Lecture 1: Introduction

(CSC2547: Introduction to Reinforcement Learning)

Amir-massoud Farahmand

University of Toronto & Vector Institute

Table of Contents

Reinforcement Learning

Figure: An agent ...

Reinforcement Learning

Figure: ... observes the world ...

Reinforcement Learning

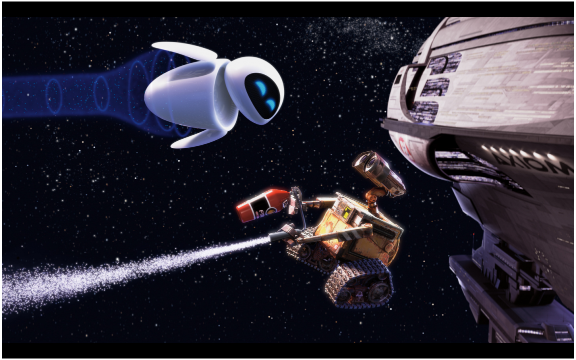


Figure: ... takes an action and its states changes ...

Reinforcement Learning

Figure: ... with the goal of achieving long-term rewards.

Reinforcement Learning in the News

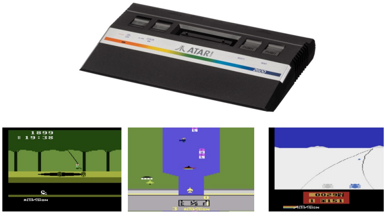


Figure: Some recent success stories!

(Potential) Applications of RL

Figure: And a lot of potential applications

This course ...

This course is about reinforcement learning (RL) and sequential decision-making under uncertainty with an emphasis on theoretical understanding.

We build the foundation, step by step, prove many results, and try to gain an understanding of why many algorithms are designed the way they are, and why they work.

Reinforcement Learning: Problem and Methods

Reinforcement Learning (RL) refers to both a type of problem and a set of computational methods.

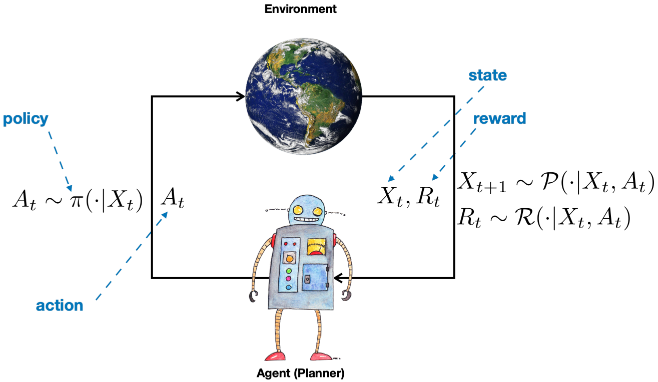
- **Problem:** How to act so that some notion of long-term performance is maximized?
- **Methods:** What kind of computation does an agent need to do in order to ensure that its actions lead to good (or even optimal) long-term performance?

Remark

Historically, only a subset of all computational methods that attempt to solve the RL problem are known as the RL methods, e.g., Q-Learning is, evolutionary computation methods are not.

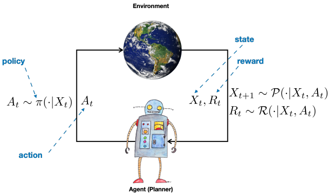
RL Problem and Agent-Environment Interaction

In RL, we often talk about an agent and its environment, and their interaction.



RL Problem and Agent-Environment Interaction

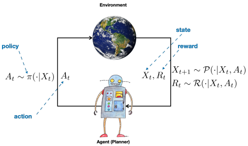
- Agent: decision maker and/or learner
  - robot
  - medical diagnosis and treatment system
  - air conditioning system
- Environment: anything outside the agent with which it interacts and attempts to control.
  - physical world outside the robot
  - patient's body
  - room



RL Problem and Agent-Environment Interaction

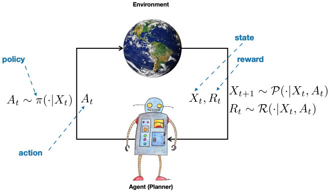
At time $t = 1, 2, \dots,$ the interaction of the agent and the environment is as follows:

- the agent observes its state X_t in the environment.
  - Examples: position of the robot, vital information of a patient, the room temperature, etc.
- The agent picks an action A_t according to its policy π, e.g., $A_t = \pi(X_t)$ or $A_t \sim \pi(\cdot|X_t)$.
- The state of the agent in the environment changes and becomes X_{t+1} according to transition probability kernel (or distribution), i.e., $X_{t+1} \sim \mathcal{P}(\cdot|X_t, A_t)$.
- The agent also receives a reward signal R_t, i.e., $R_t \sim \mathcal{R}(\cdot|X_t, A_t)$.



