Reinforcement Learning
Sequential Decisions, MDP and Policies

Erwan Le Pennec
Erwan.Le-Pennec@polytechnique.edu

ÉCOLE POLYTECHNIQUE

M2DS - Reinforcement Learning – Fall 2024

Outline

Decision or Decisions

Source: W. Powell

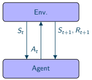
Sequential Decision Setting

Sequential Decision Setting

- 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 decisions.
- Different framework than supervised learning (no immediate feedback here) and unsupervised learning (well defined goal here).

Source: W. Powell

From Sequential Decision to Reinforcement Learning



Sequential Decision

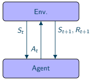
Sequential Decision

- Sequence of action A_t as a response of an environment defined by a state S_t
- Feedback through a reward R_t

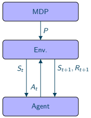
Actions?

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

From Sequential Decision to Reinforcement Learning



Sequential Decision



MDP Modeling

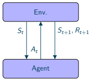
Markov Decision Process Modeling

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

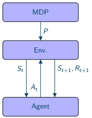
Actions?

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

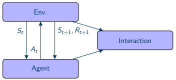
From Sequential Decision to Reinforcement Learning



Sequential Decision



MDP Modeling



Reinforcement Learning

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

Sequential Decisions

- MDP / Reinforcement Learning:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_t R_t\right]$$

- Optimal Control:

$$\min_{u} \mathbb{E}\left[\sum_t C(x_t, u_t)\right]$$

Related settings...

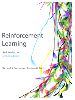
- (Stochastic) Search:

$$\max_{\theta} \mathbb{E}[F(\theta, W)]$$

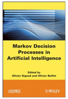
- Online Regret:

$$\max \sum_k \mathbb{E}[F(\theta_k, W)]$$

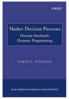
References



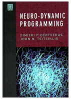
R. Sutton and A. Barto.
Reinforcement Learning, an Introduction (2nd ed.)
MIT Press, 2018



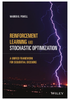
O. Sigaud and O. Buffet.
Markov Decision Processes in Artificial Intelligence.
Wiley, 2010



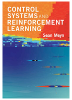
M. Puterman.
Markov Decision Processes. Discrete Stochastic Dynamic Programming.
Wiley, 2005



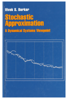
D. Bertsekas and J. Tsitsiklis.
Neuro-Dynamic Programming.
Athena Scientific, 1996



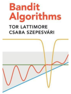
W. Powell.
Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions.
Wiley, 2022



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



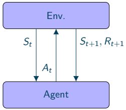
V. Borkar.
Stochastic Approximation: A Dynamical Systems Viewpoint.
Springer, 2008



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

Outline

Environnement

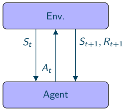


Decision Process and Environment

- At time step $t \in \mathbb{N}$:
  - State $S_t \in \mathcal{S}$: 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. $\mathcal{S}$ finite, $\mathcal{A}(s)$ finite and $\mathcal{R}$ finite.
- Extension: Non finite bounded $\mathcal{R}$: easy / Non finite $\mathcal{S}$: hard / Non finite $\mathcal{A}$: harder.

Environnement



Stochastic Model

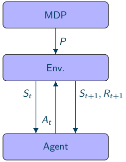
- Dynamic defined by:

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

where $H_t = (R_t, S_{t-1}, A_{t-1}, R_{t-1}, S_{t-2}, \dots)$ is the past and (S_t, A_t) the present.

