MFML: MDP and Reinforcement Learning

Nicolas Gast

November 25, 2024

Overview of the course

Up to now:

1. Supervised / unsupervised learning.
   - Data $\mapsto$ model
2. Online learning
   - Decision $\mapsto$ Data $\mapsto$ Decisions

Overview of the course

Up to now:

1. Supervised / unsupervised learning.
   - Data $\mapsto$ model
2. Online learning
   - Decision $\mapsto$ Data $\mapsto$ Decisions

End of the course:

3. Reinforcement learning
   - State $\mapsto$ Decision $\mapsto$ Reward and new state

What is Reinforcement Learning?

And why it differs from supervised or unsupervised learning

What is Reinforcement Learning?

And why it differs from supervised or unsupervised learning

- No i.i.d. dataset, but an environmeent.
- No labels, but observation of rewards.
- We design an agent, that maps states to actions.

What is Reinforcement Learning?

And why it differs from supervised or unsupervised learning

- No i.i.d. dataset, but an environmeent.
- No labels, but observation of rewards.
- We design an agent, that maps states to actions.

Challenges:

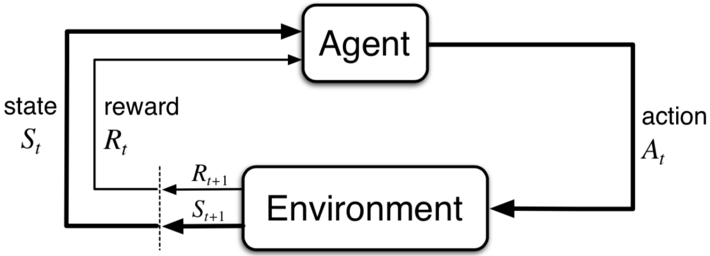
- Many possible states, actions
- Reward can be delayed, or sparse.

Applications

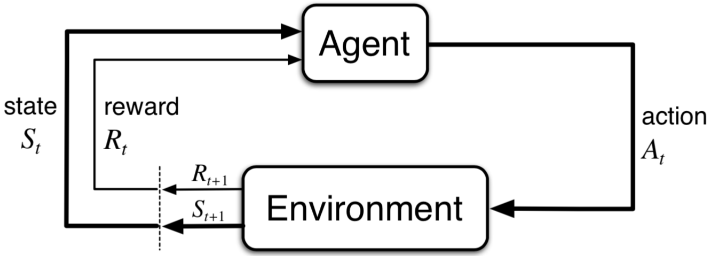
- Games (Go, Atari, StarCraft,...) ▸ StarCraft
- Auto-piloting vehicles ▸ Robots , ▸ Helicopter
- Supply management, energy ▸ data-center
- Trading, bidding ▸ Bidding
- Toy models ▸ AIGym
- . . .

The number of application is increasing.

RL is about interacting with an environment



RL is about interacting with an environment



1. Get an observation of the state of the environment
2. Choose an action
3. Obtain a reward

You goal is to select actions to maximize the total reward.

Reward signal

At time t, we observe S_t, take action A_t, and obtain a reward R_{t+1}.

$$S_1, A_1 \qquad R_2, S_2, A_2 \qquad R_3, S_3, A_3 \qquad \dots \qquad R_T, S_T$$

Reward signal

At time t, we observe S_t, take action A_t, and obtain a reward R_{t+1}.

$$S_1, A_1 \qquad R_2, S_2, A_2 \qquad R_3, S_3, A_3 \qquad \dots \qquad R_T, S_T$$

Impact of actions can be delayed.

- On which actions does the reward depend?

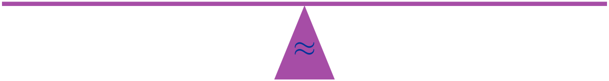
Impact of actions can be weak

- or noisy

Objective of this course

Theory

Practice



MDPs
Tabular RL
Algorithms guarantees
Regret minimization

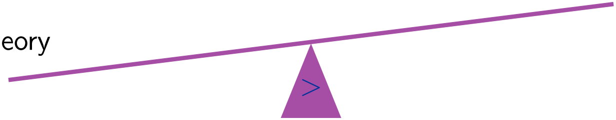
Deep RL
Policy gradient

Exercises

Objective of this course

Practice

Theory



MDPs
Tabular RL
Algorithms guarantees
Regret minimization

Deep RL
Policy gradient

Exercises

Personal work

Week	November 26	Dec 10	Dec 17
Tuesday	MDP	Tabular RL	"Modern" RL

- Mini-exam (Dec 10 and/or Dec 17)
- Final exam in January.