RL Problem and Agent-Environment Interaction

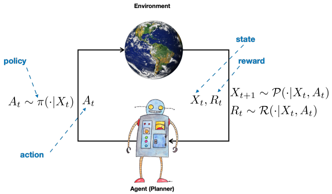
State: A variable that summarizes whatever has happened to the agent so far.
Policy: Action selection mechanism. Usually a mapping from states to actions. It can be deterministic ($A_t = \pi(X_t)$) or stochastic ($A_t \sim \pi(\cdot|X_t)$).
Transition probability kernel: Describes the dynamics. For example, a set of electromechanical equations describing how the position of the robot (including its joints) change when a certain command is sent to its motor. Or how the patient's physiology changes after the administration of the treatment.



RL Problem and Agent-Environment Interaction

Reward: A real number specifying the *immediate* desirability of the choice of action A_t at the state X_t (possibly leading to state X_{t+1}) has been. Examples:

- Positive if robot successfully picks up an object, negative if it breaks the object.
- Infection subsides
- The room temperature becomes comfortable.



Remark

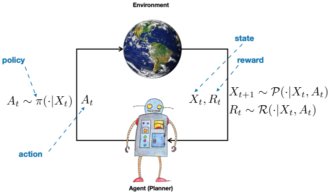
The reward signal/function/distribution only encodes the desirability of the action from the immediate perspective. A good action now may not be good in the long-term.

RL Problem and Agent-Environment Interaction

This process repeats and as a result, the agent receives a sequence of state, actions, and rewards:

$$X_1, A_1, R_1, X_2, A_2, R_2, \cdots .$$

This sequence might terminate after a fixed number of time steps (say, T), or until the agent gets to a certain region of the state space, or it might continue forever.



Markov Decision Process (MDP)

Let us formally define some important concepts that we require throughout the course. Beforehand, some commonly used notations:

Given a space Ω.

- $\mathcal{M}(\Omega)$: the space of all probability distributions defined over the space Ω.
- $B(\Omega)$: the space of all bounded functions defined over Ω

Examples: $\Omega = \{1, 2, \dots, n\}, \mathbb{N}, \mathbb{R}, \mathbb{R}^d$, etc.

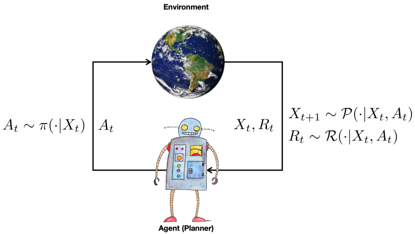
$\mathcal{M}(\mathbb{R})$: The space of distributions on the real line

$B(\mathbb{R})$: The space of bounded functions on the real line

Markov Decision Process (MDP)

Definition

A *discounted MDP* is a 5-tuple $(\mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{X}$ is a measurable state space, $\mathcal{A}$ is the action space,
$\mathcal{P} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X})$ is the transition probability kernel with domain $\mathcal{X} \times \mathcal{A}$,
$\mathcal{R} : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathbb{R})$ is the immediate reward distribution, and $0 \leq \gamma < 1$ is the discount factor.



Markov Decision Process (MDP)

MDPs encode the temporal evolution of a discrete-time stochastic process controlled by an *agent*.

- Initial state $X_1 \sim \rho$ with $\rho \in \mathcal{M}(\mathcal{X})$.
- Agent chooses action $A_t \in \mathcal{A}$.
- Agent goes to $X_{t+1} \sim \mathcal{P}(\cdot|X_t, A_t)$ and receives reward $R_t \sim \mathcal{R}(\cdot|X_t, A_t)$.
- The process repeats. The trajectory is $\xi = (X_1, A_1, R_1, X_2, A_2, R_2, \cdots)$, which is random.

