Università degli Studi di Pisa

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI
Corso di Laurea in Matematica

Tesi di Laurea Magistrale

State space exploration through Policy Gradient for sample efficient Reinforcement Learning

Candidate:
Marco Miani

Thesis supervisor:
Prof. Marco Romito
Prof. Maurizio Parton

Anno Accademico 2019/2020

Contents

Introduction

The idea that we learn by interacting with our environment is probably the first to occur to us when we think about the nature of learning. This interaction naturally produce a wealth of information about cause and effect, about consequences of actions, and about what to do in order to achieve goals. Whether we are learning to drive a car or to hold a conversation, we are acutely aware of how our environment responds to what we do, and we seek to influence what happens through our behaviour. Learning from interaction is a foundational idea underlying nearly all theories of learning and intelligence. In this Thesis we explore a computational approach to learning from interaction: Reinforcement Learning.

A behaviour is, matematically, a map from situations to actions. Reinforcement Learning is *learning a behaviour so as to maximize a reward.* The learner is not told which action to take, but instead must discover which action yields the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards.

Reinforcement Learning is simultaneously a problem, a class of solution methods that work well on the problem, and the field that studies this problem and its solution methods. It is convenient to use a single name for all three things, but at the same time essential to keep the three conceptually separate.

We formalize the problem of Reinforcement Learning in Chapter 1 as an interaction between an agent and an environment, where the former is described through a *policy* while the latter is described through a *Markov decision process.* The basic idea, and reason for this setting, is to capture the most important aspects of the real problem facing a learning agent interacting over time with its environment to achieve a goal. A learning agent must be able to sense the state of its environment to some extent and must be able to take actions that affect the state. The agent must also have a goal or goals relating to the state of the environment. Markov decision processes are intended to include just these three aspects -sensation, action and goal- in their simplest possible forms without trivializing any of them.

We also introduce important concepts often neglected in literature such as *state visitation distribution* and *sampling model.* The formalization and deep comprehension of these concepts is fundamental in order to fully understand the power of the *Coverage Hypothesis*, which is a key aspect of this work.

The key idea of Reinforcement Learning solution methods is to use value functions (1.11) to organize and structure the search of good policies. In Chapter 2 we give an overview of the two macro families in which learning algorithms split: value based methods and policy based methods.

In Chapter 3 we focus our attention on Policy Gradient methods, uncovering and analyzing the issues that can arise and how these issues are dealt with in the literature. Difficulties are isolated in toy examples (in particular Example 3.6 and Example 3.10) and possible solutions are described, from the addition of regularization term in order to stabilize iterations, to gradient based bounds of the distance from the optimum. The problem of visiting every state frequently, *i.e.* of exploring the state space, is addressed in Section 3.3, particular focus is given to the *Coverage Hypothesis.* The coverage hypothesis in particular is a key aspect when dealing with the exploration-exploitation dilemma: it assures that trajectories can start from any state and thus automatically guarantees the exploration of the state space.

In Chapter 4 we deepen in a specific learning algorithm. First we derive some useful results on the relations between value functions, gradients and state visitation distributions. We illustrate and then rely on gradient domination results in order to quantify the goodness of a policy with respect to an optimal one. In Section 4.3 we explain some of the most common single trajectory gradient approximator for the target function and for each of them we state a result that leads to a common *Approximation Hypothesis.* Any other gradient approximator may be used as long as it satisfies this approximation hypothesis.

In Chapter 5 we describe and slightly generalize a very recent learning algorithm, originally proposed by [10]. The importance of this algorithm relies on it being the first in literature to obtain a convergence rate sublinear in the number of samples and polynomial in the problem relevant parameters, without assuming diverging minibatch size. The coverage hypothesis is crucial for this result.

Finally, we develop an iterative state space exploration strategy that builds over the previously described algorithm. The idea is defining a set of poorly visited states and then simulating a positive reward on them. This, searching for optimal simulated value policies, leads to more frequent visitations, which in turn lead to a simulation of a more covering sampling model. This is then used as starting point in the successive step. We prove that this exploration strategy can get around the coverage hypothesis and thus leads to stronger solving methods. To the best of our knowledge, this is an original contribution.

We conclude with Chapter 6 containing numerical experiments. We illustrate empirically the theoretical results; in particular providing examples where our exploration strategy is more efficient than the one in [10].

Chapter 1

Setting

1.1 Reinforcement Learning

Consider the problem in which an agent is faced with the task of influencing an environment through the action it takes. At each timestep the agent is at a state in the environment and it must make a decision of which action to perform. This action alters the state the agent is at and determines the reward the agent recieves.

Example 1.1. Pick-and-Place Robot Consider using Reinforcement Learning to control the motion of a robot arm in a repetitive pick-and-place task. First we want to represent the robot arm mechanics, *i.e.* the environment. The *states* $\mathcal{S}$ are all the possible position of the arm: joint angles and velocities. The *actions* $\mathcal{A}$ are all the the possible voltages applied to each motor at each joint, that is what the agent has choice on. The length and weight of the arm and on the specs of the motors define the physics of the system which in turn determine how the system evolve in time, this is the *transition model* $\mathcal{P}$. Finally, the agent recives a positive *reward* r for each object successfully picked up and placed.
The more the "resolved" objects are, the larger the total reward will be.