Personal work

Week	November 26	Dec 10	Dec 17
Tuesday	MDP	Tabular RL	"Modern" RL

- Mini-exam (Dec 10 and/or Dec 17)
- Final exam in January.

A few advice:

- Question what you learn
- Try to do some exercises.
  - Program, go deeper, ask follow-up questions.
- Ask questions during or after the course.

Personal work

Week	November 26	Dec 10	Dec 17
Tuesday	MDP	Tabular RL	“Modern” RL

- Mini-exam (Dec 10 and/or Dec 17)
- Final exam in January.

A few advice:

- Question what you learn
- Try to do some exercises.
  - Program, go deeper, ask follow-up questions.
- Ask questions during or after the course.

Read books (and/or research articles)

- (Introduction to Reinforcement Learning (Sutton-Barto, 2018 last ed.))
- Algorithms for Reinforcement Learning (Szepesvari, 2010)
- Deep Reinforcement learning: hands on (Maxim Lapan, 2020)

Content of the course

1. Markov Decision Processes (MDPs)
2. Tabular reinforcement learning
3. Large state-spaces and approximations
4. Monte-Carlo tree search (MCTS)

Outline

Illustrative example: the wheel of fortune

Policy:

number \ time	1	2	3	4	5
1	D			D	D
2	D				D
3					S
⋮		S	S	S	S
10	S	S	S	S	S

$$Q_g(7, \text{STOP}) \overset{?}{>} Q_g(7, \text{DRAW})$$

time — number on the wheel (state) — action

You can draw a wheel indefinitely. After time t:

- If you draw, the wheel stops on $X_t \in \{1 \dots 10\}$ (uniformly).
- You can draw again or stop and earn X_t.

- You can draw the wheel up to $T = 10$ times. How do you play?

The environement lives in a "space"

- $\mathcal{S}$ – state space.
- $\mathcal{A}$ – action space.
- $\mathcal{R}$ – reward space.

$\mathcal{S} = \{1, \dots, 10, \text{STOP}\}$

$\mathcal{A} = \{\text{STOP}, \text{DRAW}\}$

$\mathcal{R} \subseteq \mathbb{R}$

Dynamics:

- (possibly random) evolution of states
- (possibly random) rewards

Markov decision processes

$S_t, A_t \overset{\text{random}}{\underset{\text{transition}}{\rightsquigarrow}} R_{t+1}, S_{t+1}$

A MDPs is defined by:

- $\mathcal{S}$ state space
- $\mathcal{A}$ action set
- Evolution is driven by Markovian transitions

$$\mathbb{P}(S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a) = P(s', r \mid s, a).$$

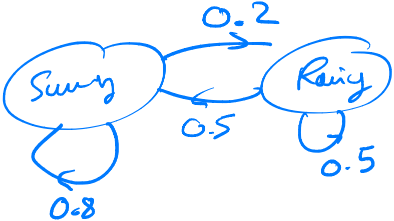
"

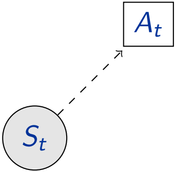
$\mathbb{P}(S_{t+1} = s', R_{t+1} = a \mid S_t = s, A_t = a$ and $H_t)$ history up to "A_t".

MDP = Markov chain + decisions

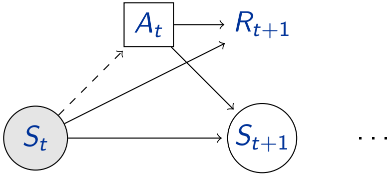
Most reinforcement learning problems can be framed as MDPs.

Graphical representation





Graphical representation



Some examples

- Wind production problem

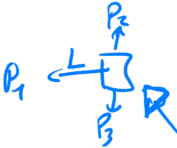
A Wind turbine produces $(W_t)^3 \cos(\theta_t)$ where W_t is the wind speed and θ_t is your angle with respect to wind. Assume that:

- Wind direction changes of ± 1 degree with probability $1/2$.
- Turning your turbine costs you $a > 0$.

Write the MDP for different models:

- Assuming that W_t is constant.
- Assuming that $W(t)$ evolve over time $W(t+1) = \min(1, \max(0, W(t) \pm b))$.
- Assuming that the direction in which the wind changes stays the same with probabilty 90%.

