Reinforcement Learning
Sequential Decisions, MDP and Policies

E. Le Pennec

M2DS - Reinforcement Learning – Fall 2023
Outline

1. Decision Process and Markov Decision Process
2. Returns and Value Functions
3. Prediction and Planning
4. Operations Research and Reinforcement Learning
5. Control
6. Survey
7. References
Decision or Decisions

Source: W. Powell
In many (most?) settings, not a single decision but a sequence of decisions. Need to take into account the (not necessarily immediate) consequences of the sequence of decisions/actions rather than of each decision. Different framework than supervised learning (no immediate feedback here) and unsupervised learning (well defined goal here).
From Sequential Decision to Reinforcement Learning

Sequential Decision

- Sequence of action $A_t$ as a response of an environment $S_t$
- Feedback through a reward $R_t$

Actions?

- Is my current way of choosing actions good?
- How to make it better?
From Sequential Decision to Reinforcement Learning

Markov Decision Process Modeling
- Specific modeling of the environment.
- Goal as a (weighted) sum of a scalar reward.

Actions?
- Is my current way of choosing actions good?
- How to make it better?
From Sequential Decision to Reinforcement Learning

Sequential Decision

$$S_t, A_t, S_{t+1}, R_{t+1}$$

Agent

MDP Modeling

$$S_t, A_t, S_{t+1}, R_{t+1}$$

Agent

Reinforcement Learning

$$S_t, A_t, S_{t+1}, R_{t+1}$$

Agent

Interaction

Reinforcement Learning

- Same modeling...
- But no direct knowledge of the MDP.

Actions?

- Is my current way of choosing actions good?
- How to make it better?
Sequential Decision Settings

- MDP / Reinforcement Learning:
  \[ \max_{\pi} \mathbb{E}_\pi \left[ \sum_t R_t \right] \]

- Optimal Control:
  \[ \min_u \mathbb{E} \left[ \sum_t C(x_t, u_t) \right] \]

- (Stochastic) Search:
  \[ \max_{\theta} \mathbb{E}[F(\theta, W)] \]

- Online Regret:
  \[ \max \sum_k \mathbb{E}[F(\theta_k, W)] \]
References

R. Sutton and A. Barto. 
*Reinforcement Learning, an Introduction (2nd ed.)*
MIT Press, 2018

O. Sigaud and O. Buffet. 
*Markov Decision Processes in Artificial Intelligence.*
Wiley, 2010

M. Puterman. 
Wiley, 2005

D. Bertsekas and J. Tsitsiklis. 
*Neuro-Dynamic Programming.*
Athena Scientific, 1996

W. Powell. 
Wiley, 2022

S. Meyn. 
*Control Systems and Reinforcement Learning.*
Cambridge University Press, 2022

V. Borkar. 
*Stochastic Approximation: A Dynamical Systems Viewpoint.*
Springer, 2008

T. Lattimore and Cs. Szepesvári. 
*Bandit Algorithms.*
Cambridge University Press, 2020
Outline

1 Decision Process and Markov Decision Process
2 Returns and Value Functions
3 Prediction and Planning
4 Operations Research and Reinforcement Learning
5 Control
6 Survey
7 References
Decision Process and Environment

- At time step $t \in \mathbb{N}$:
  - State $S_t \in S$: representation of the environment
  - Action $A_t \in \mathcal{A}(S_t)$: action chosen
  - Reward $R_{t+1} \in \mathcal{R}$: instantaneous reward
  - New state $S_{t+1}$

- Focus on the discrete setting, i.e. $S$ finite, $\mathcal{A}(s)$ finite and $\mathcal{R}$ finite.
- Extension: Non finite bounded $\mathcal{R}$: easy / Non finite $S$: hard / Non finite $\mathcal{A}$: harder.
Stochastic Model

- Dynamic defined by:

\[ \mathbb{P}(S_{t+1} = s', R_{t+1} = r | (S_{t'}, A_{t'}, R_{t'}), t' \leq t) = \mathbb{P}(S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a, H_t) \]

where \( H_t = (R_t, S_{t-1}, A_{t-1}, R_{t-1}, S_{t-2}, \ldots) \) is the past and \( (S_t, A_t) \) the present.
Markov Decision Process and Environment

Markovian Environment

- Markovian Dynamic Assumption: \( S_{t+1} \) and \( R_{t+1} \) are independent of the past \( H_t = (R_t, S_{t-1}, A_{t-1}, R_{t-1}, S_{t-2}, \ldots) \) conditionally to the present \((S_t, A_t)\).