1.1.1 Environment formalization

A Markov Decision Process formalizes the environment behaviour regarding the possible action made.

Definition 1.2. A **Markov Decision Process (MDP)** $\mathcal{M}$ is a tuple which consists of:

- a state space $\mathcal{S}$;

- a action space $\mathcal{A}$;
- a transition model $\mathcal{P}(s'|s,a)$ for any $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$;
- a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

The transition model $\mathcal{P}(\cdot|s,a)$ is a probability distribution on $\mathcal{S}$, for each $s \in \mathcal{S}$ and $a \in \mathcal{A}$. The probability $\mathcal{P}(s'|s,a)$ represents the probability of transitioning to s' after performing action a in state s.

The reward function is assumed to be bounded, *i.e.* there exist two constants $r_{\text{inf}}, r_{\text{sup}} \in \mathbb{R}$ such that $r(s,a) \in [r_{\text{inf}}, r_{\text{sup}}]$ for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$.

Example 1.3. Pole-balancing Cart Consider using Reinforcement Learning to control a cart, the objective here is to apply forces to the cart moving along a track so as to keep a pole hinged to the cart from falling over. A failure is said to occur if the pole falls past a given angle α from vertical or if the cart runs off the track. This task is presented in Figure 1.1. First of all we are interested in representing the task as a MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r)$:

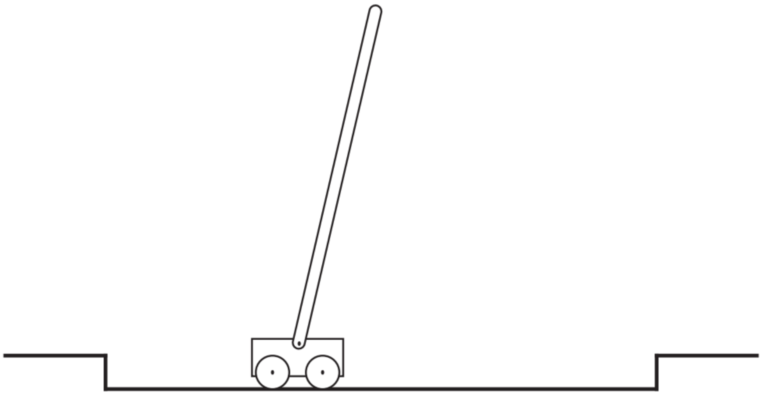


Figure 1.1: The pole-balancing task.

- The state space in this case might be $\mathcal{S} := [0, L] \times \mathbb{R} \times \Theta \times \mathbb{R}$, a state $s = (s_p, s_v, s_a, s_\omega)$ is the tuple representing position and velocity of the cart and angle and angular velocity of the pole.
- The action space might be $\mathcal{A} := [-F, F]$, the possible forces applied to the cart.
- The transition model $\mathcal{P}(s'|s,a)$ is defined by the physics of the system. Given the length and mass distribution of the pole, it is uniquely determined the state s' of the system, given the previous s and a (in a time discretized physics).

- The reward r in this case could be $+1$ for every time step on which failure did not occur. In details, $r(s, a) = 1$ if $0 \neq s_p \neq L$ and $-\alpha \neq s_a \neq \alpha$.

With this setting the sum of rewards would be the number of steps until failure, which is maximized by keeping the pole balanced for as long as possible. The *value function* (1.11) is, roughly, a function from cart-pole situation to how many timesteps can the agent survive from that situation.

Remark. Given a MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r)$ we can assume without loss of generality that $r_{inf} = 0$ and $r_{sup} = 1$.
In fact, given $\mathcal{M}$, define $r'(s, a) := \frac{r(s,a)-r_{inf}}{r_{sup}-r_{inf}}$ and set $\mathcal{M}' := (\mathcal{S}, \mathcal{A}, \mathcal{P}, r')$. It is clear that $r'(s, a) \in [0, 1]$ and, by linearity, all the *relations and ordering* of $\mathcal{M}$ are preserved and thus the results provided in this Thesis easily generalize to the case of an arbitral interval. This reduction is very helpful both for normalizing the error measures and, not least importantly, for clarity of presentation.

Fact. Some authors in literature, see for example [7], define rewards to be stochastic, *i.e.* r is not a function to $\mathbb{R}$ but to distributions over $\mathbb{R}$. That is, r is not a function $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$ but a set of distribution $\{r(\cdot|s, a)\}_{s\in\mathcal{S},a\in\mathcal{A}}$.

1.1.2 Agent formalization

The agent behaviour in a Markov Decision Process is formalized by a policy. A policy specifies a sequence of decision rules for action selection at all timestep. We only consider policies such that the action selection only depends on the current state and is thus indepentent from the previously visited states. These policies are called *Markovian* policies and could possibily depend on the timestep.

Definition 1.4. Given a Markov Decision Process $\mathcal{M}$, a stationary **policy** π is the set of distributions $\{\pi(\cdot|s)\}_{s\in\mathcal{S}}$ where $\pi(\cdot|s)$ is a distribution over the action space $\mathcal{A}$. The set of all possible stationary policies is $\Pi_{\mathcal{M}}$.

The probability $\pi(a|s)$ represents the probability of the agent taking the action a in state s.

Definition 1.5. Given a MDP $\mathcal{M}$, a stationary **deterministic policy** is a function $\pi : \mathcal{S} \to \mathcal{A}$.