Some examples



- Wind production problem
- Frozen-lake ▸ Link (this is a gridworld example)

S	F	F	F
F	H	F	H
F	F	F	H
H	F	F	G

- ▸ Set space: $\{(0,0),\ldots,(3,3)\}$.
- ▸ Actions: $\{L,R,U,D\}$.
- ▸ Transitions: 1/3 in right direction.
- ▸ Rewards: there are Holes and a Goal.
  - ★ Jumping to the goal gives you "1".

- There also some deterministic MDPs
  - ▸ Shortest paths probmems
  - ▸ Deterministic games (e.g., go, chess)

Table of contents

Policies

A (deterministic) policy specifies which action to take in a given state:

$$\pi : \mathcal{S} \to \mathcal{A}.$$

It indicates which action to take in a given state: $A_t = \pi(S_t)$. This defines the behavior of the agent.

Policies

A (deterministic) policy specifies which action to take in a given state:

$$\pi : \mathcal{S} \to \mathcal{A}.$$

It indicates which action to take in a given state: $A_t = \pi(S_t)$. This defines the behavior of the agent.

A stochastic policy specifies a distribution over actions:

$$\pi : \mathcal{S} \times \mathcal{A} \to [0, 1].$$

The agent takes $A_t \sim \pi(\cdot|S_t)$.

$\pi(a|s)$

Policies

A (deterministic) policy specifies which action to take in a given state:

$$\pi : \mathcal{S} \to \mathcal{A}.$$

It indicates which action to take in a given state: $A_t = \pi(S_t)$. This defines the behavior of the agent.

A stochastic policy specifies a distribution over actions:

$$\pi : \mathcal{S} \times \mathcal{A} \to [0, 1].$$

The agent takes $A_t \sim \pi(\cdot|S_t)$.

Deterministic	Stochastic
Optimal in general	It forces exploration Useful in games / non-Markovian Differentiable

Example of a (deterministic) policy

Return of a policy

We want to compute the best policy... But what is the best policy?

Return of a policy

We want to compute the best policy... But what is the best policy?

Do we choose A_t to optimize:

- R_{t+1}? (no: too greedy)

Return of a policy

We want to compute the best policy... But what is the best policy?

Do we choose A_t to optimize:

- R_{t+1}? (no: too greedy)
- R_T? (only final reward?)

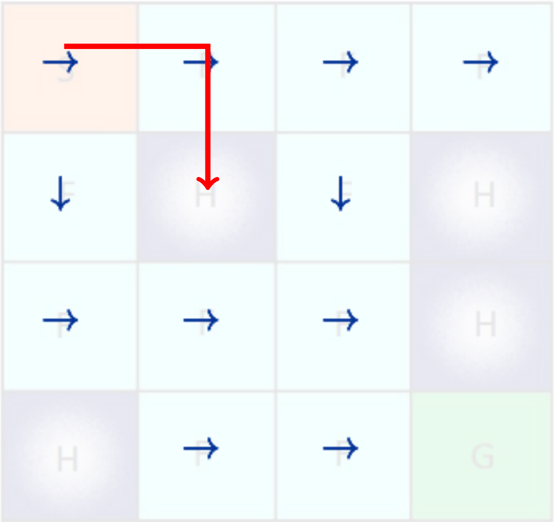
Return of a policy (finite horizon)

Sometimes, a problem has a known finite horizon T. In which case, the return (a.k.a. gain) at time t is:

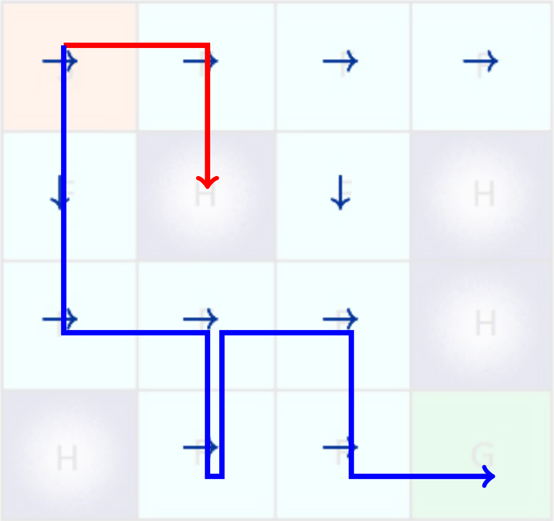
$$G_t = R_{t+1} + R_{t+2} + \cdots + R_T.$$

The return is random.