- Dynamic entirely defined by state-reward transition probabilities

\[
\mathbb{P}(S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a, H_t) = \mathbb{P}(S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a) = p(s', r | s, a)
\]

in the discrete setting.

- Informally, this means that \( S_t \) encodes all the information related to the past.
Markov Decision Process and State-Action

- State-Reward transition probabilities for a given state-action:
  \[
  \mathbb{P}(S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a, H_t) = \mathbb{P}(S_{t+1} = s', R_{t+1} = r | S_t = s, A_t = a) \\
  = p(s', r | s, a)
  \]

**Induced State-action laws**

- State transition probabilities for a given state-action:
  \[
  \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a, H_t) = \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a) \\
  = p(s' | s, a) = \sum_r p(s', r | s, a)
  \]

- Expected reward for a given state-action:
  \[
  \mathbb{E}[R_{t+1} | S_t = s, A_t = a, H_t] = \mathbb{E}[R_{t+1} | S_t = s, A_t = a] \\
  = r(s, a) = \sum_r \sum_{r', s'} p(s', r | r, a)
  \]

- From now on, we will always assume that the Markovian property holds for the environment.
### Decision Process and Markov Decision Process

#### Examples

| $s$    | $a$      | $s'$   | $p(s'|s,a)$ | $r(s,a,s')$ |
|--------|----------|--------|-------------|-------------|
| high   | search   | high   | $\alpha$   | $r_{search}$|
| high   | search   | low    | $1 - \alpha$| $r_{search}$|
| low    | search   | high   | $1 - \beta$ | $-3$        |
| low    | search   | low    | $\beta$    | $r_{search}$|
| high   | wait     | high   | 1           | $r_{wait}$  |
| high   | wait     | low    | 0           | -           |
| low    | wait     | high   | 0           | -           |
| low    | wait     | low    | 1           | $r_{wait}$  |
| low    | recharge | high   | 1           | 0           |
| low    | recharge | low    | 0           | -           |

#### Diagram

![Markov Decision Process Diagram](image)
Decision Process, Agent and Policy

Agent

- Interact with the environment by choose the action given the past.

Policy $\Pi$: specification of how to choose the action

- General stochastic policy $\Pi = (\pi_0, \pi_1, \ldots, \pi_t, \ldots)$:
  \[ \Pi_t(A_t = a) = \pi_t(A_t = a | S_t = a, A_t = a, H_t) \]

- General deterministic policy $\Pi = (\pi_0, \pi_1, \ldots, \pi_t, \ldots)$ (with as slight abuse of notation):
  \[ \Pi_t(A_t = a) = 1_{A_t = \pi_t(S_t = a, A_t = a, H_t)} \]
Markov Decision Process, Agent and Policy

**Agent**
- Interact with the environment by choose the action given the past.

**Policy \( \Pi \): specification of how to choose the action**
- History dependent stochastic policy \( \Pi = (\pi_0, \pi_1, \ldots, \pi_t, \ldots) \):
  \[
  \Pi_t(A_t = a) = \pi_t(A_t = a | S_t = s, H_t)
  \]
- Markovian stochastic policy \( \Pi = (\pi_0, \pi_1, \ldots, \pi_t, \ldots) \):
  \[
  \Pi_t(A_t = a) = \pi_t(A_t = a | S_t = s) = \pi_t(a | s)
  \]
- Stationary Markovian stochastic policy \( \Pi = (\pi, \pi, \ldots, \pi, \ldots) \):
  \[
  \Pi_t(A_t = a) = \pi(A_t = a | S_t = s) = \pi(a | s)
  \]