The set $\Pi^D_{\mathcal{M}} := \{\pi \in \Pi_{\mathcal{M}} |\ \forall s \in \mathcal{S}, \exists a \in \mathcal{A} \text{ such that } \pi(a|s) = 1\} \subseteq \Pi_{\mathcal{M}}$ is in one-to-one correspondence with the set of all possible deterministic policies $\mathcal{A}^{\mathcal{S}}$, this correspondence is shown explicitely by

$$\begin{aligned} \mathcal{A}^{\mathcal{S}} &\to \Pi_{\mathcal{M}}^{D} \\ \pi &\mapsto \left\{(1 \text{ if } \pi(s) = a \text{ else } 0)_{a\in\mathcal{A}}\right\}_{s\in\mathcal{S}} \end{aligned}$$

that is clearly invertible. With a slight abuse of notation, we identify the two characterizations when dealing with deterministic policies, the difference will be clear from context.

Definition 1.6. Given a MDP $\mathcal{M}$ and a finite time horizon T a **non-stationary policy** is a function $\pi : \{0, ..., T-1\} \to \Pi_{\mathcal{M}}$.

Non-stationary policies can be viewed as time dependent distributions $\pi(\cdot|s,t)$. In the case of **finite time horizon MDP**, where the process stops after at most T timestep, it is quite straightforward to extend convergence results to non-stationary policies at the cost of at most a factor T. This is shown more extensively by Kakade in [3]. Given this, we choose to limit this work to stationary policies in order to maximize the clarity of presentation.

1.1.3 Trajectories

A Markov Decision Process and a policy, mutually interacting, completely define how the system evolves with time. More specifically, the agent and the environment interact at each of a sequence of discrete time steps $t = 0, 1, 2, 3, ..., \infty$.

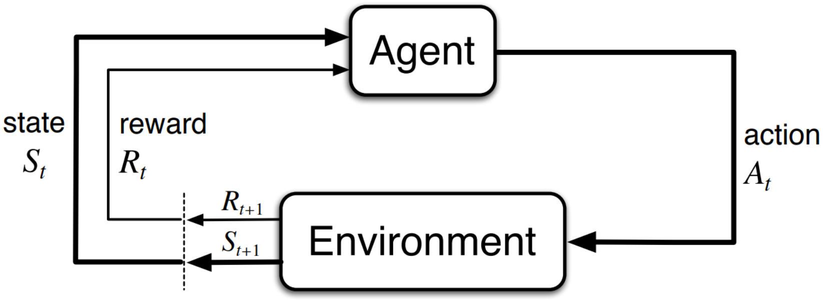


Figure 1.2: The agent–environment interaction in a Markov Decision Process.

At each timestep t, the agent recieves some representation of the environment state $S_t \in \mathcal{S}$, and on that basis select an action $A_t \in \mathcal{A}$. One timestep later, in part as a consequence of its action and in part due to the transition model $\mathcal{P}$, the agent receives a numerical reward $R_{t+1} = r(S_t, A_t)$ and find itself in a new state, S_{t+1}

Definition 1.7. A **trajectory** τ of length H is a sequence of state-action

$$\tau = (s_0, a_0, s_1, a_1, s_2, a_2, ..., s_{H-1}, a_{H-1}).$$

Definition 1.8. The **trajectories space of length** H

$$T_H := (\mathcal{S} \times \mathcal{A})^H$$

is the set of all trajectories of length H. The **trajectory space** is the union of them

$$\mathcal{T} := \bigcup_{H>0} T_H.$$

The mutual interaction beteween agent and environment defines how trajectories are sampled or, more precisely, how they evolve. The agent samples the next action given the current state, the environment samples the next state given the current state and action.

Definition 1.9. Given a MDP $\mathcal{M}$ and a policy $\pi \in \Pi_{\mathcal{M}}$, an **episode** is a trajectory $\tau \in \mathcal{T}$ that is sampled according to

$$a_t \sim \pi(\cdot|s_t) \quad s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t) \quad t \geq 0, \tag{1.1}$$

where s_0 is choosen by a *sampling model* and the length H by some *termination condition.*

We define sampling models and termination conditions in deeper details in Section 1.3. For now we focus on the mutual interaction between agent and environment and we assume a fixed initial state s_0 and episode length H.

Given the iterative definition (1.1), the initial state $s_0 \in \mathcal{S}$, a MDP $\mathcal{M}$ and a policy π induce a probability distribution over the trajectories space T_H, for every length $H > 0$. Since every action is independent from the others, because we are considering Markovian and stationary policies, the probability of a trajectory $\tau = (s_0, a_0, s_1, a_1, ..., s_{H-1}, a_{H-1})$ is:

$$\mathbb{P}(\tau|\pi, \mathcal{M}, s_0) = \pi(a_0|s_0) \prod_{t=1}^{H-1} \mathcal{P}(s_t|s_{t-1}, a_{t-1})\pi(a_t|s_t). \tag{1.2}$$

In the same spirit, fixing both the initial state $s_0 \in \mathcal{S}$ and the initial action $a_0 \in \mathcal{A}$ we have

$$\mathbb{P}(\tau|\pi, \mathcal{M}, s_0, a_0) = \prod_{t=1}^{H-1} \mathcal{P}(s_t|s_{t-1}, a_{t-1})\pi(a_t|s_t). \tag{1.3}$$

Of course some trajectories may have zero probability. This can happen if some action is never chosen in a certain state or if there is no transition between a state and its successor according to the MDP.