Remark

The reward distribution could also depend on the next-state X_{t+1}. In that case, we would have a different reward kernel $\mathcal{R}'$ and the reward would be $R_t \sim \mathcal{R}'(\cdot|X_t, A_t, X_{t+1})$. But we can absorb the dynamics within $\mathcal{R}$, i.e., $\mathcal{R}(\cdot|x, a) = \int \mathcal{R}'(\cdot|x, a, x')\mathcal{P}(\mathrm{d}x'|x, a)$.

Markov Decision Process (MDP)

This is a general framework.

State space:

- Finite: $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ (or $\mathcal{X} = \{1, 2, \ldots, n\}$) with $n < \infty$
- Infinite but countable: $\mathcal{X} = \{x_1, x_2, \ldots\}$ (or $\mathcal{X} = \mathbb{N}$)
- Continuous: $\mathcal{X} \subset \mathbb{R}^d$

Dynamics:

- Stochastic
- Deterministic

Remark

A deterministic dynamical system always behave exactly the same given the same starting state and action. They can be described by the transition function $f : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$ *and* $x_{t+1} = f(x_t, a_t)$.

Policy

Definition

A policy is a sequence $\bar{\pi} = \{\pi_1, \pi_2, \ldots\}$ such that for each t,

$$\pi_t(a_t | X_1, A_1, X_2, A_2, \ldots, X_{t-1}, A_{t-1}, X_t)$$

is a stochastic kernel on $\mathcal{A}$ given $\underbrace{\mathcal{X} \times \mathcal{A} \times \cdots \times \mathcal{X} \times \mathcal{A} \times \mathcal{X}}_{2t-1 \text{ elements}}$

satisfying

$$\pi_t(\mathcal{A} | X_1, A_1, X_2, A_2, \ldots, X_{t-1}, A_{t-1}, X_t) = 1$$

for every $(X_1, A_1, X_2, A_2, \ldots, X_{t-1}, A_{t-1}, X_t)$.

Policy

Definition

If π_t is parametrized only by X_t, that is

$$\pi_t(\cdot|X_1, A_1, X_2, A_2, \ldots, X_{t-1}, A_{t-1}, X_t) = \pi_t(\cdot|X_t),$$

$\bar{\pi}$ is a Markov policy.

If for each t and $(X_1, A_1, X_2, A_2, \ldots, X_{t-1}, A_{t-1}, X_t)$, the policy π_t assigns mass one to a single point in $\mathcal{A}$, $\bar{\pi}$ is called a *deterministic (nonrandomized)* policy; if it assigns a distribution over $\mathcal{A}$, it is called *stochastic* or *randomized* policy.

If $\bar{\pi}$ is a Markov policy in the form of $\bar{\pi} = (\pi, \pi, \ldots)$, it is called a stationary policy.

A policy $\pi(\cdot|x)$ is a stationary Markov policy. We often work with such policies. If it is also deterministic, we denote it by $\pi(x)$.

Policy-Induced Transition Kernels

An agent is "following" a Markov stationary policy π whenever A_t is selected according to the policy $\pi(\cdot|X_t)$, i.e., $A_t = \pi(X_t)$ (deterministic) or $A_t \sim \pi(\cdot|X_t)$ (stochastic).
The policy π induces two transition probability kernels $\mathcal{P}^\pi : \mathcal{X} \to \mathcal{M}(\mathcal{X})$ and $\mathcal{P}^\pi : \mathcal{X} \times \mathcal{A} \to \mathcal{M}(\mathcal{X} \times \mathcal{A})$. For a (measurable) subset A of $\mathcal{X}$ and a (measurable) subset B of $\mathcal{X} \times \mathcal{A}$ and a deterministic policy π, denote

$$(\mathcal{P}^\pi)(A|x) \triangleq \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}y|x, \pi(x))\mathbb{I}_{\{y \in A\}},$$

$$(\mathcal{P}^\pi)(B|x, a) \triangleq \int_{\mathcal{X}} \mathcal{P}(\mathrm{d}y|x, a)\mathbb{I}_{\{(y, \pi(y)) \in B\}}.$$

Policy-Induced Transition Kernels

When we have a countable state-action space, we sometimes use summation instead of integrals. For example,

$$(\mathcal{P}^{\pi})(A|x) \triangleq \sum_{y \in \mathcal{X}} \mathcal{P}(y|x, \pi(x)) \mathbb{I}_{\{y \in A\}} = \sum_{y \in A} \mathcal{P}(y|x, \pi(x)).$$

So for a particular $y \in \mathcal{X}$, we have $(\mathcal{P}^{\pi})(y|x) = \mathcal{P}(y|x, \pi(x))$. Also we can extend the definition of $\mathcal{P}^{\pi}$ to following a policy for m-steps $(m \geq 1)$ inductively. We use $(\mathcal{P}^{\pi})^m$ to denote such a transition kernel.