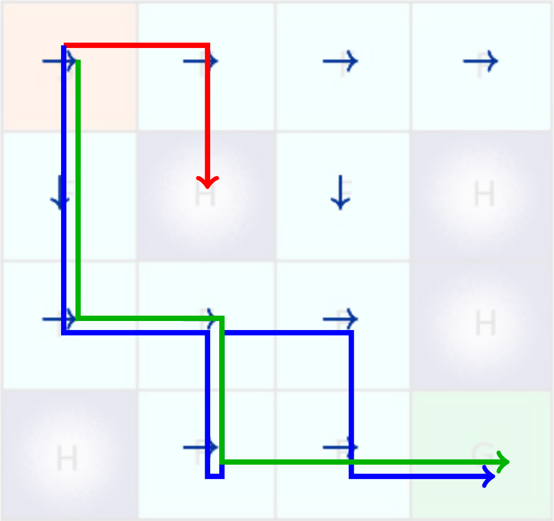
Return: example



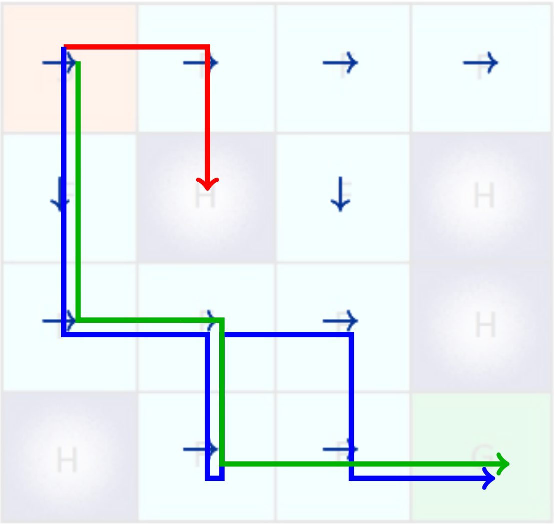
Return: example



Return: example



Return: example



Return(Red) = 0
Return(Green)=1
Return(Blue) = 1.

The return is random.

In practice, we will look at the expected return $\mathbb{E}[G_t]$.

Table of contents

Value function

state

The value function of a policy π is

$$V_t^{\pi}(s) = \mathbb{E}^{\pi}\left[G_t \mid S_t = s\right],$$

where $\mathbb{E}^{\pi}[\cdot]$ means $\mathbb{E}\left[\cdot \mid A_{t+k} \sim \pi(S_{t+k}) \quad (k \geq 0)\right]$.

time

Value function

The value function of a policy π is

$$V_t^{\pi}(s) = \mathbb{E}^{\pi}\left[G_t \mid S_t = s\right],$$

where $\mathbb{E}^{\pi}\left[\cdot\right]$ means $\mathbb{E}\left[\cdot \mid A_{t+k} \sim \pi(S_{t+k}) \quad (k \geq 0)\right]$.

It specifies the expected return. For each t, it is a vector of $|\mathcal{S}|$ values. If $\mathcal{S} = \{s_1 \dots s_4\}$

	s_1	s_2	s_3	s_4
V				

Bellman's Equation (policy evaluation, finite horizon)

We have $V_t^\pi(s) = \mathbb{E}^\pi[G_t \mid S_t = s]$ and

$$\begin{aligned} G_t &= R_{t+1} + R_{t+2} + \dots R_T \\ &= R_{t+1} + G_{t+1}. \end{aligned}$$

Hence:

$$\begin{aligned} V_t^\pi(s) &= \mathbb{E}^\pi[R_{t+1} + G_{t+1} \mid S_t = s] \\ &= \mathbb{E}^\pi[R_{t+1} \mid S_t = s] + \mathbb{E}^\pi[G_{t+1} \mid S_t = s] \\ &= r(S_t, A_t) + \sum_{s'} \mathbb{E}^\pi[G_{t+1} \mid S_{t+1} = s'] \mathbb{P}^\pi(S_{t+1} = s' \mid S_t = s) \\ &= r(S_t, \pi(S_t)) + \sum_{s'} V_{t+1}^\pi(s') P(s' \mid s, \pi(s)) \end{aligned}$$

where $r(s, a) = \sum_{r'} r' p(r' \mid s, a) = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$

Bellman's Equation (policy evaluation, finite horizon)

We have $V_t^\pi(s) = \mathbb{E}^\pi\left[G_t \mid S_t = s\right]$ and

$$\begin{aligned} G_t &= R_{t+1} + R_{t+2} + \dots R_T \\ &= R_{t+1} + G_{t+1}. \end{aligned}$$