$(s', r) = F(S_t = s, A_t = a, H_t, w)$ ← random part

Markov Decision Process and Environment



Markovian Environment

- Markovian Dynamic Assumption: S_{t+1} and R_{t+1} are independent of the past $H_t = (R_t, S_{t-1}, A_{t-1}, R_{t-1}, S_{t-2}, \dots)$ conditionally to the present (S_t, A_t).
- Dynamic entirely defined by state-reward transition probabilities

$$\begin{aligned}\mathbb{P}(S_{t+1} = s', R_{t+1} = r \big| S_t = s, A_t = a, H_t) &= \mathbb{P}(S_{t+1} = s', R_{t+1} = r \big| S_t = s, A_t = a) \\ &= p(s', r \big| s, a)\end{aligned}$$

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:

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

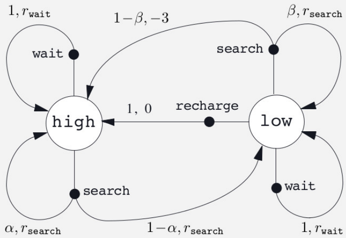
- Expected reward for a given state-action:

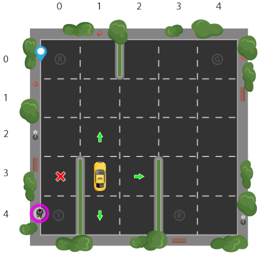
$$\mathbb{E}[R_{t+1} | S_t = s, A_t = a, H_t] = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$$
$$= r(s, a) = \sum_r r \sum_{s'} p(s', r | r, a)$$

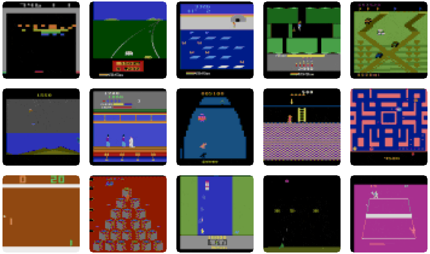
- From now on, we will always assume that the Markovian property holds for the environment.

Examples

s	a	s'	$p(s' \mid s, a)$	$r(s, a, s')$
high	search	high	α	r_{search}
high	search	low	$1-\alpha$	r_{search}
low	search	high	$1-\beta$	-3
low	search	low	β	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	-

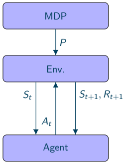






Sources:: Sutton and Barto / learndatasci.com / danijar.com

Decision Process, Agent and Policy



Agent

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

Policy Π : specification of how to choose the action

- General stochastic policy $\Pi = (\pi_0, \pi_1, \ldots, \pi_t, \ldots)$:

$$\Pi_t(A_t = a) = \pi_t(A_t = a | S_t = s, A_t = a, H_t)$$

- General deterministic policy $\Pi = (\pi_0, \pi_1, \ldots, \pi_t, \ldots)$ (with as slight abuse of notation):

$$\Pi_t(A_t = a) = \mathbf{1}_{A_t = \pi_t(S_t = s, A_t = a, H_t)}$$

Markov Decision Process, Agent and Policy

Agent

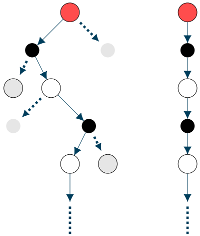
- Interact with the environment by choose the action given the past.

Policy Π: 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)

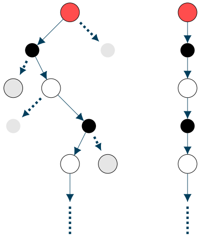
Decision Process and Trajectories



Trajectories

- Trajectory $(S_0, A_0, R_1, S_1, A_1, \ldots)$ defined by
  - an initial distribution $\mathbb{P}_0$ 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
$$\mathbb{P}(S_{t+1}, R_{t+1} | S_t, A_t, H_t)$$

Decision Process and Trajectories

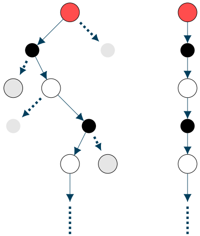


Trajectories

- Induced probability:

$$
\begin{aligned}
\mathbb{P}(S_0 = s_0, A_0 = a_0, R_1 = r_1, S_1 = s_1, A_1 = a_1, \ldots S_t = s_t, R_t = r_t)\\
= \mathbb{P}_0(S_0 = s_0)\\
\times \pi_0(A_0 = a_0 | S_0)\, \mathbb{P}(S_1, R_1 | S_0, A_0)\, \pi_1(A_1 = a_1 | S_1 = s_1, H_1)\\
\times \cdots \times \mathbb{P}(S_t = s_t, R_t = r_t | S_{t-1} = s_{t-1}, A_{t-1} = a_{n-1}, H_{t-1})
\end{aligned}
$$