From Immediate to Long-Term Reward

RL problem: How to act so that some notion of long-term performance is maximized.

Q: What does long-term mean? How to quantify it?

Immediate Reward Problem

- At each round of interaction with its environment
  - An agent starts at a random state $X_1 \sim \rho \in \mathcal{M}(\mathcal{X})$
  - It chooses action $A_1 = \pi(X_1)$ (deterministic policy), and receives a reward of $R_1 \sim \mathcal{R}(\cdot|X_1, A_1)$.

We call each of these rounds an episode. Here the episode only lasts one time-step.

Q: How should this agent choose its policy in order to maximize its "performance"?

Q: What does performance mean?

Immediate Reward Problem

- At each round of interaction with its environment
  - An agent starts at a random state $X_1 \sim \rho \in \mathcal{M}(\mathcal{X})$
  - It chooses action $A_1 = \pi(X_1)$ (deterministic policy), and receives a reward of $R_1 \sim \mathcal{R}(\cdot|X_1, A_1)$.

We can talk about average (expected) reward that the agent receives within one episode as the measure of performance.
Average is over repeated interactions with the environment.
If we define the performance in this way, answering the question of how the agent should act to maximize this notion of performance is easy.

Immediate Reward Problem

Let us define expected reward as

$$r(x, a) \triangleq \mathbb{E}\left[R | X = x, A = a\right].$$

In order to maximize the expected reward, the best action depends on the state the agent initially starts with. At state x, it should choose

$$a^* \leftarrow \operatorname*{argmax}_{a \in \mathcal{A}} r(x, a).$$

This is the best, or *optimal*, action at state x
The optimal policy $\pi^* : \mathcal{X} \rightarrow \mathcal{A}$:

$$\pi^*(x) \leftarrow \operatorname*{argmax}_{a \in \mathcal{A}} r(x, a). \quad (1)$$

Optimal policy depends on the agent's initial state. It does not depend on initial distribution ρ.

Finite Horizon Tasks

The agent interacts with the environment for a fixed $T \geq 1$ number of steps.

- At each round (episode),
  - The agent starts at $X_1 \sim \rho \in \mathcal{M}(\mathcal{X})$.
  - It chooses action $A_1 = \pi(X_1)$ (or $A_1 \sim \pi(\cdot|X_1)$ for a stochastic policy)
  - The agent goes to the next-state $X_2 \sim \mathcal{P}(\cdot|X_1, A_1)$ and receives reward $R_1 \sim \mathcal{R}(\cdot|X_1, A_1)$.
  - (this process repeats for several steps until ...)
  - $X_T \sim \mathcal{P}(\cdot|X_{T-1}, A_{T-1})$.
  - $R_T \sim \mathcal{R}(\cdot|X_{T-1}, A_{T-1})$.

So we receive a reward sequence $(R_1, R_2, \ldots, R_T)$.
Q: How should we evaluate the performance of the agent as a function of the reward sequence?

Finite Horizon Tasks

A common choice for performance is to compute the sum of rewards:

$$G^\pi \triangleq R_1 + \ldots + R_T. \tag{2}$$

The r.v. G^π is called the return of following policy π.
Here the rewards received at all time steps are treated the same.
The agent just adds them together.

Finite Horizon Tasks

Another choice is to consider *discounted* sum of rewards. Given a discount factor $0 \leq \gamma \leq 1$, we define the return as

$$G^{\pi} \triangleq R_1 + \gamma R_2 + \ldots + \gamma^{T-1} R_T. \tag{3}$$

Whenever $\gamma < 1$, the reward that is received earlier contributes more to the return. Intuitively, this means that such a definition of return values earlier rewards more.

- A cookie today is better than a cookie tomorrow, and a cookie tomorrow is better than a cookie a week later.
- Financial interpretation (inflation rate).
- Marshmallow test (delayed gratification).

