CS 4700: Foundations of Artificial Intelligence

Spring 2019

Prof. Haym Hirsh

Lecture 15

February 27, 2019

Announcements

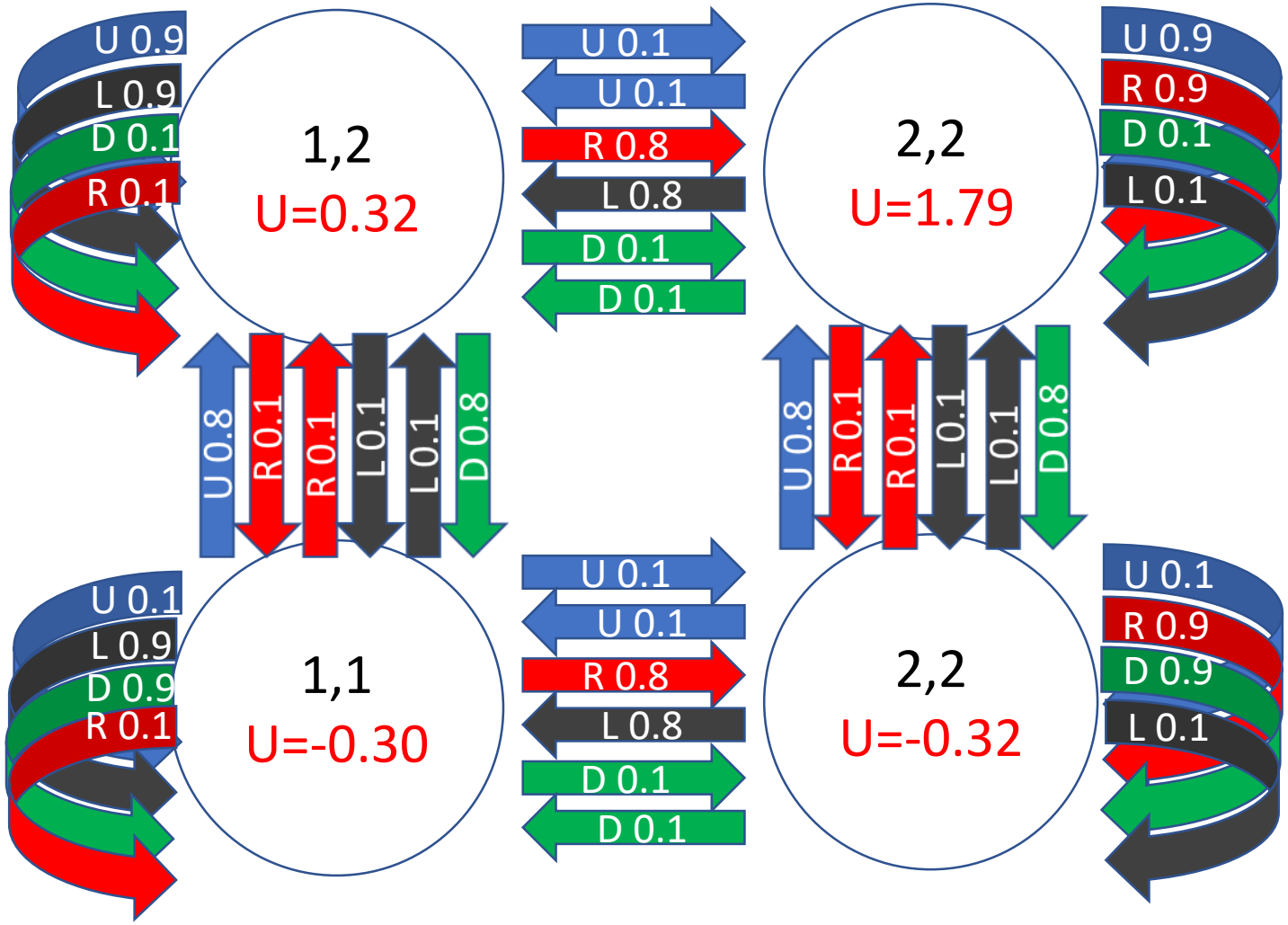
- Karma lectures:
  - Tomorrow: 4:15, Gates G01:
    “Towards Generalization and Efficiency in Reinforcement Learning”
  - Friday: 12:15, Gates 122:
    “Teaching Machines Like We Teach People”
- Textbook:
  - Chapter numbers and chapter filenames don’t always match up
  - Example: Chapter 20 is in Newchap21.pdf
  - When I refer to the textbook I’ll be using Chapter NUMBERS
  - Example: Reinforcement learning is Chapter 20
- Homework 2 grades released
  - Regrade requests due March 5, 12 noon
  - Be sensible