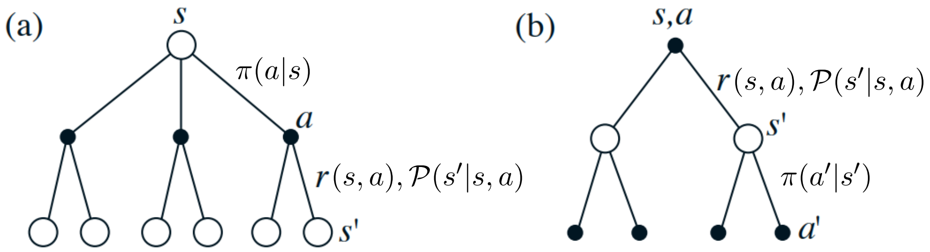


Figure 1.3: Backup diagrams where root node is a state (a) or a state-action pair (b)

Backup diagrams can help develop intuition of what is going on. Each open circle represents a state and each solid circle represents an action. Think of looking ahead from one state to its possible successor states, as suggested by Figure 1.3(a). Starting from state s, the root node at the top, the agent could take any action $a \in \mathcal{A}$ according to $\pi(\cdot|s)$. From each of these, the environment could respond with one of several next states $s' \in \mathcal{S}$ according to $\mathcal{P}(\cdot|s,a)$, along with reward, $r(s,a)$.

Different state's circles in a backup diagram does not necessarily mean different states, in fact a transition $\mathcal{P}(s|s,a)$ from a state s back into itself may have non zero probability. With this in mind, we can think of drawing all this future scenario trees, once for each state, and then merge the nodes representing the same. This idea of open circle for *state nodes* and solid circles for *action nodes* can thus be used to represent a whole Markow Decision Process as a graph. This graph in litterature is commonly called *transition graph* and an example is shown in Figure 1.4.

Definition 1.10. Given a MDP $\mathcal{M}$, its **transition graph** is a directed graph with state and action nodes connected by edges. There is a state node n_s for each possibile state $s \in \mathcal{S}$, and there is an action node $n_{s,a}$ for each state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$. For each action node $n_{s,a}$ there is an edge connecting to its respective state node n_s. There is also an edge connecting an action node $n_{s,a}$ with a state node $n_{s'}$ each time that transition is present in the transition model of the MDP, *i.e.* if $\mathcal{P}(s'|s,a) \neq 0$.

Example 1.11. Recycling robot A mobile robot has the job of collecting empty soda cans in an office environment. It has sensors for detecting cans, and an arm and gripper that can pick them up and place them in an onboard bin; it runs on rechargeable battery.

High-level decision about how to search for cans are made by a reinforcement learning agent based on current level of battery. To keep the example simple, we assume that only two charge levels can be distinguished. This leads to the state space

$$\mathcal{S} := \{\texttt{high}, \texttt{low}\}.$$

In each state, the agent can decide whether to (1) actively `search` for a can for a certain period of time, (2) remain stationary and `wait` for someone to bring it a can, or (3) head back

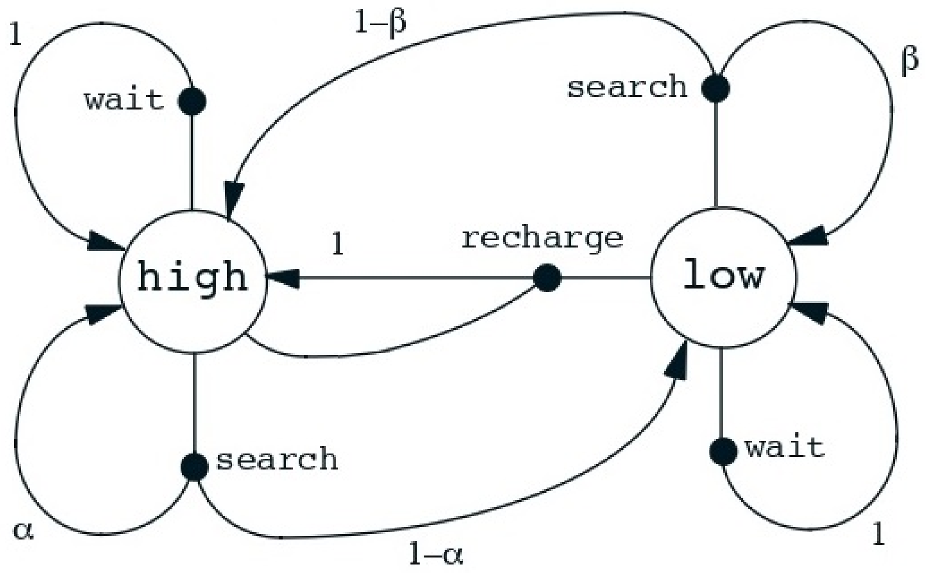


Figure 1.4: Transition graph for the recycling robot task.

