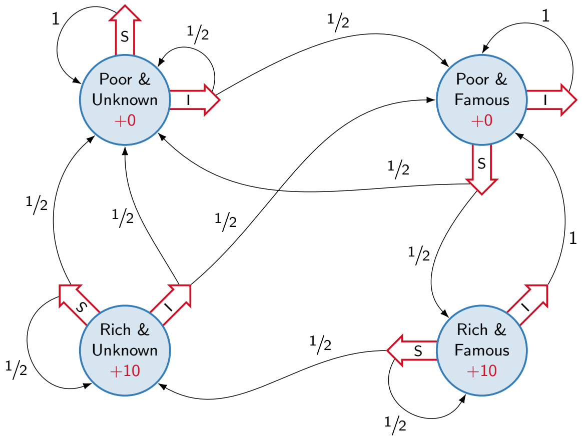
Transition Probability Function for Action Invest					
S	S'				Marginal Prob
	Poor & Unknown	Poor & Famous	Rich & Unknown	Rich & Famous	
Poor & Unknown	0.50	0.50	0.00	0.00	1.00
Poor & Famous	0.00	1.00	0.00	0.00	1.00
Rich & Unknown	0.50	0.50	0.00	0.00	1.00
Rich & Famous	0.00	1.00	0.00	0.00	1.00

Reward Model for Invest



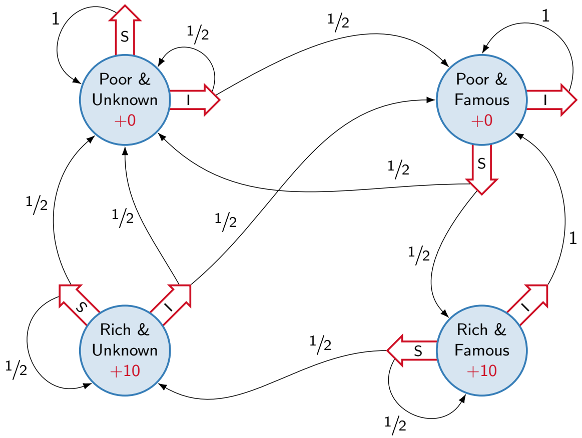
Reward Function for Action Invest				
S	S'			
	Poor & Unknown	Poor & Famous	Rich & Unknown	Rich & Famous
Poor & Unknown	0	0	0	0
Poor & Famous	0	0	0	0
Rich & Unknown	10	10	10	10
Rich & Famous	10	10	10	10

Transition Model for Save



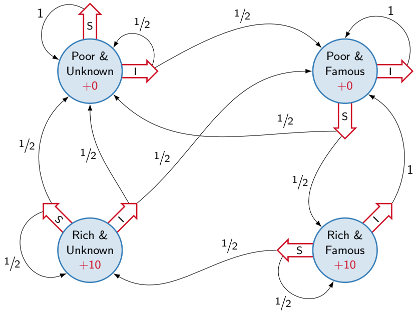
Transition Probability Function for Action Save					
S	S'				Marginal Prob
	Poor & Unknown	Poor & Famous	Rich & Unknown	Rich & Famous	
Poor & Unknown	1.00	0.00	0.00	0.00	1.00
Poor & Famous	0.50	0.00	0.00	0.50	1.00
Rich & Unknown	0.50	0.00	0.50	0.00	1.00
Rich & Famous	0.00	0.00	0.50	0.50	1.00

Reward Model for Save



Reward Function for Action Save				
S	S'			
	Poor & Unknown	Poor & Famous	Rich & Unknown	Rich & Famous
Poor & Unknown	0	0	0	0
Poor & Famous	0	0	0	0
Rich & Unknown	10	10	10	10
Rich & Famous	10	10	10	10

Values and Policies for Each State



$$V_h(RF) = \max_{\text{action}} \{R(RF, I), R(RF, S)\} = \max_a \{10, 10\} = 10$$

$$\pi_h(RF) = \text{argmax}\{R(RF, I), R(RF, S)\} = \{I, S\}$$

$$V_{h-1}(RF) = \max_{\text{action}} R(RF, \text{ action }) + \gamma \sum_{\text{state}_h} \mathbb{P}\left(s_h \mid RF, a_{h-1}\right) V\left(s_h\right) =$$

$$= \max_{\text{action}} \{10 + 0.9(1 * 0), 10 + 0.9(0.5 * 10 + 0.5 * 10)\} = \max_a \{10, 19\} = 19$$

$$\pi_{h-1}(RF) = \{S\}$$

Values and Policies for Each State

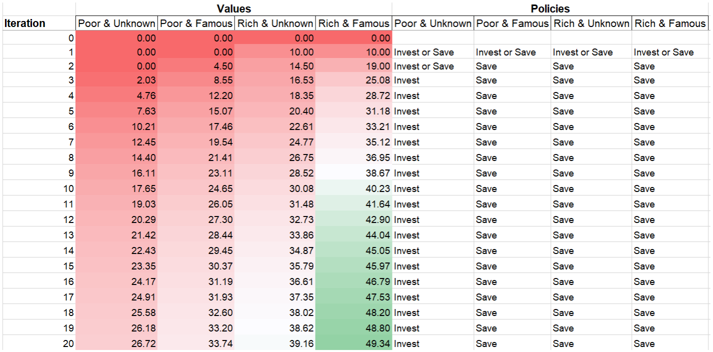
$$V_h(RF) = \max_{\text{action}} \{R(RF, I), R(RF, S)\} = \max_{a}\{10, 10\} = 10$$