Announcements

- Other upcoming events:
  - Tomorrow: 5:30-8:00pm G01:
    Cornell CS Research Night
  - This weekend:
    Microsoft Azure Hackathon – anyone know more?

Policy Π_{UR} = Go up if you're in row 1, otherwise go right

γ =0.5



But what about Π*?

But what about Π*?

Use Policy Iteration

But what about Π*?

Use Policy Iteration

(among various approaches)

Finding Π^*: Policy Iteration

Policy_Iteration(S,A,P,R,γ):

For all s$\in$S { $\hat{\pi}_0(s) \leftarrow$ <random a$\in$A>; $\hat{U}_0(s) \leftarrow 0$; i $\leftarrow$ 0 }

Repeat

i $\leftarrow$ i+1

For all s$\in$S

$$\hat{U}_i(s) \leftarrow R(s) + \gamma \sum_{s' \in S} [P(s' | s, \hat{\pi}_{i-1}(s))\, \hat{U}_{i-1}(s')]$$

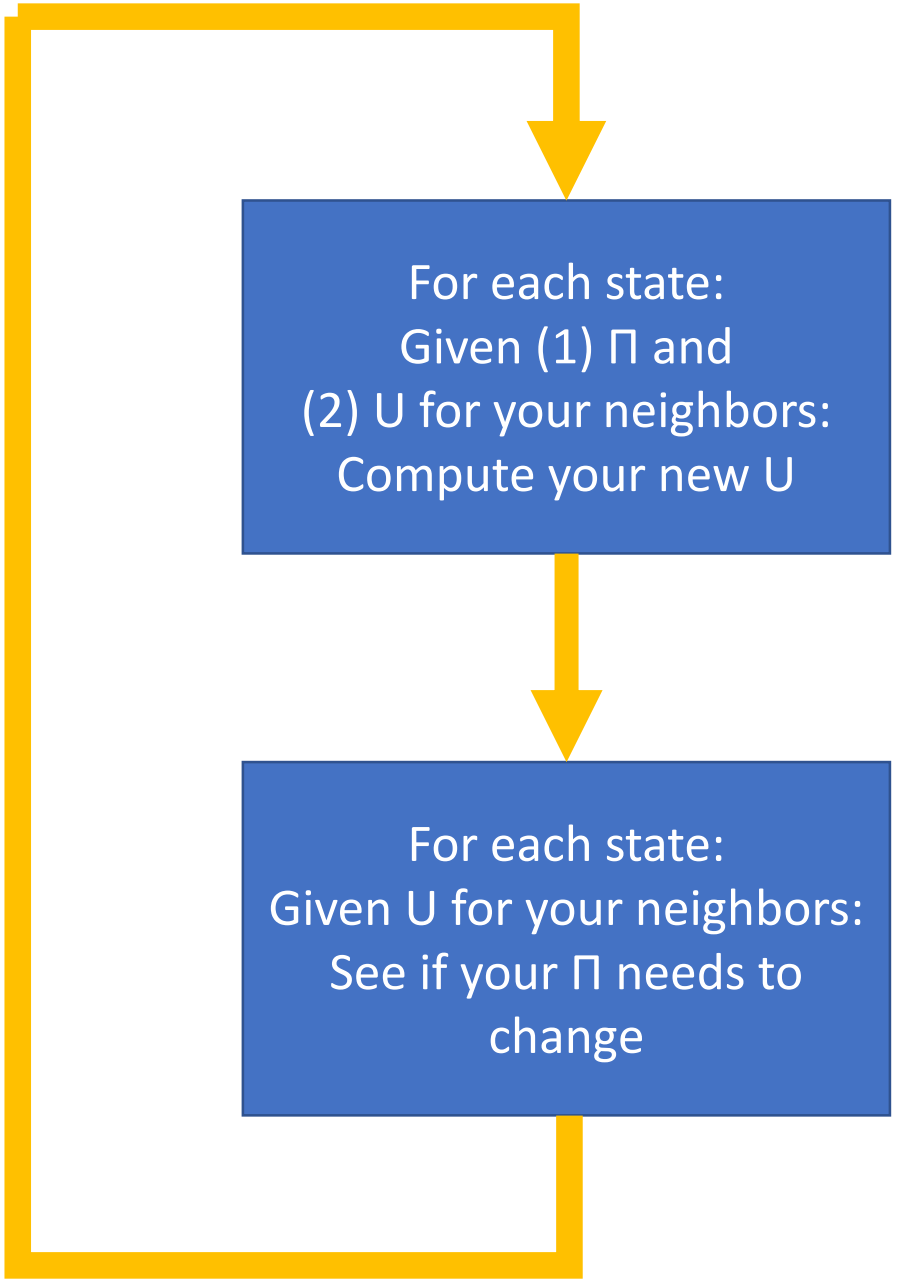
For all s$\in$S

If $\max_{a \in A} \sum_{s' \in S} [P(s' | s, a)\, \hat{U}_i(s')] > \sum_{s' \in S} [P(s' | s, \hat{\pi}_{i-1}(s))\, \hat{U}_i(s')]$

Then $\hat{\pi}_i(s) \leftarrow \operatorname{argmax}_{a \in A} \sum_{s' \in S} [P(s' | s, a)\, \hat{U}_i(s')]$

