RLearning:
Short guides to reinforcement learning

Unit 3-4: Q-Learning

Davud Rostam-Afschar (Uni Mannheim)

At this state how much good stuff will happen ... if I do THIS?

Reinforcement learning agents may or may not include the following components:

- **Model:** $\mathbb{P}\left(s' \mid s, a\right), \mathbb{P}(r \mid s, a)$
  - Environment dynamics and rewards
- **Policy:** $\pi(s)$
  - Agent action choices
- **Value function:** $V(s)$
  - Expected total rewards of the agent's policy

Important Components in Reinforcement Learning

Reinforcement learning agents may or may not include the following components:

- **Model:** $\mathbb{P}(s' \mid s, a), \mathbb{P}(r \mid s, a)$
  - Environment dynamics and rewards
- **Policy:** $\pi(s)$
  - Agent action choices
- **Value function:** $V(s)$
  - Expected total rewards of the agent's policy
- **Quality function:** $Q(s, a)$
  - Expected total rewards of taking a specific action in a given state and then following a particular policy thereafter

Bellman's Equation

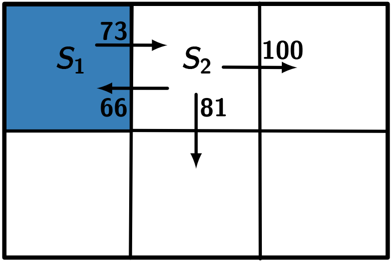
- Optimal state value function $V^*(s)$

$$V^*(s) = \max_a \mathbb{E}[r \mid s, a] + \gamma \sum_{s'} \mathbb{P}\left(s' \mid s, a\right) V^*\left(s'\right)$$

- Optimal state-action value function $Q^*(s, a)$

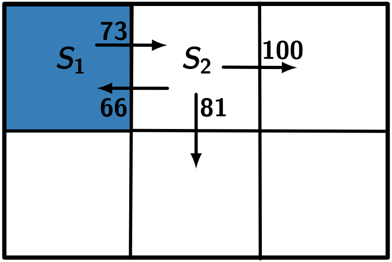
$$Q^*(s, a) = \mathbb{E}[r \mid s, a] + \gamma \sum_{s'} \mathbb{P}\left(s' \mid s, a\right) \max_{a'} Q^*\left(s', a'\right)$$

where $V^*(s) = \max_a Q^*(s, a)$
$\pi^*(s) = \operatorname*{argmax}_a Q^*(s, a)$



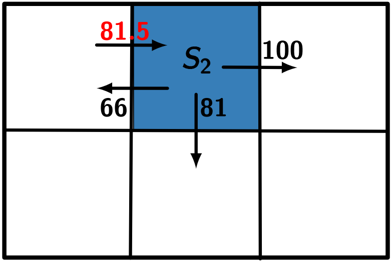
$\gamma = 0.9, \alpha = 0.5, r = 0$ for non-terminal states

Temporal Difference



$\gamma = 0.9, \alpha = 0.5, r = 0$ for non-terminal states

$$\begin{aligned} Q(s_1, right) &= Q(s_1, right) + \alpha\left(r + \gamma \max_{a'} Q(s_2, a') - Q(s_1, right)\right) \\ &= 73 + 0.5(0 + 0.9 \max\{66, 81, 100\} - 73) \\ &= 73 + 0.5(17) \\ &= 81.5 \end{aligned}$$



$\gamma = 0.9, \alpha = 0.5, r = 0$ for non-terminal states

$$\begin{aligned}
Q(s_1, right) &= Q(s_1, right) + \alpha\left(r + \gamma \max_{a'} Q(s_2, a') - Q(s_1, right)\right) \\
&= 73 + 0.5(0 + 0.9 \max\{66, 81, 100\} - 73) \\
&= 73 + 0.5(17) \\
&= 81.5
\end{aligned}$$

Q-Learning

Qlearning (s, Q^*)
  Repeat
    Select and execute a
    Observe s' and r
    Update counts: $n(s,a) \leftarrow n(s,a)+1$
    Learning rate: $\alpha \leftarrow 1/n(s,a)$
    Update Q-value:
    $Q^*(s,a) \leftarrow Q^*(s,a) + \alpha\left(r + \gamma \max_{a'} Q^*(s',a') - Q^*(s,a)\right)$
    $s \leftarrow s'$
  Until convergence of Q^*
Return Q^*

- Sample based variant of value iteration
- Model free
- Temporal difference update

Exploration vs Exploitation

- If an agent always chooses the action with the highest value then it is exploiting
  - The learned model is not the real model
  - Leads to suboptimal results
- By taking random actions (pure exploration) an agent may learn the model
  - But what is the use of learning a complete model if parts of it are never used?
- Need a balance between exploitation and exploration