- Similar deterministic policy definition.
- Partially Observed Markov Decision Process extension: the Agent has only access to a partial observation \( O_t \) at each time step... (not the focus of the lectures)
Decision Process and Trajectories

- Trajectory \((S_0, A_0, R_1, S_1, A_1, \ldots)\) defined by
  - an initial distribution \(P_0\) for \(S_0\),
  - a policy \(\Pi = (\pi_0, \pi_1, \ldots, \pi_t, \ldots)\) specifying
    \[ \Pi_t(A_t = a) = \pi_t(A_t = a|S_t, H_t) \]
  - an environment specifying
    \[ P(S_{t+1}, R_{t+1}|S_t, A_t, H_t) \]
Decision Process and Trajectories

Induced probability:

\[ P(S_0 = s_0, A_0 = a_0, R_1 = r_1, S_1 = s_1, A_1 = a_1, \ldots S_t = s_t, R_t = r_t) = P_0(S_0 = s_0) \times \pi_0(A_0 = a_0|S_0) P(S_1, R_1|S_0, A_0) \pi_1(A_1 = a_1|S_1 = s_1, H_1) \times \ldots \times P(S_t = s_t, R_t = r_t|S_{t-1} = s_{t-1}, A_{t-1} = a_{n-1}, H_{t-1}) \]
Markov Decision Process and Trajectories

Trajectories

- Trajectory \((S_0, A_0, R_1, S_1, A_1, \ldots)\) defined by
  - an initial distribution \(P_0\) for \(S_0\),
  - a policy \(\Pi = (\pi_0, \pi_1, \ldots, \pi_t, \ldots)\) specifying
    \[ \Pi_t(A_t = a) = \pi_t(A_t = a | S_t, H_t) \]
  - a Markovian environment specifying
    \[ P(S_{t+1}, R_{t+1} | S_t, A_t) \]
Markov Decision Process and Trajectories

Induced probability:
\[ P(S_0 = s_0, A_0 = a_0, R_1 = r_1, S_1 = s_1, A_1 = a_1, \ldots S_t = s_t, R_t = r_t) = P_0(S_0 = s_0) \times \pi_0(A_0 = a_0|S_0) P(S_1, R_1|S_0, A_0) \pi_1(A_1 = a_1|S_1 = s_1, H_1) \times \cdots \times P(S_t = s_t, R_t = r_t|S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1}) \]
Markov Decision Process and Trajectories

Markovian Trajectories only if the policy is Markovian

- \( P(R_{t+1}, S_{t+1}, A_{t+1}, R_{t+2}, S_{t+2}, \ldots R_{t+k}, S_{t+k} | S_t, A_t, H_t) \)
  
  \( = P(R_{t+1}, S_{t+1}, A_{t+1}, R_{t+2}, S_{t+2}, \ldots R_{t+k}, S_{t+k} | S_t, A_t) \)
  
  \( = P(S_{t+1}, R_{t+1} | S_t, A_t) \pi_{t+1}(A_{t+1} | S_{t+1}) \)
  
  \( \times \ldots \times P(S_{t+k}, R_{t+k} | S_{t+k-1}, A_{t+k-1}) \)

- Stationary if the policy is stationary.
Outline

1 Decision Process and Markov Decision Process
2 Returns and Value Functions
3 Prediction and Planning
4 Operations Research and Reinforcement Learning
5 Control
6 Survey
7 References
Rewards and Total Return

- MDP: Rewards $R_t$ encode all the feedbacks!
- Quality of a policy $\Pi$ measured from the remaining total return:
  \[ G_t = \sum_{t'=t+1}^{\infty} R_{t'} \]
- Expected total return following $\Pi$ starting from $s$:
  \[ \mathbb{E}_\Pi[G_t | S_t = s] = \sum_{t'=t+1}^{\infty} \mathbb{E}[R_{t'} | S_t = s] \]
## Total Return: Issue and Fixes

### Issues
- $G_t$ is a limiting process and thus may not be defined!
- Can diverge to $\pm \infty$ and not converge at all.