Else $\hat{\pi}_i(s) \leftarrow \hat{\pi}_{i-1}(s)$

Until <stopping condition> /* for example: $\hat{\pi}$ doesn't change */

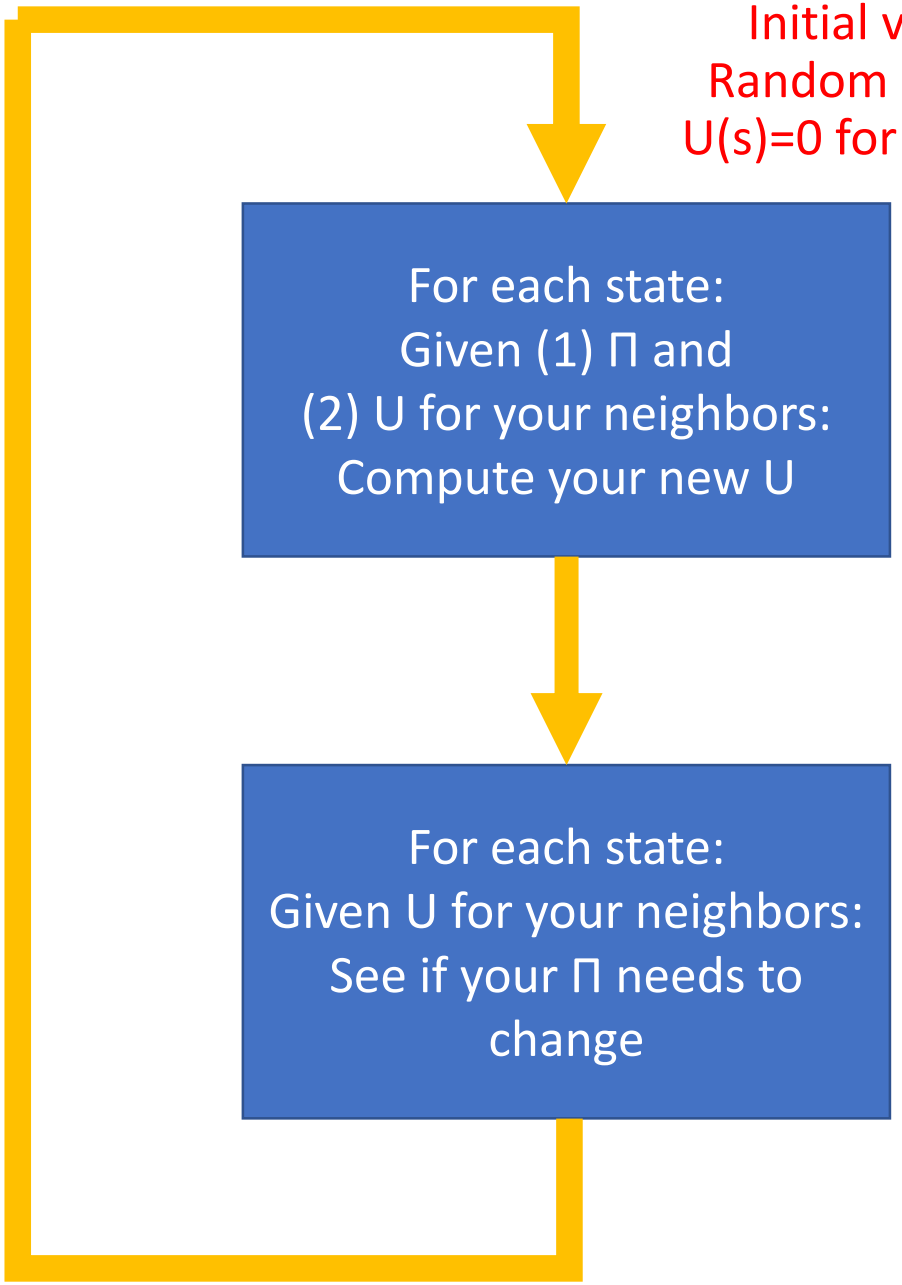


Use:

$$U^*(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s,\Pi*(s))U*(s')$$