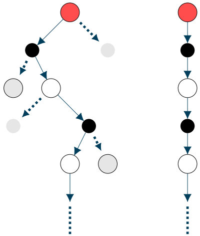
Markov Decision Process and Trajectories



Trajectories

- Trajectory $(S_0, A_0, R_1, S_1, A_1, \ldots)$ defined by
  - an initial distribution $\mathbb{P}_0$ 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
$$\mathbb{P}(S_{t+1}, R_{t+1} | S_t, A_t)$$

Markov Decision Process and Trajectories

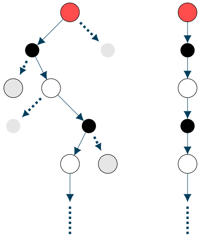


Trajectories

- Induced probability:

$$\begin{aligned}\mathbb{P}(S_0 = s_0, A_0 = a_0, R_1 = r_1, S_1 = s_1, A_1 = a_1, \dots S_t = s_t, R_t = r_t) \\ = \mathbb{P}_0(S_0 = s_0) \\ \times \pi_0(A_0 = a_0 | S_0)\, \mathbb{P}(S_1, R_1 | S_0, A_0)\, \pi_1(A_1 = a_1 | S_1 = s_1, H_1) \\ \times \cdots \times \mathbb{P}(S_t = s_t, R_t = r_t | S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1})\end{aligned}$$

Markov Decision Process and Trajectories



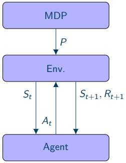
Markovian Trajectories only if the policy is Markovian

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

- Stationary if the policy is stationary.

Outline

Rewards and Total Return



Rewards and Total Returns

- MDP: Rewards R_t encode all the feedbacks!
- Quality of a policy Π measured from the remaining total return:

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

- Expected total return following Π starting from s:

$$\mathbb{E}_{\Pi}[G_t | S_t = s] = \sum_{t'=t+1}^{\infty} \mathbb{E}_{\Pi}[R_{t'} | S_t = s]$$

Total Return: Issue and Fixes

Issues

- G_t is a limiting process and thus may not be defined!
- Can diverge to $\pm\infty$ or 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 $\forall t' \geq T, R_{t'} = 0$ and $\mathbb{E}[T] < \infty$ so that $G_t = \sum_{t'=t+1}^{\infty} R_{t'}$ is well defined.
- Discounted setting: for $0 < \gamma < 1$, $G_t^\gamma = \sum_{t'=t+1}^{\infty} \gamma^{t'-(t+1)} R_{t'}$
- Average return: $\overline{G}_t = \lim \frac{1}{T} \sum_{t'=t+1}^{t+T} R_{t'}$

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_{abs} (a state that cannot be escaped and from which the reward is always 0).
  - Extend the state space $\mathcal{S}$ to $(\mathcal{S} \times \{0, \dots, T\}) \cup \{s_{\text{abs}}\}$.
  - Define an state reward transition probability:

$$p\left(\tilde{s}', r|\tilde{s}, a\right) = \begin{cases} p(s', t|s, a) & \text{if } \tilde{s} = (s, t),\ t < T \text{ and } \tilde{s'} = (s', t+1) \\ 1 & \text{if } \tilde{s} = (s, t),\ t = T,\ \tilde{s'} = s_{\text{abs}} \text{ and } r = 0 \\ 1 & \text{if } \tilde{s} = s_{\text{abs}}\ ,\ \tilde{s'} = s_{\text{abs}} \text{ and } r = 0 \\ 0 & \text{otherwise} \end{cases}$$

Episodic Setting

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

Episodic Setting

- Assumption: for any policy Π, the average duration before R_t remains equal to 0 is smaller than a finite horizon H: $$\mathbb{E}_\Pi\left[\min_{t, R_{t'}=0, \forall t' \geq t} t\right] \leq H < +\infty$$
- Strong assumption...
- Easy to interpret.