Smaller values of γ makes the agent more myopic.

Remark

The discount factor is a part of the problem definition.

From Return to Value Function

The return (3) (and (2) as a special case) is a random variable. To define a performance measure that is not random, we compute its expectation.

$$V^{\pi}(x) \triangleq \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} R_t | X_1 = x\right].$$

This is the expected value of return if the agent starts at state x and follows policy π. The function $V^{\pi} : \mathcal{X} \to \mathbb{R}$ is called the value function of π.

From Return to Value Function

Let us look at the case of $T = 1$ a bit closer. The value function V^π at state x is

$$V^\pi(x) = \mathbb{E}\left[R_1 | X = x\right].$$

This is similar to $r(x,a) = \mathbb{E}\left[R | X = x, A = a\right]$ with the difference that $r(x,a)$ is conditioned on both x and a, whereas V^π is conditioned on x.

In V^π, the choice of action is determined by the policy π, i.e., at state x, $a = \pi(x)$ or $A \sim \pi(\cdot|x)$.

If we define

$$r^\pi(x) \triangleq \mathbb{E}\left[R | X = x\right]$$

with $A \sim \pi(\cdot|x)$, we get that $r^\pi = V^\pi$.

For $T > 1$, V^π captures the long-term (discounted) average of the rewards, instead of the expected immediate reward captures by r^π.

How to Get the Optimal Policy?

Let us focus on $T = 1$ again.
Recall from before that for the immediate reward maximization problem, the optimal policy was

$$\pi^*(x) \leftarrow \operatorname*{argmax}_{a \in \mathcal{A}} \mathbb{E}\left[R | X = x, A = a\right].$$

Getting the optimal policy from V^π "seems" less straightforward.

How to Get the Optimal Policy?

We need to search over the space of all deterministic or stochastic policies. If we denote the space of all stochastic policies by

$$\Pi = \{ \pi : \pi(\cdot|x) \in \mathcal{M}(\mathcal{A}), \forall x \in \mathcal{X} \}$$

we need to find

$$\pi^* \leftarrow \operatorname*{argmax}_{\pi \in \Pi} V^\pi.$$

It turns out that this problem is not too difficult when $T = 1$. As the values of V^π at two different states $x_1, x_2 \in \mathcal{X}$ do not have any interaction with each other, we find the optimal policy at each state separately.

How to Get the Optimal Policy?

For each $x \in \mathcal{X}$,

$$V^\pi(x) = \int \mathcal{R}(\mathrm{d}r|x,a)\pi(\mathrm{d}a|x) = \int \pi(\mathrm{d}a|x) r(x,a).$$

Find a $\pi(\cdot|x)$ that maximizes $V^\pi(x)$ means that

$$\sup_{\pi(\cdot|x) \in \mathcal{M}(\mathcal{A})} \int \pi(\mathrm{d}a|x) r(x,a).$$

The maximizing distribution can concentrate all its mass at the action a^* that maximizes $r(x,a)$ (assuming it exists). Therefore, $\pi^*(a|x) = \delta(a - \mathrm{argmax}_{a' \in \mathcal{A}} r(x,a'))$ (or equivalently, $\pi^*(x) = \mathrm{argmax}_{a' \in \mathcal{A}} r(x,a')$) is an optimal policy at state x. This is for $T = 1$. For $T > 1$, the problem would be more complicated.

Episodic Tasks

In some scenarios, there is a final time T that the episode ends (or terminates), but it is not fixed a priori.

- Chess
- Finding a goal within a maze
- Robot successfully picks an object

The episode terminates whenever the agent reaches a certain state x_{terminal} within the state space, i.e., it terminates whenever $X_T = x_{\text{terminal}}$. The length of the episode T is a random variable.

Episodic Tasks

The definition of the return and value functions is as before:
For $0 \leq \gamma \leq 1$, we have

$$G^{\pi} \triangleq \sum_{k=1}^{T} \gamma^{k-1} R_k,$$

and

$$V^{\pi}(x) \triangleq \mathbb{E}\left[G^{\pi} | X_1 = x\right].$$

Remark

If $\gamma < 1$, these definitions are always well-defined. If $\gamma = 1$, we need to ensure that the termination time T is finite. Otherwise, the summation might be divergent (just imagine that all R_t are equal to 1).