Use:

$$\Pi^*(s) = \text{argmax}_{a \in A} \sum_{s' \in S} P(s'|s,a)U*(s')$$



Use:

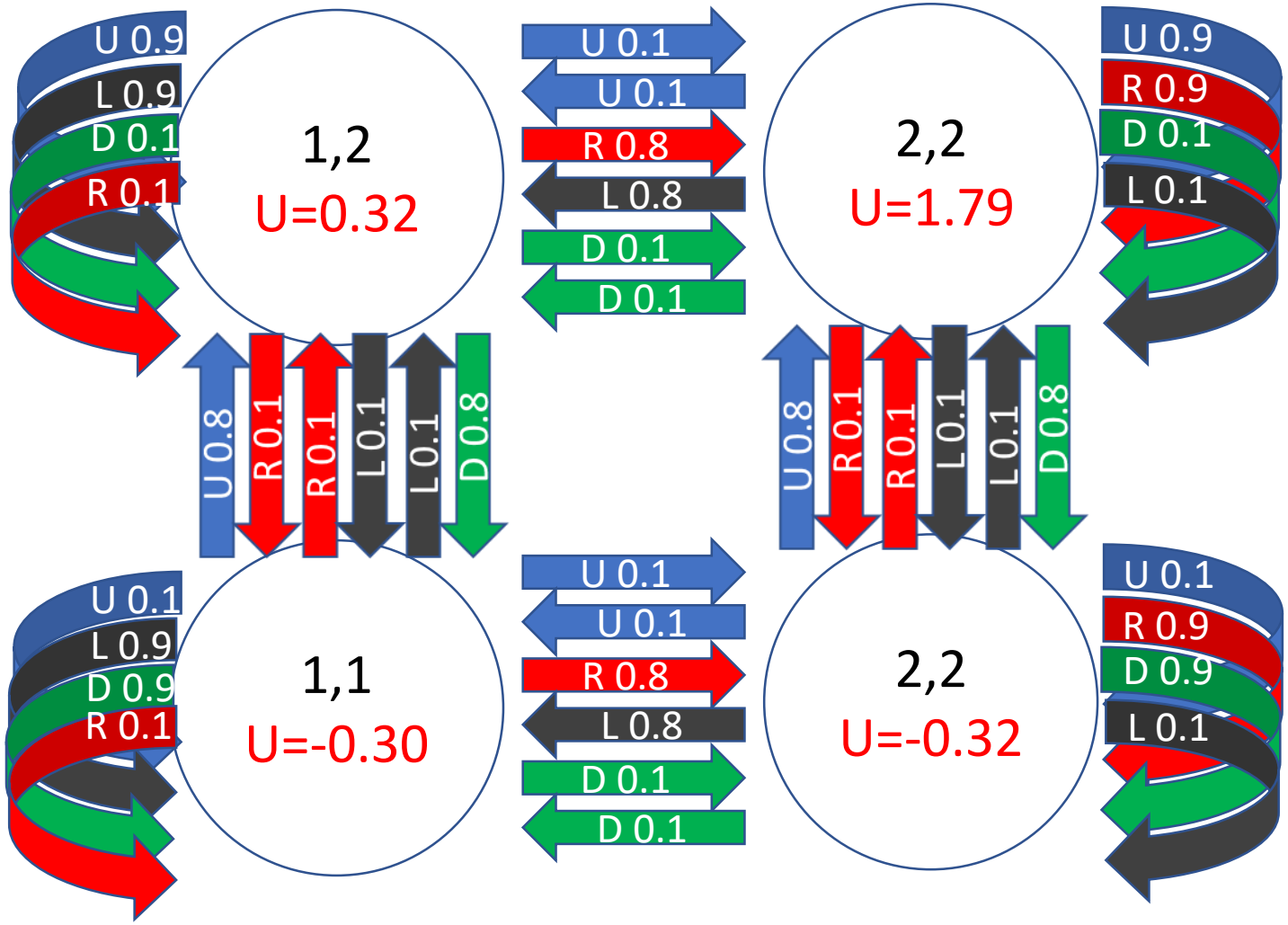
$$U^*(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s,\Pi*(s))U*(s')$$