- Slightly stronger (but more convenient) 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 Π and any initial state, the average duration to reach this state is smaller than a finite horizon H: $$\forall s, \mathbb{E}_\Pi\left[\min_{t, S_t=s_{\text{abs}}} t \middle| S_0 = s\right] \leq H < +\infty$$

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:

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

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

Average Return Setting

$$\overline{G}_t = \lim \frac{1}{T} \sum_{t'=t+1}^{t+T} R_{t'}$$

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

$$\mathbb{E}_\Pi[G_t^\gamma] = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_\Pi[R_{t+1}] = \frac{1}{1-\gamma}\mathbb{E}_\Pi[R_t] = \frac{1}{1-\gamma}\mathbb{E}_\Pi\left[\overline{G}_t\right]$$

State Value Functions

State Value Functions

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

- Depends on t for a history dependent policy!

Markovian Policy and State Value Functions

State Value Functions

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

- 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-Action Value Functions

- Return expectation for a policy Π starting from s and an action a at time t.
  - Finite horizon setting:
  $$q^T_{t,\Pi}(s,a) = \mathbb{E}_\Pi\left[G^T_t | S_t = s, A_t = a\right] = \sum_{t'=t+1}^{T} \mathbb{E}_\Pi[R_{t'} | S_t = s, A_t = a]$$
  - 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:
  $$\overline{q}_{t,\Pi}(s,a) = \mathbb{E}_\Pi\left[\overline{G}_t | S_t = s, A_t = a\right] = \lim_{T\to\infty} \frac{1}{T} \sum_{t'=t+1}^{t+T} \mathbb{E}_\Pi[R_{t'} | S_t = s, A_t = a]$$

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

State Value Function vs State-Action Value Functions

$$v_{t,\Pi}(s) = \mathbb{E}_{\Pi}[G_t | S_t = s] \quad q_{t,\Pi}(s,a) = \mathbb{E}_{\Pi}[G_t | S_t = s, A_t = a]$$

State vs State-Action

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

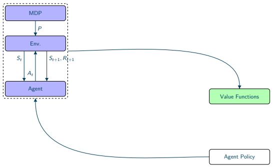
Equivalent Markovian policy in terms of value function

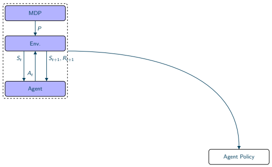
- **Thm:** For any policy Π and any initial distribution $\mathbb{P}_0(S_0)$, it exists a Markovian policy $\widetilde{\Pi}$ such that
$$\forall t, \forall s, v_{t,\Pi}(s) = v_{t,\widetilde{\Pi}}(s).$$

- Relies on the Markovian environment.
- Possible choice:
$$\widetilde{\pi}_t \left\{A_t = a_t | S_t = s_t\right\} = \mathbb{E}_{\mathbb{P},\mathbb{P}_0}[\pi_t(A_t = a_t | S_t = s_t, H_t) | S_t = s_t, S_0]$$
- **No need to consider non Markovian policy** if the goal is entirely defined in terms of value functions.

Outline

Goals





Prediction

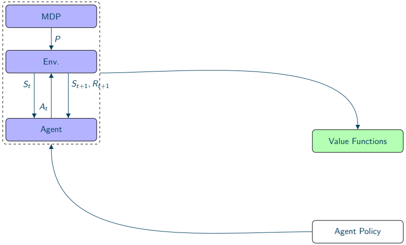
- What is the performance of a given policy?

Planning

- What is the *best* policy?

- Planning is harder than predicting.

Prediction



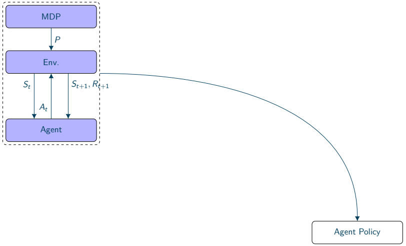
Prediction

- What is the performance of a given policy?
- Compute/Approximate/Estimate

$$v_{t,\Pi}(s) = \mathbb{E}_\Pi[G_t|S_t = s]$$

- Well defined provided the expectation exists.