Common Exploration Methods

- ε-greedy:
  - With probability ε execute random action
  - Otherwise execute best action a^*

$$a^* = \text{argmax}_a \, Q(s, a)$$

- Boltzmann exploration

$$\mathbb{P}(a) = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_a e^{\frac{Q(s,a)}{\tau}}}$$

- τ: temperature parameter
  - High τ: more random (exploration)
  - Low τ: closer to greedy (exploitation)

Exploration and Q-learning

- Q-learning converges to optimal Q-values if
  - Every state is visited infinitely often (due to exploration)
  - The action selection becomes greedy as time approaches infinity
  - The probability of exploration ε is decreased fast enough, but not too fast (sufficient conditions for ε):

$$\sum_n \varepsilon_n \to \infty \tag{1}$$

$$\sum_n \varepsilon_n^2 < \infty \tag{2}$$

Summary

- We can optimize a policy by RL when the transition and reward functions are unknown
- Model free, value based agent:
  - Monte Carlo learning (unbiased, but lots of data)
  - Temporal difference learning (low variance, less data)
- Active learning:
  - Exploration/exploitation dilemma

Q-Learning in Practice

Toy Maze Example

3	**r**	**r**	**r**	$+1$
2	**u**		**u**	-1
1	**u**	**l**	**l**	**l**
	1	2	3	4

Start state: (1,1)
Terminal states: (4,2), (4,3)
No discount: $\gamma = 1$

Reward is -0.04 for non-terminal states

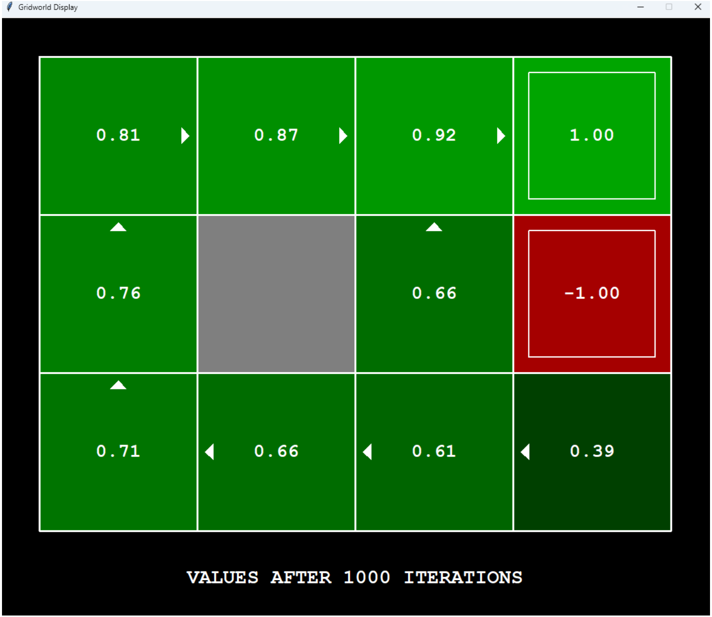
Four actions:

- up (**u**),
- left (**l**),
- right (**r**),
- down (**d**)

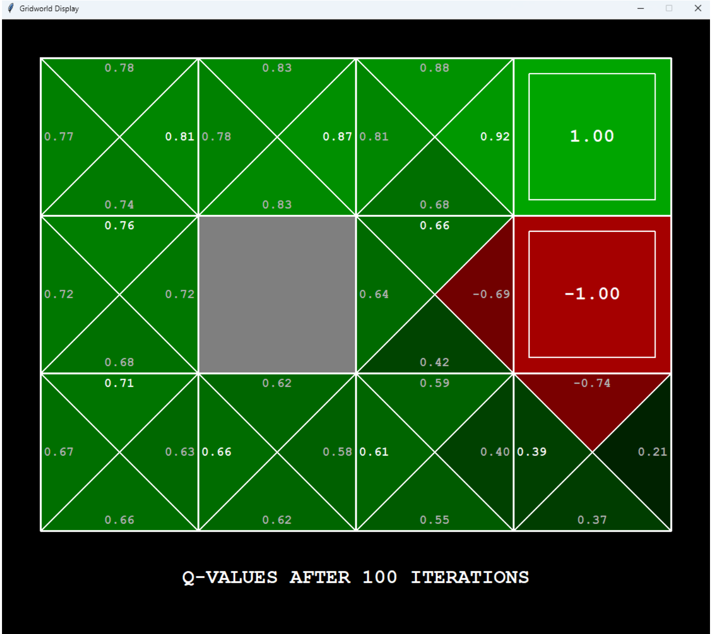
Do not know the transition probabilities