Use:

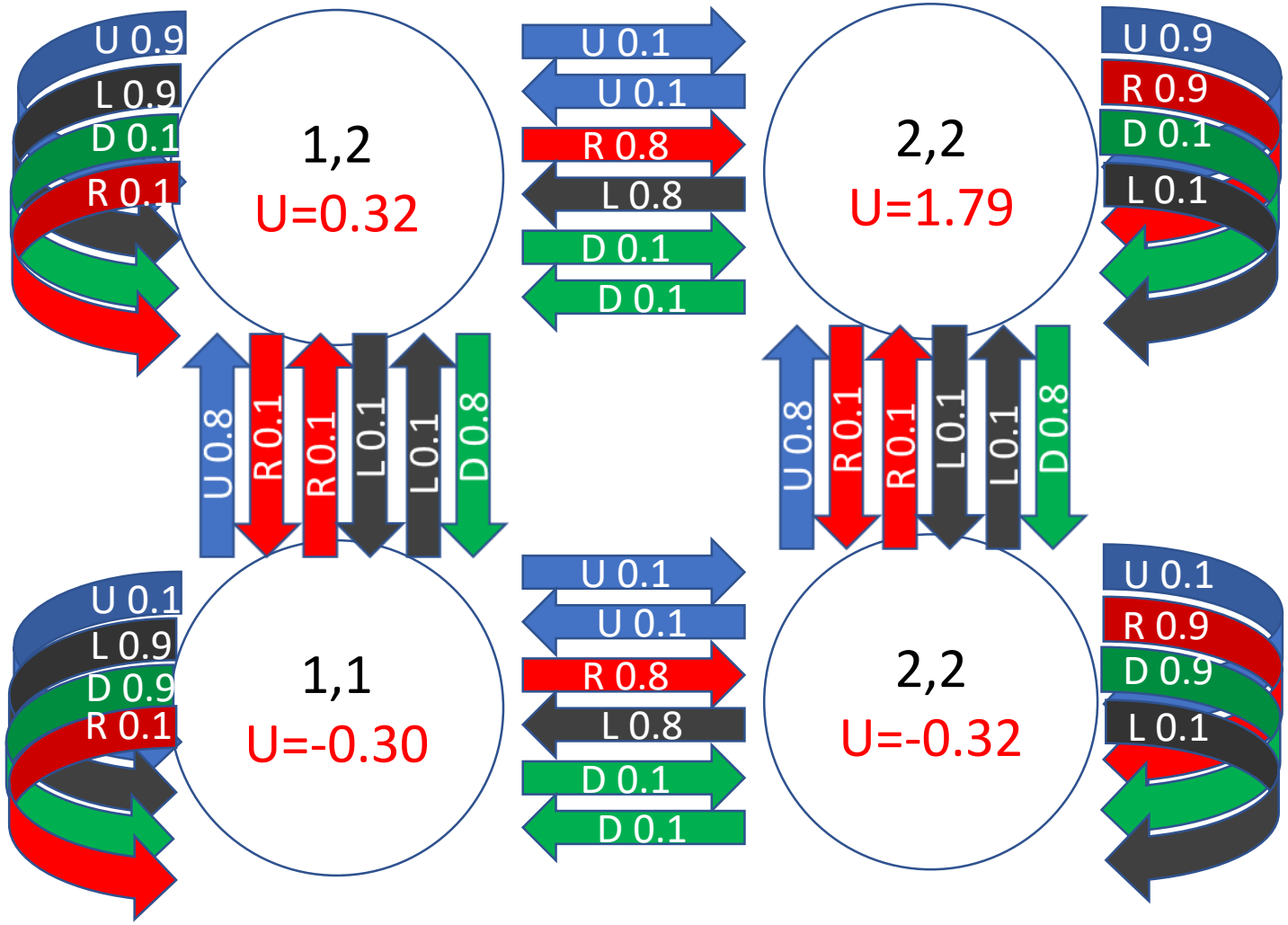
$$\Pi^*(s) = \text{argmax}_{a \in A} \sum_{s' \in S} P(s'|s,a)U*(s')$$