Hence:

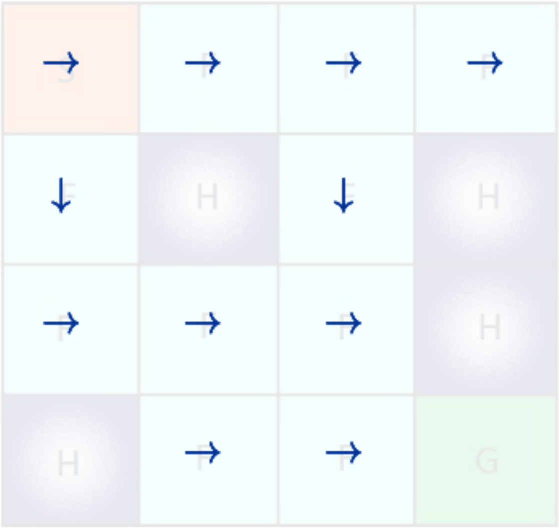
$$\begin{aligned} V_t^\pi(s) &= \underbrace{\sum_{s',r}(r + V^\pi(s'))p(s', r \mid s, a = \pi(s)).}_{=Q_{t+1}^\pi(s,\pi(s))} \\ &= \mathrm{r}(s, \pi(s)) + \sum_{s'} V_{t+1}^\pi(s')p(s' \mid s, a = \pi(s)), \end{aligned}$$

Bellman's equation

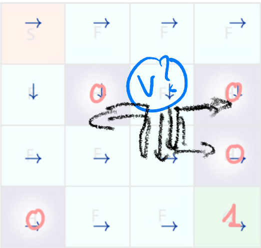
where $\mathrm{r}(s, a) = \sum_{r'} r' p(r' \mid s, a)$.

Algorithm: backward induction

Example : Finite-horizon Bellman's equation (evaluation)



Example : Finite-horizon Bellman's equation (evaluation)

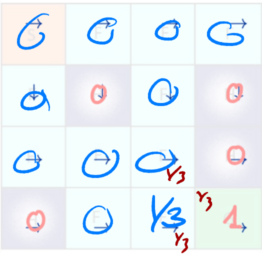


$t = 1$

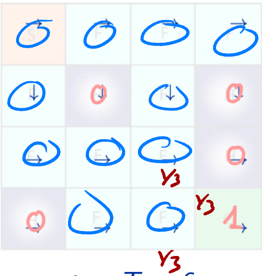
$t = 2$

$t = 3$

$t = 4$



$t = 5$



$t = T = 6$

Action-Value function

The action-value function of a policy π is

$$Q^{\pi}(s, a) = \mathbb{E}^{\pi}[G_t \mid S_t = s \wedge A_t = a].$$

Action-Value function

The action-value function of a policy π is

$$Q^{\pi}(s, a) = \mathbb{E}^{\pi}\left[G_t \mid S_t = s \wedge A_t = a\right].$$

It is a table of $|\mathcal{S}| \times |\mathcal{A}|$ values. If $\mathcal{S} = \{s_1 \dots s_4\}$ and $\mathcal{A} = \{a_1, a_2\}$:

Q	a_1	a_2
s_1		
s_2		
s_3		
s_4		

From Q, we can define a greedy policy: $a_t = \arg\max_{a \in \mathcal{A}} Q(s_t, a)$.

Optimal policy

We denote by $V_t^*(s) = \max_\pi V_t^\pi(s)$ and $Q_t^*(s,a) = \max_\pi Q_t^\pi(s, a)$
For a finite-horizon T, a policy is a function $\pi : \mathcal{S} \times \{1 \dots T\} \to \mathcal{A}$.

$$V_t^*(s) = \max_a Q_t^*(s,a)$$
$$Q_t^*(s,a) = r(s,a) + \sum_{s'} V_{t+1}^*(s') P(s'|s,a)$$