$$\pi_h(RF) = \text{argmax}\{R(RF, I), R(RF, S)\} = \{I, S\}$$

$$V_{h-1}(RF) = \max_{\text{action}} R(RF, \text{ action }) + \gamma \sum_{\text{state }h} \mathbb{P}(s_h \mid RF, a_{h-1})\, V(s_h) =$$

$$= \max_{\text{action}} \{10 + 0.9(1 * 0), 10 + 0.9(0.5 * 10 + 0.5 * 10)\} = \max_{a}\{10, 19\} = 19$$

$$\pi_{h-1}(RF) = \{S\}$$



	Values				Policies			
Iteration	Poor & Unknown	Poor & Famous	Rich & Unknown	Rich & Famous	Poor & Unknown	Poor & Famous	Rich & Unknown	Rich & Famous
0	0.00	0.00	0.00	0.00				
1	0.00	0.00	10.00	10.00	Invest or Save	Invest or Save	Invest or Save	Invest or Save
2	0.00	4.50	14.50	19.00	Invest or Save	Save	Save	Save
3	2.03	8.55	16.53	25.08	Invest	Save	Save	Save
4	4.76	12.20	18.35	28.72	Invest	Save	Save	Save
5	7.63	15.07	20.40	31.18	Invest	Save	Save	Save
6	10.21	17.46	22.61	33.21	Invest	Save	Save	Save
7	12.45	19.54	24.77	35.12	Invest	Save	Save	Save
8	14.40	21.41	26.75	36.95	Invest	Save	Save	Save
9	16.11	23.11	28.52	38.67	Invest	Save	Save	Save
10	17.65	24.65	30.08	40.23	Invest	Save	Save	Save
11	19.03	26.05	31.48	41.64	Invest	Save	Save	Save
12	20.29	27.30	32.73	42.90	Invest	Save	Save	Save
13	21.42	28.44	33.86	44.04	Invest	Save	Save	Save
14	22.43	29.45	34.87	45.05	Invest	Save	Save	Save
15	23.35	30.37	35.79	45.97	Invest	Save	Save	Save
16	24.17	31.19	36.61	46.79	Invest	Save	Save	Save
17	24.91	31.93	37.35	47.53	Invest	Save	Save	Save
18	25.58	32.60	38.02	48.20	Invest	Save	Save	Save
19	26.18	33.20	38.62	48.80	Invest	Save	Save	Save
20	26.72	33.74	39.16	49.34	Invest	Save	Save	Save

Value Iteration Converges

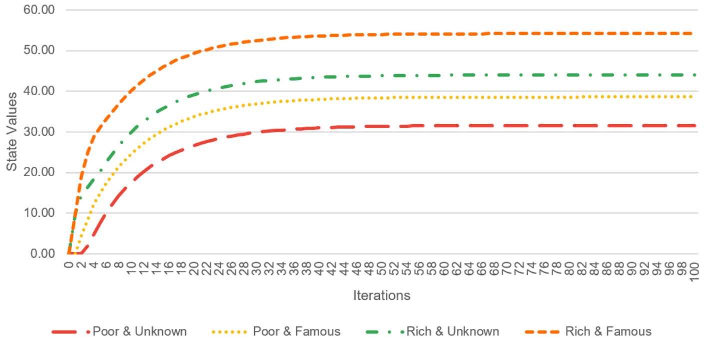
$$V_h(RF) = \max_{\text{action}} \{R(RF, I), R(RF, S)\} = \max_a \{10, 10\} = 10$$

$$\pi_h(RF) = \text{argmax}\{R(RF, I), R(RF, S)\} = \{I, S\}$$

$$V_{h-1}(RF) = \max_{\text{action}} R(RF, \text{ action }) + \gamma \sum_{\text{state } h} \mathbb{P}(s_h \mid RF, a_{h-1}) V(s_h) =$$

$$= \max_{\text{action}} \{10 + 0.9(1 * 0), 10 + 0.9(0.5 * 10 + 0.5 * 10)\} = \max_a \{10, 19\} = 19$$

$$\pi_{h-1}(RF) = \{S\}$$



Endgame Effects

Finite Horizon

- When h is finite,
- Non-stationary optimal policy
- Best action different at each time step
- Intuition: best action varies with the amount of time left

Infinite Horizon

- When h is infinite,
- Stationary optimal policy
- Same best action at each time step
- Intuition: same (infinite) amount of time left at each time step, hence same best action
- Problem: value iteration does an infinite number of iterations...

Infinite Horizon

- Problem: value iteration does an infinite number of iterations...
- Assuming a discount factor γ, after n time steps, rewards are scaled down by γ^n
- For large enough n, rewards become insignificant since $\gamma^n \rightarrow 0$
- Solution:
  - pick large enough n
  - run value iteration for n steps
  - Execute policy found at the n^{th} iteration

References I

BELLMAN, R. (1957): *Dynamic Programming*. Princeton University Press.

Takeaways

How to Get The Best Now And in The Future?

- Bellman equation relates immediate rewards to future values
- Value iteration solves for optimal policies by dynamic programming
- Finite horizon problems lead to non-stationary policies
- Infinite horizon problems yield stationary policies, stabilized with discounting