to its home base to `recharge` its battery. The action space is thus

$$\mathcal{A} := \{\texttt{search}, \texttt{wait}, \texttt{recharge}\}$$

s'	s	a	$\mathcal{P}(s'\|s,a)$
high	high	search	α
low	high	search	$1-\alpha$
high	low	search	$1-\beta$
low	low	search	β
high	high	wait	1
low	high	wait	0
high	low	wait	0
low	low	wait	1
high	high	recharge	1
low	high	recharge	0
high	low	recharge	1
low	low	recharge	0

Figure 1.5: Transition probabilities for the recycling robot task

The best way to find cans is to actively search for them, but this runs out the robot's battery, whereas waiting does not. This is represented by $\mathcal{P}(s|s,\texttt{wait}) = 1$ for every state s.
A period of active search that begins with a `high` energy level leaves the enery level `high` with probability α and reduces it to `low` with probability $1-\alpha$. This is represented by $\mathcal{P}(\texttt{high}|\texttt{high},\texttt{search}) = \alpha$ and $\mathcal{P}(\texttt{low}|\texttt{high},\texttt{search}) = 1-\alpha$.
On the other hand, a period of searching undertaken when the energy level is `low` leaves it `low` with probability β and depletes the battery with probability $1-\beta$. In the latter case, the robot must be rescued, and the battery is then recharged back to `high`. This is represented by $\mathcal{P}(\texttt{low}|\texttt{low},\texttt{search}) = \beta$ and $\mathcal{P}(\texttt{high}|\texttt{low},\texttt{search}) = 1-\beta$.
Lastly, a period of recharging always lead to a `high` energy level. This is represented by $\mathcal{P}(\texttt{high}|s,\texttt{recharge}) = 1$ for every state s. All those transition probabilities are reported in Figure 1.5.

The rewards are zero most of the time, but become positive when the robot secures an empty can. Let $r_{\texttt{search}}$ and $r_{\texttt{wait}}$, with $r_{\texttt{search}} > r_{\texttt{wait}}$, respectively denote the expected number of cans the robot will collect while searching and while waiting. Thus, the reward function is defined by $r(s, \texttt{search}) = r_{\texttt{search}}$, $r(s, \texttt{wait}) = r_{\texttt{wait}}$ and $r(s, \texttt{recharge}) = 0$.

Notice that this model can be improved, for example, by adding one more state for the situation of `depleted` battery. This leads to greater problem complexity but also to greater expressiveness, a compromise has to be made at some point.

Remark. Before going further we need to make a key observation in order to clear the notation in the next chapters. We will often calculate the expected value with episodes, *i.e.* trajectories τ following the distribution (1.2) or (1.3). In principle this expectation is length dependant, but this is not the case thanks to the fact that the trajectories spaces are monotonically increasing in the length, *i.e.* $\bigcup_{H \le n} T_H \subset \bigcup_{H \le n+1} T_H$.
To see this rigorously, consider some $\tau \in T_H$ and a state-action pair $(s_H, a_H) \in \mathcal{S} \times \mathcal{A}$. Now let $\tau' \in T_{H+1}$ be the trajectory of length $H+1$ obtained by juxtapposition of τ and (s_H, a_H). By (1.2)

$$\mathbb{P}(\tau'|\pi, \mathcal{M}, s_0) = \mathbb{P}(\tau|\pi, \mathcal{M}, s_0)\mathcal{P}(s_H|s_{H-1}, a_{H-1})\pi(a_H|s_H)$$

and this holds for every $\tau \in T_H$ and $(s_H, a_H) \in \mathcal{S} \times \mathcal{A}$. To makes things clearer we explicit the length of the trajectory as subscript of $\mathbb{P}$, that is

$$\mathbb{P}_{H+1}(\tau'|\pi, \mathcal{M}, s_0) = \mathbb{P}_H(\tau|\pi, \mathcal{M}, s_0)\mathcal{P}(s_H|s_{H-1}, a_{H-1})\pi(a_H|s_H) \tag{1.4}$$

Now considering a generic function $f : T_H \to \mathbb{R}$:

$$\begin{aligned}
\mathbb{E}_{\tau \sim \mathbb{P}_H(\cdot|\pi,\mathcal{M},s_0)}[f(\cdot)] &= \sum_{\tau \in T_H} \mathbb{P}_H(\tau|\pi, \mathcal{M}, s_0) f(\cdot) = \\
&= \sum_{\tau \in T_H} \mathbb{P}_H(\tau|\pi, \mathcal{M}, s_0) f(\cdot) \sum_{s_H} \mathcal{P}(s_H|s_{H-1}, a_{H-1}) = \\
&= \sum_{\tau \in T_H} \mathbb{P}_H(\tau|\pi, \mathcal{M}, s_0) f(\cdot) \sum_{s_H} \mathcal{P}(s_H|s_{H-1}, a_{H-1}) \sum_{a_H} \pi(a_H|s_H) = \\
&= \sum_{\tau \in T_H} \sum_{s_H, a_H} \mathbb{P}_H(\tau|\pi, \mathcal{M}, s_0)\mathcal{P}(s_H|s_{H-1}, a_{H-1})\pi(a_H|s_H) f(\cdot) = \\
&= \sum_{\tau' \in T_{H+1}} \mathbb{P}_H(\tau|\pi, \mathcal{M}, s_0)\mathcal{P}(s_H|s_{H-1}, a_{H-1})\pi(a_H|s_H) f(\cdot) = \\
&= \sum_{\tau' \in T_{H+1}} \mathbb{P}_{H+1}(\tau'|\pi, \mathcal{M}, s_0) f(\cdot) = \\
&= \mathbb{E}_{\tau' \sim \mathbb{P}_{H+1}(\cdot|\pi,\mathcal{M},s_0)}[f(\cdot)],
\end{aligned}$$

where the first and last equalities are the definition of the expected value. The second and third equalities follow since $\mathcal{P}(\cdot|s, a)$ and $\pi(\cdot|s)$ respectively are probability distributions. The fourth equality is a rearrangement that is true beacause $f(\cdot)$ depends at most from the first H steps of the trajectory. The fifth equality is by definition of T_{H+1} and the sixth follows from (1.4).