Continuing Tasks

Sometimes the interaction between the agent and its environment does not break into episodes that terminates. It goes on continually forever.

- Life-long robot
- Chemical plant that is supposed to work for a long time
- An approximate model for finite-horizon problem with very large T.

Continuing Tasks

Consider the sequence of rewards $(R_1, R_2, \dots)$ generated after the agent starts at state $X_1 = x$ and follows policy π. Given the discount factor $0 \le \gamma < 1$, the return is

$$G_t^{\pi} \triangleq \sum_{k \ge t} \gamma^{k-t} R_k. \tag{4}$$

Continuing Tasks

Definition (Value Functions)

The (state-)value function V^π and the action-value function Q^π for a policy π are defined as follows: Let $(R_t; t \geq 1)$ be the sequence of rewards when the process is started from a state X_1 (or (X_1, A_1) for the action-value function) drawn from a positive probability distribution over $\mathcal{X}$ (or $\mathcal{X} \times \mathcal{A}$) and follows the policy π for $t \geq 1$ (or $t \geq 2$ for the action-value function). Then,

$$V^\pi(x) \triangleq \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t | X_1 = x\right],$$

$$Q^\pi(x, a) \triangleq \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t | X_1 = x, A_1 = a\right].$$

Continuing Tasks

- The value function V^π evaluated at state x is the expected discounted return of following the policy π from state x.
- The action-value function Q^π evaluated at (x, a) is the expected discounted return when the agent starts at state x, takes action a, and then follows policy π.

Remark

If $\gamma = 0$, $Q^\pi = \mathbb{E}\left[R_1 | X_1 = x, A_1 = a\right]$. *This is the same as the expected immediate reward* $r(x, a)$. *The same way that we could easily compute the optimal action using* $r(x, a)$ *in the finite-horizon problem with* $T = 1$, *we can use* Q^π *(in continual task) in order to easily compute the optimal policy.*

Optimal Policy and Optimal Value Function

Q: What does it mean for an agent to act optimally?
First, let us think about how we can compare two (Markov stationary) policies π and π'. We say that π is better than or equal to π' (i.e., $\pi \geq \pi'$) iff $V^{\pi}(x) \geq V^{\pi'}(x)$ for all states.
Optimal policy: If we can find a policy π^* that satisfies $\pi^* \geq \pi$ for any π, we call it an optimal policy.

Remark

There may be more than one optimal policy, but their values are the same.

Optimal Policy and Optimal Value Function

If we denote Π as the space of all stationary Markov polices, this means that

$$\pi^* \leftarrow \underset{\pi \in \Pi}{\operatorname{argmax}} V^\pi,$$

where one of the maximizers is selected in an arbitrary way. The value function of this policy is the called the optimal value function, and is denoted by V^{π^*} or simply V^*.
We can also define the optimal policy based on Q^π, i.e.,

$$\pi^* \leftarrow \underset{\pi \in \Pi}{\operatorname{argmax}} Q^\pi.$$

The optimal action-value function is denoted by Q^{π^*} or Q^*.

Optimal Policy and Optimal Value Function

For the immediate reward maximization problem (finite horizon with $T = 1$), it is easy to find the optimal value function.
Recall that $\pi^*(x) \leftarrow \text{argmax}_{a \in \mathcal{A}} r(x, a)$ (1), so for any $x \in \mathcal{X}$,

$$V^*(x) = V^{\pi^*}(x) = \max_{a \in \mathcal{A}} r(x, a).$$

It is clear that for any $\pi : \mathcal{X} \to \mathcal{A}$,

$$V^*(x) = \max_{a \in \mathcal{A}} r(x, a) \geq r(x, \pi(x)).$$

The conclusion would be the same for stochastic policies.

Optimal Policy and Optimal Value Function

When we go to continual tasks (or even finite horizon with $T > 1$), we can ask several questions:

- Does any optimal policy exist? Maybe no single policy can dominate all others for all states.
  For example, it is imaginable that at best we can only hope to find a π^* that is better than any other policy π only on a proper subset of $\mathcal{X}$.
- Is the optimal policy necessarily a stationary policy?
  Isn't it possible to have a policy $\bar{\pi} = \{\pi_1, \pi_2, \ldots\}$ that depends on the time step and acts better than any stationary policy $\bar{\pi} = \{\pi, \pi, \ldots\}$?
- More pragmatic question: How can we find an optimal policy (if it exists)?