### Fixes?
- Finite horizon: $G^T_t = \sum_{t'=t+1}^{T} R_{t'}$
- Episodic setting: it exists a random $T$ such that $\forall t' \geq R$, $R_{t'} = 0$ and $\mathbb{E}[T] < \infty$ so that $G_t = \sum_{t'=t+1}^{\infty} R_{t'}$ is well defined.
- Discounted setting: for $0 < \gamma < 1$, $G^\gamma_t = \sum_{t'=t+1}^{\infty} \gamma^{t'-(t+1)} R_{t'}$
- Average return: $\bar{G}_t = \lim_{T \to \infty} \frac{1}{T} \sum_{t'=t+1}^{t+T} R_{t'}$
Finite Horizon

\[ G_t^T = \sum_{t'=t+1}^{T} R_{t'} \]

Finite Horizon Setting

- Always well defined and easy to interpret.
- Loss of Markovianity as we need to know the time step...
- Can be put in a classical Markov framework!
  - Define an absorbing state \( s_{\text{abs}} \) (a state that cannot be escaped and from which the reward is always 0).
  - Extend the state space \( S \) to \( (S \times \{0, \ldots, T\}) \cup \{s_{\text{abs}}\} \).
  - Define an state reward transition probability:
    \[
    p(\tilde{s}', r|\tilde{s}, a) = \begin{cases} 
    p(s', t|s, a) & \text{if } \tilde{s} = (s, t), \ t < T \text{ and } \tilde{s}' = (s', t+1) \\
    1 & \text{if } \tilde{s} = (s, t), \ t = T, \ \tilde{s}' = s_{\text{abs}} \text{ and } r = 0 \\
    1 & \text{if } \tilde{s} = s_{\text{abs}}, \ \tilde{s}' = s_{\text{abs}} \text{ and } r = 0 \\
    0 & \text{otherwise}
    \end{cases}
    \]
Episodic Setting

\[ G_t = \sum_{t'=t+1}^{\infty} R_{t'} \]

Assumption: for any policy \( \Pi \), the average duration before \( R_t = 0 \) is smaller than a finite horizon \( H \):

\[ E_{\Pi} \left[ \min_{t, R_{t'} = 0, \forall t' \geq t} t \right] \leq H < +\infty \]

Strong assumption...

Easy to interpret.

Equivalent def.:
- Replace all the states from which \( R_t \) remains equal to 0 whatever the policy by a single absorbing state \( s_{abs} \),
- Assumption: for any policy \( \Pi \), the average duration to reach this state is smaller than a finite horizon \( H \):

\[ E_{\Pi} \left[ \min_{t, S_t = s_{abs}} t \right] \leq H < +\infty \]
Discounted

$G_t^\gamma = \sum_{t'=t+1}^T \gamma^{t'-(t+1)} R_{t'}$

Discounted

- Always defined but not that easy to interpret.
- Easiest theoretical setting!

Equivalent to an episodic setting if one adds an absorbing state $s_{abs}$ and changes all state-reward transition probabilities to:

$$p(s', r|s, a) = \begin{cases} 
\gamma p(s', r|s, a) & \text{if } s' \neq s_{abs}, s \neq s_{abs} \\
(1 - \gamma) & \text{if } s' = s_{abs}, r = 0, s \neq s_{abs} \\
1 & \text{if } s' = s_{abs}, r = 0, s = s_{abs} \\
0 & \text{otherwise}
\end{cases}$$

- Horizon $H = 1/(1 - \gamma)$. 
Average Return Setting

\[ \bar{G}_t = \lim_{T \to \infty} \frac{1}{T} \sum_{t'=t+1}^{t+T} R_{t'} \]

**Average Return**

- Not always defined. (Cesaro Average)
- Always equal to 0 in the episodic setting!
- Natural definition in a *stationary* setting...
- Complex theoretical analysis!