Iterating this argument we obtain that is well defined the expectation with respect to trajectories of non-specific length.

$$\begin{aligned}\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0)}[f(\cdot)] &:= \mathbb{E}_{\tau\sim\mathbb{P}_H(\cdot|\pi,\mathcal{M},s_0)}[f(\cdot)] \\ &= \mathbb{E}_{\tau'\sim\mathbb{P}_{H+k}(\cdot|\pi,\mathcal{M},s_0)}[f(\cdot)] \qquad \forall k \geq 0\end{aligned}$$

for some H big enough so that f makes sense. In the exact same way, starting from (1.3), the expectation with respect to trajectories of non-specific length is well defined also in the case of both fixed initial state and initial action $\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0,a_0)}[f(\cdot)]$.

1.1.4 State visitation distribution

A trajectory of length H is a path between a state s_0 and another state s_{H-1}. Having fixed the initial state s_0, each state has a certain probability of being s_{H-1}, the state visited at timestep $H-1$. So the distribution over the trajectories space T_H naturally induces a distribution over the state space $\mathcal{S}$. The **state visitation** distribution after H timesteps is defined as

$$\begin{aligned}d^{\mathcal{M},\pi}_{s_0,H}(s) &:= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0)}[\mathbb{1}_{\{s_{H-1}=s\}}] = \\ &= \sum_{\tau\in T_H} \mathbb{1}_{\{s_{H-1}=s\}}\mathbb{P}(\tau|\pi,\mathcal{M},s_0)\end{aligned}$$

where $\mathbb{1}_X$ is the indicator function of X and the dependance from the initial state s_0 is made clear from the subscript.

It is now sort of natural to consider a sum over H of this distributions. It will be useful to define a discounted distribution, that is an average with geometric weight of the state visitation distributions as H varies.

Definition 1.12. Given a MDP $\mathcal{M}$, a policy π and a real number $\gamma \in [0,1)$, for every $s_0 \in \mathcal{S}$ the **discounted state visitation** $d^{\mathcal{M},\pi,\gamma}_{s_0}$ is a distribution over the state space $\mathcal{S}$ and it is defined by

$$\begin{aligned}d^{\mathcal{M},\pi,\gamma}_{s_0}(s) &:= (1-\gamma)\sum_{H=0}^{\infty}\gamma^H d^{\mathcal{M},\pi}_{s_0,H+1}(s) \qquad (1.5)\\ &= (1-\gamma)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0)}\left[\sum_{H=0}^{\infty}\gamma^H\mathbb{1}_{\{s_H=s\}}\right].\end{aligned}$$

The term $(1-\gamma)$ is a normalization factor.

In a similar way the distribution on trajectories space T_H induce also a distribution over the state-action space $\mathcal{S} \times \mathcal{A}$. For the definition we overload the notation, but the difference will always be clear from the number of arguments or from the context.

The **state-action visitation** distribution for length H is the distribution of (s_{H-1}, a_{H-1}) for trajectories starting from s_0.

$$\begin{aligned} d^{\mathcal{M},\pi}_{s_0,H}(s,a) &:= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0)}[\mathbb{1}_{\{s_{H-1}=s\}}\mathbb{1}_{\{a_{H-1}=a\}}] = \\ &= \sum_{\tau\in T_H} \mathbb{1}_{\{s_{H-1}=s\}}\mathbb{1}_{\{a_{H-1}=a\}}\mathbb{P}(\tau|\pi,\mathcal{M},s_0). \end{aligned}$$

Definition 1.13. Given a MDP $\mathcal{M}$, a policy π and a real number $\gamma \in [0,1)$, for every $s_0 \in \mathcal{S}$ the **discounted state-action visitation** $d^{\mathcal{M},\pi,\gamma}_{s_0}$ is a distribution over the state-action space $\mathcal{S} \times \mathcal{A}$ and it is defined as

$$\begin{aligned} d^{\mathcal{M},\pi,\gamma}_{s_0}(s,a) &:= (1-\gamma)\sum_{H=0}^{\infty} \gamma^H d^{\pi,\mathcal{M}}_{s_0,H+1}(s,a) = \\ &= (1-\gamma)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0)}\left[\sum_{H=0}^{\infty}\gamma^H \mathbb{1}_{\{s_H=s\}}\mathbb{1}_{\{a_H=a\}}\right]. \end{aligned} \tag{1.6}$$

For every $s \in \mathcal{S}$ and $a \in \mathcal{A}$ the state distribution and the state-action distribution satify the relation

$$d^{\mathcal{M},\pi}_{s_0,H}(s,a) = \pi(a|s)\, d^{\mathcal{M},\pi}_{s_0,H}(s),$$

which easily extends by a linearity argument to the discounted distributions

$$d^{\mathcal{M},\pi,\gamma}_{s_0}(s,a) = \pi(a|s)\, d^{\mathcal{M},\pi,\gamma}_{s_0}(s). \tag{1.7}$$

Remark. The real number $\gamma \in [0,1)$ involved in Definitions 1.12 and 1.13 is the **discount factor**. The discount factor represents how much some quantities lose relevance with increasing time distance, where those quantities can be the state visitation distributions, as in (1.5), or the rewards, as in (1.8). We refer to it with the letter γ throughout this whole Thesis. The case $\gamma = 1$ may also be considered but it requires extra hypotheses in order to guarantee measurability of value functions (1.11) and (1.12).