Planning

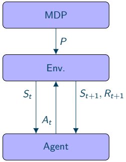


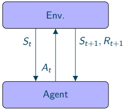
Planning

- What is the *best* policy?
- A possible definition: $\underset{\Pi}{\operatorname{argmax}} \sum_{s,t} \mu(s,t) v_{t,\Pi}(s)$
- Not necessarily well defined...
- Several choices for μ!
- More realistic goal: find a *good* policy...

Outline

What Do We Know?





Model

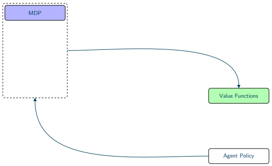
- Able to use the MDP transition probabilities.
- Markov Decision Process / Operations Research.
- Probability world.

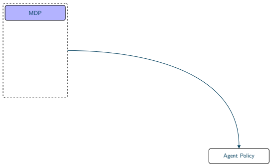
Only Observations

- 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

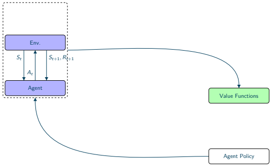


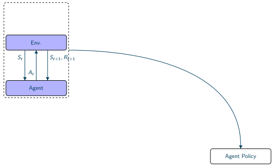


MDP / OR

- 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

MDP vs Discrete Control

MDP

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

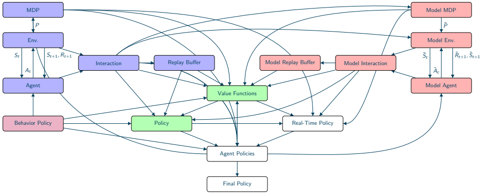
Discrete Control

- 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

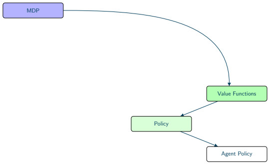
RL: What Are We Going To See?



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

Operations Research and MDP

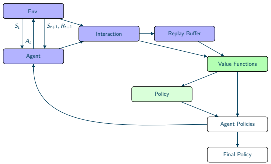


How to find the best policy knowing the MDP?

- 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.

Reinforcement Learning and Interactions

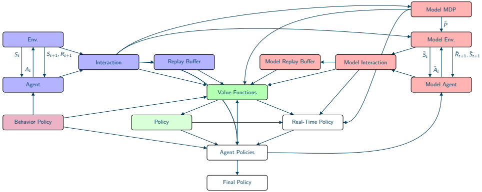


How to find the best policy not knowing the MDP?

- 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.

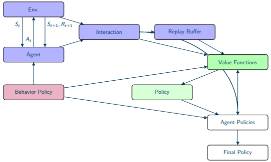
More Tabular Reinforcement Learning



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).

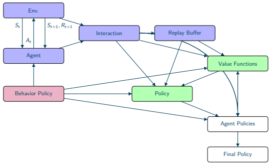


How to Deal with a Large/Infinite states/action space?

- 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.

Actor/Critic: a Policy Point of View



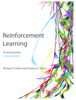
Could We Directly Parameterized the Policy?

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

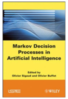
- State Of The Art Algorithms (DPG,PPO, SAC...)

Outline

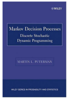
References



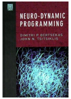
R. Sutton and A. Barto.
Reinforcement Learning, an Introduction (2nd ed.)
MIT Press, 2018



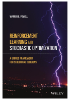
O. Sigaud and O. Buffet.
Markov Decision Processes in Artificial Intelligence.
Wiley, 2010



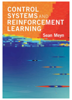
M. Puterman.
Markov Decision Processes. Discrete Stochastic Dynamic Programming.
Wiley, 2005



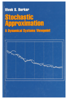
D. Bertsekas and J. Tsitsiklis.
Neuro-Dynamic Programming.
Athena Scientific, 1996



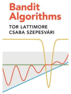
W. Powell.
Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions.
Wiley, 2022



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

Contributors

- Main contributor: E. Le Pennec
- Contributors: S. Boucheron, A. Dieuleveut, A.K. Fermin, S. Gadat, S. Gaiffas, A. Guilloux, Ch. Keribin, E. Matzner, M. Sangnier, E. Scornet.