- Under a strict stationarity assumption \((R_t \sim R_{t'})\), link with discounted setting as

\[
\mathbb{E}_\pi[G_t^\gamma] = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_\pi[R_{t+1}] = \frac{1}{1 - \gamma} \mathbb{E}_\pi[R_t] = \frac{1}{1 - \gamma} \mathbb{E}_\pi[\bar{G}_t]
\]
State Value Functions

- Return expectation for a policy $\Pi$ starting from $s$ at time $t$
  - Finite horizon setting:
    \[ v_{t,\Pi}^T(s) = \mathbb{E}_{\Pi}[G^T_t | S_t = s] = \sum_{t' = t+1}^T \mathbb{E}_{\Pi}[R_{t'} | S_t = s] \]
  - Episodic setting:
    \[ v_{t,\Pi}(s) = \mathbb{E}_{\Pi}[G_t | S_t = s] = \sum_{t' = t+1}^{\infty} \mathbb{E}_{\Pi}[R_{t'} | S_t = s] \]
  - Discounted:
    \[ v_{t,\Pi}^\gamma(s) = \mathbb{E}_{\Pi}[G^\gamma_t | S_t = s] = \sum_{t' = t+1}^{\infty} \gamma^{t'-(t+1)} \mathbb{E}_{\Pi}[R_{t'} | S_t = s] \]
  - Average return setting:
    \[ \bar{v}_{t,\Pi}(s) = \mathbb{E}_{\Pi}[\bar{G}_t | S_t = s] = \lim_{T \to \infty} \frac{1}{T} \sum_{t' = t+1}^{t+T} \mathbb{E}_{\Pi}[R_{t'} | S_t = s] \]

- Depends on $t$ for a history dependent policy!
Returns and Value Functions

State Value Functions

- Return expectation for a Markovian policy $\Pi$ starting from $s$ at time $t$.
  - Finite horizon setting (with time extended state space):
    \[ v_{t,\Pi}^T(s) = \mathbb{E}_\Pi [G_t^T | S_t = s] = \sum_{t' = t+1}^{T} \mathbb{E}_\Pi [R_{t'} | S_t = s] \]
  - Episodic setting:
    \[ v_{t,\Pi}(s) = \mathbb{E}_\Pi [G_t | S_t = s] = \sum_{t' = t+1}^{\infty} \mathbb{E}_\Pi [R_{t'} | S_t = s] \]
  - Discounted:
    \[ v_{t,\Pi}^\gamma(s) = \mathbb{E}_\Pi [G_t^\gamma | S_t = s] = \sum_{t' = t+1}^{\infty} \gamma^{t'-(t+1)} \mathbb{E}_\Pi [R_{t'} | S_t = s] \]
  - Average return setting:
    \[ \bar{v}_{t,\Pi}(s) = \mathbb{E}_\Pi [\bar{G}_t | S_t = s] = \lim_{T \to \infty} \frac{1}{T} \sum_{t' = t+1}^{t+T} \mathbb{E}_\Pi [R_{t'} | S_t = s] \]

- Becomes independent on $t$ if the policy is stationary and Markovian the generic case (except in the finite horizon setting).
State-Action Value Functions