1.2 Aim

Given a Markov Decision Process, the policy chosen by the agent induces a distribution over trajectories which in turn induces a distribution over the sequence of rewards the agent recieves. The objective of the agent is to obtain a reward sequence that is as "large" as possible. This Section defines some standard optimality criteria.

In order to make clear the differences between successive timesteps without abusing the notation we make use of variable S_t, A_t and R_t for respectively, state, action and reward, at timestep t in an episode. The process is stochastic and so those are random variables. The randomness comes from transition probabilities, action choices and sampling model.

At timestep t, the sum of future earned rewards R_{t+k} for $k \geq 0$

$$G_t := R_{t+1} + R_{t+2} + R_{t+3} + R_{t+4} + ...$$

is called **return**. We seek to maximize it or, more precisely, its expected value.

In order to have a nice finiteness property of this return we need either to be in the finite horizon setting where $t \leq T$ or, being in the infinite horizon setting, to introduce the *discount factor*. This immediately leads to the definition of the *discounted return*.

Definition 1.14. Given a MDP and a discount factor $\gamma \in [0, 1)$, the **discounted return** G_t at timestep t is a random variable defined by

$$G_t := R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{1.8}$$

It is the sum of future *discounted* earned rewards R_{t+k} for $k \geq 0$.

The discount factor here represents how much the reward lose value with increasing time distance: a reward received k timestep in the future is worth only γ^{k-1} times what it would be worth if it were received immediately.

Fact. Assume $0 \leq r(s, a) \leq 1$ for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$, see Remark 1.1.1. Then for every timestep $t \geq 0$ the discounted return (1.8) is bounded by

$$0 \leq G_t \leq \frac{1}{1-\gamma}. \tag{1.9}$$

Proof. Since $r(s, a)$ is bounded for every $(s, a) \in \mathcal{S} \times \mathcal{A}$, the random variable $R_{t'}$ representing the reward at timestep t' is also bounded in the interval $[0, 1]$ for every $t' \geq 0$. Then

$$0 \leq G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \leq \sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}.$$

□

Discounted returns at successive timesteps are related to each other in a way that is important for the theory and algorithms of reinforcement learning, in fact this is the foundation of the Bellman equations (1.15) and (1.16)

$$\begin{aligned} G_t =& R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} ... \\ =& R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} ...\right) \\ =& R_{t+1} + \gamma G_{t+1} \end{aligned} \tag{1.10}$$

1.2.1 Value functions

The aim of the agent is to maximize the expected value of the discounted return. The *value function* of a state s under a policy π, denoted as $V_{\mathcal{M},\pi,\gamma}(s)$, is the expected discounted return when starting in s and following π. Intuitively

$$\begin{aligned} V_{\mathcal{M},\pi,\gamma}(s) &:= (1-\gamma)\mathbb{E}_{\mathcal{M},\pi}[G_t|S_t = s] = \\ &= (1-\gamma)\mathbb{E}_{\mathcal{M},\pi}\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\middle| S_t = s\right] \end{aligned}$$

More rigorously for an MDP $\mathcal{M}$, using the distribution on trajectories that π induces:

Definition 1.15. Given and MDP $\mathcal{M}$, a policy π and a discount factor γ, the **value function** $V_{\mathcal{M},\pi,\gamma} : \mathcal{S} \mapsto [0,1]$ is

$$V_{\mathcal{M},\pi,\gamma}(s) := (1-\gamma)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right]. \tag{1.11}$$

Note that since we are interested in ϵ-optimality criteria, we define *normalized* value functions adding the factor $(1-\gamma)$.

Similarly, we define the value of taking action a in state s under a policy π, denoted as $Q_{\mathcal{M},\pi,\gamma}(s,a)$, as the expected return starting from s, taking the action a, and thereafter following policy π:

$$\begin{aligned} Q_{\mathcal{M},\pi,\gamma}(s,a) &:= (1-\gamma)\mathbb{E}_{\mathcal{M},\pi}[G_t|S_t = s, A_t = a] = \\ &= (1-\gamma)\mathbb{E}_{\mathcal{M},\pi}\left[\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}\middle| S_t = s, A_t = a\right]. \end{aligned}$$

Definition 1.16. Given and MDP $\mathcal{M}$, a policy π and a discount factor γ, the **state-action value function** $Q_{\mathcal{M},\pi,\gamma} : \mathcal{S} \times \mathcal{A} \mapsto [0,1]$ is

$$Q_{\mathcal{M},\pi,\gamma}(s,a) := (1-\gamma)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s,a_0=a)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right] \tag{1.12}$$

Here, similarly to (1.11), we normalize with the factor $(1-\gamma)$.