Policy Π_{UR} = Go up if you're in row 1, otherwise go right

γ =0.5



Π*?


U 0.9
L 0.9
D 0.1
R 0.1
1,2
U=0.32
U 0.1
U 0.1
R 0.8
L 0.8
D 0.1
D 0.1
2,2
U=1.79
U 0.9
R 0.9
D 0.1
L 0.1
U 0.8
R 0.1
R 0.1
L 0.1
L 0.1
D 0.8
U 0.8
R 0.1
R 0.1
L 0.1
L 0.1
D 0.8
U 0.1
L 0.9
D 0.9
R 0.1
1,1
U=-0.30
U 0.1
U 0.1
R 0.8
L 0.8
D 0.1
D 0.1
2,2
U=-0.32
U 0.1
R 0.9
D 0.9
L 0.1

Example

https://bit.ly/2T6lYWD

(You will need to copy it to your own spreadsheet to play with it)

What if we don’t know P and R?

What if we don’t know P and R?

Are we

learning while moving about in the world
(active reinforcement learning)?

or

learning from historical traces
(passive reinforcement learning)?

What if we don’t know P and R?

Are we

learning while moving about in the world
(active reinforcement learning)?

or

learning from historical traces
(passive reinforcement learning)?

Active Reinforcement Learning

Key concept:

You are getting (possibly bad) rewards while exploring

How do you balance learning about the world vs taking a hit for doing so?
When do you stop exploring because you've learned enough?

Active Reinforcement Learning

Key concept:

You are getting (possibly bad) rewards while exploring

How do you balance learning about the world vs taking a hit for doing so?
When do you stop exploring because you've learned enough?

Exploration vs Exploitation

Active Reinforcement Learning with Q Learning

Key idea: Q: S x A → $\mathbb{R}$

Q(s,a): Expected value of taking action a and acting optimally thereafter

Active Reinforcement Learning with Q Learning

Key idea: Q: S x A → $\mathbb{R}$

Q(s,a): Expected value of taking action a and acting optimally thereafter

$$U^*(s) = \max_a Q(s, a)$$

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

Active Reinforcement Learning with Q Learning

Key idea: Q: S x A → $\mathbb{R}$

Q(s,a): Expected value of taking action a and acting optimally thereafter

$$U^*(s) = \max_a Q(s, a)$$

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

If I have Q I know how to act (even without having R and P)

Active Reinforcement Learning with Q Learning

Key idea: Q: S x A → $\mathbb{R}$

Q(s,a): Expected value of taking action a and acting optimally thereafter

$$U^*(s) = \max_a Q(s, a)$$

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

If I have Q I know how to act (even without having R and P)

We're going to learn Q

Q Learning

How do we learn Q?