What is the value $V(s)$ of being in state s

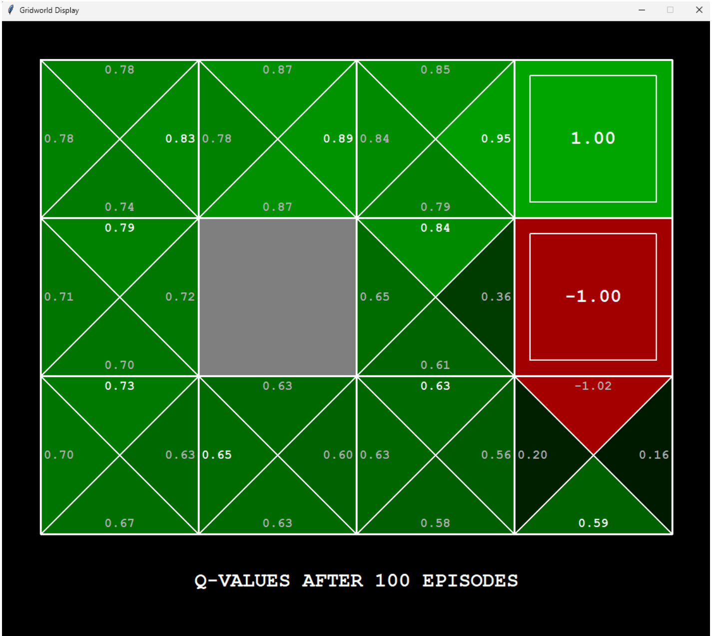
Toy Maze Example (No Learning, Noise 20%)



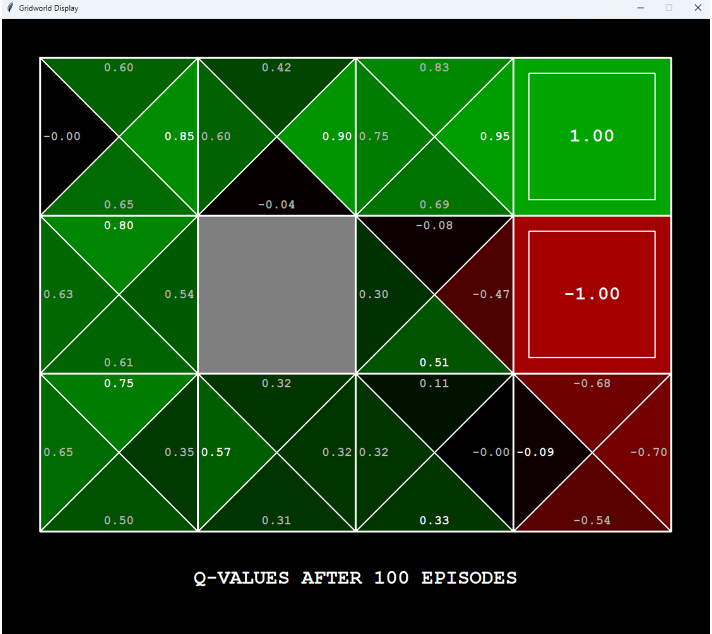
Toy Maze Example (No Learning, Noise 20%)



Toy Maze Example ($\varepsilon = 0.9$, Noise 20%)



Toy Maze Example ($\varepsilon = 0.1$, Noise 20%)



References I

DeNero, J., D. Klein, B. Miller, N. Hay, and P. Abbeel (2013): "The Pacman AI Projects," http://inst.eecs.berkeley.edu/~cs188/pacman/, Developed at UC Berkeley. Core by John DeNero and Dan Klein; student autograding by Brad Miller, Nick Hay, and Pieter Abbeel.

Poupart, P. (2025): "Pascal Poupart's Homepage," https://cs.uwaterloo.ca/~ppoupart/, Accessed: 2025-05-24.

Russell, S. J., and P. Norvig (2016): *Artificial intelligence: a modern approach*. Pearson.

Sigaud, O., and O. Buffet (2013): *Markov decision processes in artificial intelligence*. John Wiley & Sons.

Sutton, R. S., and A. G. Barto (2018): "Reinforcement learning: An introduction," *A Bradford Book*, Available at http://incompleteideas.net/book/the-book-2nd.html.

Szepesvári, C. (2022): *Algorithms for reinforcement learning*. Springer nature, Available at https://sites.ualberta.ca/~szepesva/RLBook.html.

Takeaways

Learn Value of Taking Actions in Specific States

- Q-Learning learns optimal actions without knowing the model
- Balancing exploration and exploitation is crucial
- ϵ-greedy and Boltzmann are common exploration methods
- Sufficient exploration guarantees convergence