$$V_T^*(s) = 0$$

Optimal policy

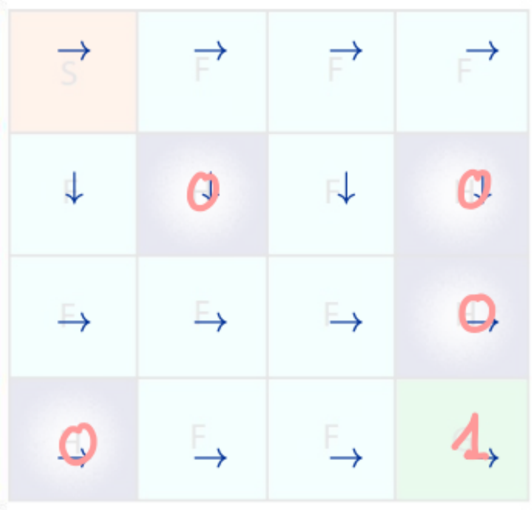
We denote by $V_t^*(s) = \max_\pi V_t^\pi(s)$ and $Q_t^*(s,a) = \max_\pi Q_t^\pi(s)$.
For a finite-horizon T, a policy is a function $\pi : \mathcal{S} \times \{1 \dots T\} \to \mathcal{A}$.

$$V_t^*(s) = \max_a Q_t(s,a)$$
$$Q_t^*(s,a) = \sum_{s',a} (r + V_{t+1}^*(s')) P(s', r \mid s, a)$$

Initial condition:

$$V_T^*(s) = \max_a r(s,a)$$

Optimal policy (finite horizon): illustration



$T = 6$

$Q_5^*((2,3),D) = 1/3$

$Q_5^*((2,3),R) = 1/3$

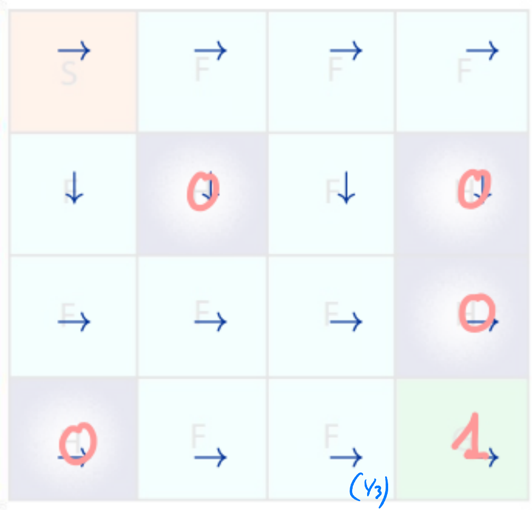
$Q_5^*((2,3),L) = 0$

$Q_5^*((2,3),U) = 1/3$

$$V_5^*((2,3)) = \frac{1}{3}$$

$$\pi_5^*((2,3)) \in \{D, R, U\}$$

Optimal policy (finite horizon): illustration



$T = 6$

$Q_5^*((2,3), D) =$ 1/3

$Q_5^*((2,3), R) =$ 1/3

$Q_5^*((2,3), L) =$ 0

$Q_5^*((2,3), U) =$ 1/3

$V_5^*((2,3)) = 1/3$

$Q_4^*((2,3), D) =$ 4/9

$Q_4^*((2,3), R) =$ 4/9

$Q_4^*((2,3), L) =$ 1/9

$Q_4^*((2,3), U) =$ 1/3

$V_4^*((2,3)) = 4/9$

$\pi^*((2,3)) \in \{D, R\}$

Optimal policy (finite horizon): illustration

```
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.33  1
L L L L
L L L L
L L L L
L L D L
```

```
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.11 0.00
0.00 0.11 0.44  1
L L L L
L L L L
L L L L
L D D L
```

```
0.00 0.00 0.00 0.00
0.00 0.00 0.04 0.00
0.00 0.07 0.15 0.00
0.00 0.19 0.52  1
L L L L
L L L L
L D L L
L D D L
```

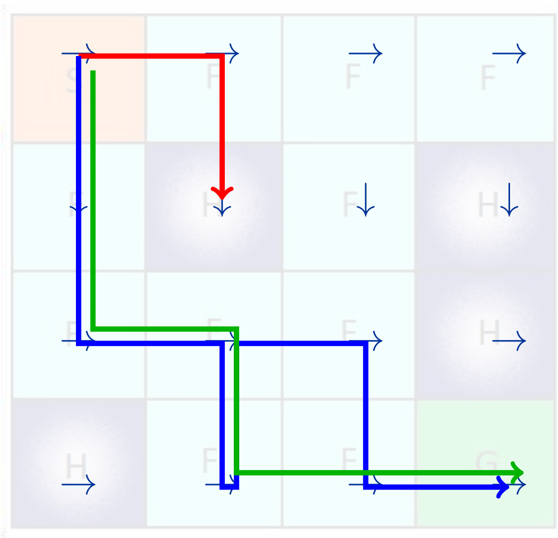
```
0.00 0.00 0.01 0.00
0.00 0.00 0.05 0.00
0.02 0.11 0.21 0.00
0.00 0.26 0.57  1
L L L L
L L L L
D D L L
L R D L
```

```
0.00 0.00 0.02 0.00
0.01 0.00 0.07 0.00
0.05 0.16 0.24 0.00
0.00 0.31 0.61  1
L D L L
L L L L
D D L L
L R D L
```

```
0.00 0.01 0.03 0.01
0.02 0.00 0.09 0.00
0.07 0.20 0.28 0.00
0.00 0.36 0.64  1
D R L U
L L L L
U D L L
L R D L
```

Table of contents

Which trajectory is best?