Optimal Policy and Optimal Value Function

When we go to continual tasks (or even finite horizon with $T > 1$), we can ask several questions:

- (Planning Problem) How can we find an optimal policy (if it exists) given the model $\mathcal{P}$ and $\mathcal{R}$?
- (RL Problem) How we can *learn* π^* (or a close approximation) without actually knowing the MDP, but only have samples coming from interacting with the MDP?

Q-Learning

It takes a while before we learn the necessary background before facing the first RL algorithms. Before that, let's have a sneak peak at one of the most well-known RL algorithms: Q-Learning. Q-Learning is the quintessential RL algorithm, introduced by Christopher Watkins [Watkins, 1989, Chapter 7 – Primitive Learning]. Q-Learning itself is an example of the Temporal Difference (TD) learning [Sutton, 1988].

Q-Learning

Require: Step size $\alpha \in (0, 1]$

Initialize $Q : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ arbitrary, except that for x_{terminal}, set $Q(x_{\text{terminal}}, \cdot) = 0$.

for each episode **do**

  Initialize $X_1 \sim \rho$

  for each step t of episode **do**

    $A_t \sim \pi(\cdot|X_t)$,

    Take action A_t, observe X_{t+1} and R_t

    Update $Q(X_t, A_t)$ using the following update rule

$$Q(X_t, A_t) \leftarrow Q(X_t, A_t) + \alpha \left[R_t + \gamma \max_{a' \in \mathcal{A}} Q(X_{t+1}, a') - Q(X_t, A_t) \right].$$

  end for

end for

Q-Learning

Variety of policies can be selected. A commonly-used one is ε-greedy policy:

$$A_t = \begin{cases} \operatorname{argmax}_{a \in \mathcal{A}} Q(X_t, a) & \text{w.p. } 1 - \varepsilon \\ \text{uniform}(\mathcal{A}) & \text{w.p. } \varepsilon \end{cases}$$

Usually the value of ε is small and may go to zero as the agent learns more about its environment.

Remark

Under certain conditions, including how the learning rate α should be selected, the Q-Learning algorithm on finite state-action MDPs can be guaranteed to converge to the optimal action-value function Q^*.

Course Information

- Course Website: https://amfarahmand.github.io/IntroRL/
- Main source of information is the course webpage. Check regularly!
- We will also use Quercus for **announcements & grades**.
- We will use Piazza for **discussions**.

Course Information

- Lectures will be delivered synchronously via Zoom. They will be recorded and shared for asynchronous viewing.
- You may download recorded lectures for your personal academic use, but you should not redistribute them.
- During lectures, please keep yourself on mute, unless you have a question.
- Please refer to `http://www.illnessverification.utoronto.ca` in case of illness (you need to fill out an absence declaration form on ACORN and contact me).
- If you require additional academic accommodations, please contact UofT Accessibility Services: `https://studentlife.utoronto.ca/department/accessibility-services/`
- I realize that this is an unusually difficult time for all of us. I try to be as accommodating as possible.

Course Evaluation

This is tentative and may change in the next few days:

- Three (3) assignments (15% each, for a total of 45%)
  - Combination of mathematical derivations, proofs, and programming exercises.
- Research Project (30%)
  - Research proposal, written report, peer reviewing, and class presentation.
- Take-Home Exam (15%)
- Read some seminal papers. (10%)
  - Short 1-paragraph summary and two questions on how the method(s) can be used or extended.
- Bonus (5%)
  - First one to find typos, etc. in the lecture notes
  - Active class participation
  - etc ... !

Collaboration and Assignments

Collaboration:

- Collaboration on the assignments is **not** allowed. Each student is responsible for their own work. Discussion of assignments should be limited to clarification of the handout itself, and should not involve any sharing of derivations, pseudocode or code, or simulation results.
- You need to form a team of 2-3 members to work on your projects (the exact number will be determined after finalizing the number of students enrolled).

Late Submissions (assignments, proposals, reports, etc):

- Submissions should be handed in by deadline; a late penalty of 10% per day will be assessed thereafter (up to 3 days, then submission is blocked).
- Extensions will be granted only in special situations, and you will need a Student Medical Certificate or a written request approved by the course coordinator at least one week before the due date.

References

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

Christopher J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, University of Cambride, 1989.