Q Learning

How do we learn Q?

Want something analogous to this (from Policy Iteration):

$$\hat{\mathrm{U}}_i(\mathrm{s}) \leftarrow \mathrm{R(s)} + \boldsymbol{\gamma} \sum_{s' \in \mathrm{S}} [P(s' | \mathrm{s}, \hat{\pi}_{i-1}(\mathrm{s}))\ \hat{\mathrm{U}}_{i-1}(s')]$$

$$\hat{\pi}_i(\mathrm{s}) \leftarrow \underset{\mathrm{a} \in \mathrm{A}}{\mathrm{argmax}} \sum_{s' \in \mathrm{S}} [P(\mathrm{s}' | \mathrm{s}, \mathrm{a})\ \hat{\mathrm{U}}_i(\mathrm{s}')]$$

Q Learning

How do we learn Q?

Want something analogous to this (from Policy Iteration):

$$\hat{\mathrm{U}}_i(\mathrm{s}) \leftarrow \mathrm{R(s)} + \boldsymbol{\gamma} \sum_{s' \in \mathrm{S}} [P(s' | \mathrm{s}, \hat{\pi}_{i-1}(\mathrm{s}))\ \hat{\mathrm{U}}_{i-1}(s')]$$

$$\hat{\pi}_i(\mathrm{s}) \leftarrow \underset{\mathrm{a} \in \mathrm{A}}{\mathrm{argmax}} \sum_{s' \in \mathrm{S}} [P(\mathrm{s}' | \mathrm{s}, \mathrm{a})\ \hat{\mathrm{U}}_i(\mathrm{s}')]$$

Which are in turn derived from

$$\mathrm{U}^*(\mathrm{s}) = \mathrm{R}(\mathrm{s}) + \boldsymbol{\gamma} \sum_{\mathrm{s}' \in \mathrm{S}} \mathrm{P}(s' | \mathrm{s}, \Pi^*(\mathrm{s}))\mathrm{U}^*(s')$$

$$\Pi^*(\mathrm{s}) = \mathrm{argmax}_{\ \mathrm{a} \in \mathrm{A}} \sum_{\mathrm{s}' \in \mathrm{S}} \mathrm{P}(s' | \mathrm{s}, \mathrm{a})\mathrm{U}^*(s')$$

Q Learning

$$U^*(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s,\Pi*(s))U*(s')$$

Q Learning

$$U^*(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s, \Pi^*(s)) U^*(s')$$

$$U^*(s) = \max_a Q(s, a)$$

Q Learning

$$U^*(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s, \Pi^*(s)) U^*(s')$$

$$U^*(s) = \max_a Q(s, a)$$

$$\downarrow$$

$$U^*(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s, \Pi^*(s)) \max_{a' \in AQ}(s', a')$$

Q Learning

$$U^*(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s, \Pi^*(s)) U^*(s')$$

$$U^*(s) = \max_a Q(s, a)$$

$$\downarrow$$

$$U^*(s) = R(s) + \gamma \sum_{s' \in S} P(s'|s, \Pi^*(s)) \max_{a' \in AQ}(s', a')$$

$$\downarrow$$

$$Q(s,a) = R(s) + \gamma \sum_{s' \in S} P(s'|s, a) \max_{a' \in AQ}(s', a')$$

Q Learning

Policy iteration update rule:

$$\hat{\mathrm{U}}_i(\mathrm{s}) \leftarrow \mathrm{R(s)} + \boldsymbol{\gamma} \sum_{s' \in \mathrm{S}} [P(s' | \mathrm{s}, \hat{\pi}_{i-1}(\mathrm{s}))\ \hat{\mathrm{U}}_{i-1}(s')]$$

Replaces $\hat{\mathrm{U}}_i(\mathrm{s})$ with one-step lookahead using $\hat{\mathrm{U}}_{i-1}(s')$

Can we do something analogous for Q(s,a)?