Return of a policy (discounted infinite horizon)

$\beta \quad \sqrt{\beta} \quad \beta^{1/3} \quad \beta^{1/4} \quad \dots$

$e^{-\alpha} \quad e^{-2\alpha} \quad e^{-3\alpha} \quad \dots$

When T is not specified, it is common to look at the discounted return:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$$
$$= \sum_{k=0}^{\infty} \gamma^k R_{t+1+k},$$

with $\gamma \in [0, 1)$.

$$G_t = R_{t+1} + \gamma G_{t+1}$$

- $\gamma = 0$: myopic (greedy).
- $\gamma = 1$: total reward.

Finite h~~orizon~~

$$V_t^{\pi}(s) = r(s,\pi(s)) + \sum_{s'} V_{t+1}^{\pi}(s')\, P(s' \mid s,\pi(s))$$

Discount:

$$V_t^{\pi}(s) = r(s,\pi(s)) + \gamma \sum_{s'} V_{t+1}^{\pi}(s')\, P(s' \mid s,\pi(s))$$

$$G_t = R_{t+1} + \gamma\, G_{t+1}$$

Value of a policy and value iteration

Call T^π the operator that associates to a vector V the vector $T^\pi V$:

$$T^\pi V(s) = \mathrm{r}(s, \pi(s)) + \gamma \sum_{s'} V(s') p(s' \mid s, a = \pi(s))$$

The value of a policy is the unique vector V^π such that $T^\pi V^\pi = V^\pi$.

$$V_t(s) = (T^\pi V_{t+1})(s)$$

Value of a policy and value iteration

Call T^π the operator that associates to a vector V the vector $T^\pi V$:

$$T^\pi V(s) = \mathrm{r}(s, \pi(s)) + \gamma \sum_{s'} V(s') p(s' \mid s, a = \pi(s))$$

The value of a policy is the unique vector V^π such that $T^\pi V^\pi = V^\pi$.

Proof. T^π is contracting for the $\|v\| = \max_s |v(s)|$.

$$\| T^\pi V - T^\pi V' \| \leq \gamma \| V - V' \|$$

How to compute V^π

$$\leftarrow \cdots \quad V_{T-2} \leftarrow V_{T-1} \leftarrow V_T$$

$$V^{(2)} \leftarrow V^{(1)} \leftarrow V^{(0)}$$

Two solutions:

1. Solve the linear system.
2. Initialize $V^{(0)} = 0$ and apply $V^{(k+1)} = T^\pi V^{(k)}$ until convergence.

The optimal policy

We denote by $V^*(s) = \max_\pi V^\pi(s)$ and $Q^*(s,a) = \max_\pi Q^\pi(s,a)$.

The optimal policy π^* is such that:

$$\pi^* = \arg\max_\pi V^\pi(s) \qquad \forall s \in \mathcal{S}$$

or equivalently:

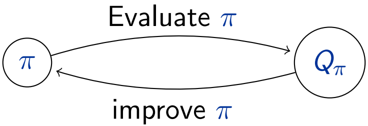
$$\pi^* = \arg\max_\pi Q^\pi(s,a) \qquad \forall s \in \mathcal{S}, s \in \mathcal{A}$$

Iterative solutions

$$\begin{tikzcd} \pi \arrow[r, bend left, "\text{Evaluate } \pi"] & Q_\pi \arrow[l, bend left, "\text{improve } \pi"] \end{tikzcd}$$

If you know the transitions and reward: value iteration or policy iteration.