State Value Functions

- Return expectation for a policy $\Pi$ starting from $s$ and an action $a$ at time $t$.
  - Finite horizon setting:
    \[ q^T_{t,\Pi}(s, a) = \mathbb{E}_\Pi [G^T_t | S_t = s, A_t = a] = \sum_{t'=t+1}^T \mathbb{E}_\Pi [R_{t'} | S_t = s, A_t = a] \]
  - Episodic setting:
    \[ q_{t,\Pi}(s, a) = \mathbb{E}_\Pi [G_t | S_t = s, A_t = a] = \sum_{t'=t+1}^\infty \mathbb{E}_\Pi [R_{t'} | S_t = s, A_t = a] \]
  - Discounted:
    \[ q^\gamma_{t,\Pi}(s, a) = \mathbb{E}_\Pi [G^\gamma_t | S_t = s, A_t = a] = \sum_{t'=t+1}^\infty \gamma^{t'-t-1} \mathbb{E}_\Pi [R_{t'} | S_t = s, A_t = a] \]
  - Average return setting:
    \[ \bar{q}_{t,\Pi}(s, a) = \mathbb{E}_\Pi [\bar{G}_t | S_t = s, A_t = a] = \lim_{T \to \infty} \frac{1}{T} \sum_{t'=t+1}^{t+T} \mathbb{E}_\Pi [R_{t'} | S_t = s, A_t = a] \]

- Different strategy for action at time $t$ than after...
- Independent of $t$ for a Markovian policy except for the finite horizon setting!
State vs State-Action

- Performance measure of a policy $\Pi$:
  - starting from a state $s$ for the state value function,
  - starting from a state $s$ and an action $a$ (not necessarily related to $\Pi$) for the state-action value function.

- State value function at time $t$ from state-action value function:
  $$v_{t,\Pi}(s) = \sum_a \Pi_t(a) q_{t}(s, a)$$
Do We Really Need The History Dependent Policies?

Equivalent Markovian policy in terms of value function

- **Thm:** For any policy $\Pi$ and any initial distribution $P_0(S_0)$, it exists a Markovian policy $\tilde{\Pi}$ such that

$$\forall t, \forall s, v_{t,\Pi}(s) = v_{t,\tilde{\Pi}}(s).$$

- Relies on the Markovian environment.
- Possible choice:

$$\tilde{\pi}_t \{ A_t = a_t | S_t = s_t \} = \mathbb{E}_{P,P_0}[\pi_t(A_t = a_t | S_t = s_t, H_t) | S_t = s_t, S_0]$$

- **No need to consider non Markovian policy** if the goal is entirely defined in terms of value functions.
Outline

1. Decision Process and Markov Decision Process
2. Returns and Value Functions
3. Prediction and Planning
4. Operations Research and Reinforcement Learning
5. Control
6. Survey
7. References
Prediction

What is the performance of a given policy?

Planning

What is the best policy?

Planning is harder than predicting.
What is the performance of a given policy?

Compute/Approximate/Estimate

\[ v_{t,n}(s) = \mathbb{E}_n[G_t | S_t = s] \]

Well defined provided the expectation exists.
Planning

What is the best policy?

A possible definition: \( \text{argmax} \sum_{s,t} \mu(s, t) v_{t, \pi}(s) \)

Not necessarily well defined...

Several choices for \( \mu \)!

More realistic goal: find a good policy...
Outline

1. Decision Process and Markov Decision Process
2. Returns and Value Functions
3. Prediction and Planning
4. Operations Research and Reinforcement Learning
5. Control
6. Survey
7. References
What Do We Know?

Model
- Able to use the MDP transition probabilities.
- Probability world.

Only Observations
- No access to the MDP transition probabilities.
- Reinforcement Learning.
- Statistic world.

- Reinforcement Learning is harder than Markov Decision Process / Operations Research.
Markov Decision Process / Operations Research

MDP / OR

- Stochastic setting in which the world is known.
- MDP model assumption.
- Probability world / Idealized setting...
- Lots of insight for the RL problem.
Reinforcement Learning

RL

- Stochastic setting in which the world is observed through interactions.
- Still MDP model assumption.
- More realistic setting?
- More difficult theoretical analysis.
Outline

1. Decision Process and Markov Decision Process
2. Returns and Value Functions
3. Prediction and Planning
4. Operations Research and Reinforcement Learning
5. Control
6. Survey
7. References
MDP vs Discrete Control

**MDP**
- State $s$ and action $a$
- Dynamic model:
  \[ P(s'|s, a) \]
- Reward $r$ defined by $P(r|s', s, a)$.
- Policy $\Pi$: $a_t = \pi_t(S_t, H_t)$
- Goal:
  \[ \max \mathbb{E}_\Pi \left[ \sum_t R_t \right] \]