Iterative solutions



If you know the transitions and reward: value iteration or policy iteration.
Value iteration:

- Initialize V^0 (for instance to 0).
- For $k \geq 0$ and $s \in \mathcal{S}$, do:
  $$V^{k+1}(s) := \max_{a\in\mathcal{A}} \left(\mathrm{r}(s,a) + \gamma \sum_{s'} V^k(s')p(s' \mid s,a)\right)$$

"Theorem": If $\gamma < 1$, then $\|V^k - V^*\| = O(\gamma^k)$.

$$\left(V = \underbrace{R + \gamma P V}_{T^\pi V} \right)$$

Illustration

```
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.00    1
L L L L
L L L L
L L L L
L L L L
```

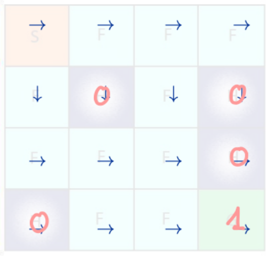
$\gamma = 0.8$

```
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.00    1
L L L L
L L L L
L L L L
L L L L
```

$\gamma = 0.9$

Iteration $k = 0$

Illustration



```
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.33   1
L L L L
L L L L
L L L L
L L D L
```

$\gamma = 0.8$

```
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.33   1
L L L L
L L L L
L L L L
L L D L
```

$\gamma = 0.9$

Iteration $k = 1$

Illustration

```
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.09 0.00
0.00 0.09 0.42   1
L  L  L  L
L  L  L  L
L  L  L  L
L  D  D  L
```

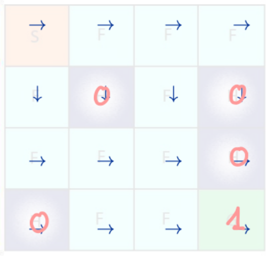
$\gamma = 0.8$

```
0.00 0.00 0.00 0.00
0.00 0.00 0.00 0.00
0.00 0.00 0.10 0.00
0.00 0.10 0.43   1
L  L  L  L
L  L  L  L
L  L  L  L
L  D  D  L
```

$\gamma = 0.9$

Iteration $k = 2$

Illustration



```
0.00 0.00 0.00 0.00
0.00 0.00 0.02 0.00
0.00 0.05 0.11 0.00
0.00 0.14 0.47   1
L  L  L  L
L  L  L  L
L  D  L  L
L  D  D  L
```

$\gamma = 0.8$

```
0.00 0.00 0.00 0.00
0.00 0.00 0.03 0.00
0.00 0.06 0.13 0.00
0.00 0.16 0.49   1
L  L  L  L
L  L  L  L
L  D  L  L
L  D  D  L
```

$\gamma = 0.9$

Iteration $k = 3$

Illustration

```
0.00 0.00 0.01 0.00
0.01 0.00 0.04 0.00
0.03 0.10 0.17 0.00
0.00 0.21 0.52   1
D R L U
L L L L
U D L L
L R D L
```

$\gamma = 0.8$

```
0.00 0.01 0.02 0.01
0.01 0.00 0.06 0.00
0.05 0.14 0.22 0.00
0.00 0.28 0.57   1
D R L U
L L L L
U D L L
L R D L
```

$\gamma = 0.9$

Iteration $k = 4$

Illustration

0.02	0.02	0.03	0.02
0.03	0.00	0.06	0.00
0.06	0.13	0.20	0.00
0.00	0.25	0.54	1

```
D U R U
L L L L
U D L L
L R D L
```

$\gamma = 0.8$

0.07	0.06	0.07	0.06
0.09	0.00	0.11	0.00
0.15	0.25	0.30	0.00
0.00	0.38	0.64	1

```
L U L U
L L L L
U D L L
L R D L
```

$\gamma = 0.9$

Iteration $k = 90$

$$\frac{\gamma^k}{1-\gamma}$$

Policy iteration

Policy iteration:

- Initialize π^0 (to some random value).
- For $k \geq 0$:

  Compute Q^{π^k} (=linear system)

  For all $a \in \mathcal{A}$: $\pi^{k+1}(s) := \arg\max_{a \in \mathcal{A}} Q^{\pi^k}(s, a)$.

"Theorem": If $\gamma < 1$, then after a finite number of iterations: $V^k = V^*$.

Exercise: the wheel of fortune

You can draw a wheel indefinitely. After time t:

- If you draw, the wheel stops on $X_t \in \{1 \dots 10\}$ (uniformly). You earn X_t.
- You can draw again or keep $X_{t+1} := X_t$.

How do you play knowing that you want to maximize your discounted reward: $\mathbb{E}\left[\sum_{t=1}^{\infty} \delta^t X_t\right]$ with $\gamma = 0.9$?

- Compare value iteration and policy iteration algorithms.

Table of contents

Important concepts

(to be filled by you!)