**Discrete Control**
- State $x$ and control $u$
- Dynamic model:
  \[ x' = f(x, u, W) \]
  with $W$ a stochastic perturbation.
- Cost: $C(x, u, W)$.
- Control strategy $U$: $u_t = u(x_t, H_t)$
- Goal:
  \[ \min_U \mathbb{E}_U \left[ \sum_t C(x_t, u_t, W_t) \right] \]

- Almost the same setting but with a different vocabulary!
Outline

1. Decision Process and Markov Decision Process
2. Returns and Value Functions
3. Prediction and Planning
4. Operations Research and Reinforcement Learning
5. Control
6. Survey
7. References
RL: What Are We Going To See?

Outline

- Operations Research and MDP.
- Reinforcement learning and interactions.
- More tabular reinforcement learning.
- Reinforcement and approximation of value functions.
- Actor/Critic: a Policy Point of View
How to find the best policy knowing the MDP?

- Is there an optimal policy?
- How to estimate it numerically?

- Finite states/actions space assumption (tabular setting).
- Focus on interative methods using value functions (dynamic programming).
- Policy deduced by a statewise optimization of the value function over the actions.
- Focus on the discounted setting.
How to find the best policy not knowing the MDP?

- How to interact with the environment to learn a good policy?
- Can we use a Monte Carlo strategy outside the episodic setting?
- How to update value functions after each interaction?

- Focus on stochastic methods using tabular value functions (Q learning, SARSA...)
- Policy deduced by a statewise optimization of the value function over the actions.
More Tabular Reinforcement Learning

Can We Do Better?

- Is there a gain to wait more than one step before updating?
- Can we interact with a different policy than the one we are estimating?
- Can we use an estimated model to plan?
- Can we plan in real time instead of having to do it beforehand?

- Finite states/actions space setting (tabular setting).
Reinforcement and Approximation of Value Functions

How to Deal with a Large/Infinite states/action space?

- How to approximate value functions?
- How to estimate good approximation of value functions?

- Finite action space setting.
- Stochastic algorithm (Deep Q Learning...).
- Policy deduced by a statewise optimization of the value function over the actions.
Could We Directly Parameterized the Policy?

- How to parameterize a policy?
- How to optimize this policy?
- Can we combine parametric policy and approximated value function?

- State Of The Art Algorithms (DPG, PPO, SAC...)

Survey
Outline

1. Decision Process and Markov Decision Process
2. Returns and Value Functions
3. Prediction and Planning
4. Operations Research and Reinforcement Learning
5. Control
6. Survey
7. References
References

R. Sutton and A. Barto. 
*Reinforcement Learning, an Introduction (2nd ed.)*
MIT Press, 2018

O. Sigaud and O. Buffet. 
*Markov Decision Processes in Artificial Intelligence.*
Wiley, 2010

M. Puterman. 
Wiley, 2005

D. Bertsekas and J. Tsitsiklis. 
*Neuro-Dynamic Programming.*
Athena Scientific, 1996

W. Powell. 
Wiley, 2022

S. Meyn. 
*Control Systems and Reinforcement Learning.*
Cambridge University Press, 2022

V. Borkar. 
*Stochastic Approximation: A Dynamical Systems Viewpoint.*
Springer, 2008

T. Lattimore and Cs. Szepesvári. 
*Bandit Algorithms.*
Cambridge University Press, 2020
Licence and Contributors

Creative Commons Attribution-ShareAlike (CC BY-SA 4.0)

- You are free to:
  - **Share:** copy and redistribute the material in any medium or format
  - **Adapt:** remix, transform, and build upon the material for any purpose, even commercially.

- Under the following terms:
  - **Attribution:** You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
  - **ShareAlike:** If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
  - **No additional restrictions:** You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Contributors

- **Main contributor:** E. Le Pennec