Backup diagram are helpful for the intuition of the relationship occurring between value function $V_{\mathcal{M},\pi,\gamma}$ and state-action value function $Q_{\mathcal{M},\pi,\gamma}$, see Proposition 1.18.

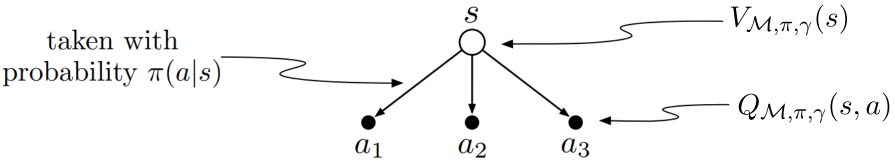
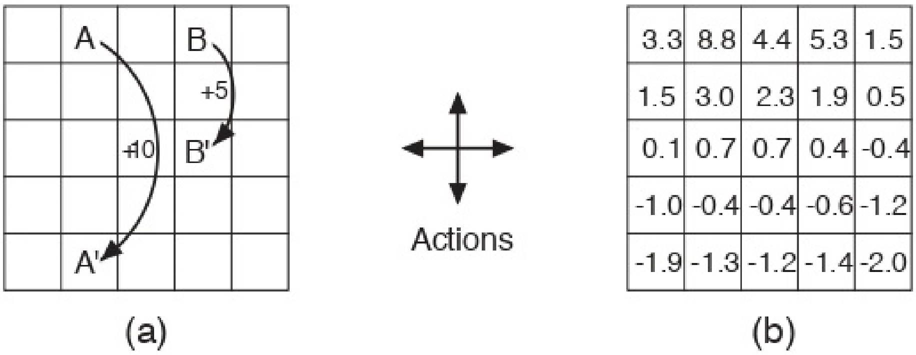

s
$V_{\mathcal{M},\pi,\gamma}(s)$
taken with probability $\pi(a|s)$
$Q_{\mathcal{M},\pi,\gamma}(s,a)$
a_1
a_2
a_3

Figure 1.6: Value function and action-value function in a backup diagram.

Example 1.17. Gridworld Figure 1.7(a) shows a rectangular gridworld representation of a simple finite MDP. The cells of the grid correspond to the states $\mathcal{S}$. At each cell, four action are possible: $\mathcal{A} = \{\texttt{north}, \texttt{south}, \texttt{east}, \texttt{west}\}$, which deterministically cause the agent to move one cell in the respective direction on the grid. Actions that would take the agent off the grid leave its location unchanged, but also results in a reward of -1. Other actions result in a reward of 0, except those that move the agent out of the special states A and B. From state A, all four actions yield a reward of $+10$ and take the agent to A'. From state B, all four actions yield a reward of $+5$ and take the agent to B'.



3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

Figure 1.7: Gridworld example: exceptional reward dynamics (a) and state value function for the equiprobable random policy (b).

Suppose a equiprobable random policy $\pi^U \in \Pi_{\mathcal{M}}$ such that $\pi^U(a|s) = \frac{1}{|\mathcal{A}|}$ for every $s \in \mathcal{S}$, $a \in \mathcal{A}$, representing an agent that selects all four actions with equal probability in all states. Set discount factor $\gamma = 0.9$. Figure 1.7(b) shows the value function $V_{\mathcal{M},\pi^U,\gamma}(s)$ of π^U for every cell $s \in \mathcal{S}$.

A foundamental property of value functions used throughout reinforcement learning and dynamic programming is that they satisfy recursive relationships similar to the one we have already established for the discounted return (1.10). For any policy π and any state s, the following consistency conditions hold between the value of s and the value of its possible successor states:

$$\begin{aligned} V_{\mathcal{M},\pi,\gamma}(s) &= (1-\gamma)\mathbb{E}_{\mathcal{M},\pi}[G_t|S_t = s] = && \text{(by 1.10)} \\ &= (1-\gamma)\mathbb{E}_{\mathcal{M},\pi}[R_{t+1} + \gamma G_{t+1}|S_t = s] = \\ &= (1-\gamma)\mathbb{E}_{\mathcal{M},\pi}[R_{t+1} + \gamma \mathbb{E}_{\pi,\mathcal{M}}[G_{t+1}]|S_t = s] = \\ &= \mathbb{E}_{\mathcal{M},\pi}[(1-\gamma)R_{t+1} + \gamma V_{\mathcal{M},\pi,\gamma}(S_{t+1})|S_t = s]. \end{aligned}$$

This succession of equalities gives us the intuition of the recursivity inherited by the Bellman equations. Of course a similar intuition can be given for the action-value function.

In a rigorous way we can establish two relations between the value function and the state-action value function.

Proposition 1.18. Given an MDP $\mathcal{M}$, a policy π and a discount factor γ, the value function $V_{\mathcal{M},\pi,\gamma}$ and the state-action value function $Q_{\mathcal{M},\pi,\gamma}$ satisfy:

$$\begin{aligned} V_{\mathcal{M},\pi,\gamma}(s) &= \sum_{a\in\mathcal{A}} \pi(a|s) Q_{\mathcal{M},\pi,\gamma}(s,a) \\ &= \mathbb{E}_{a\sim\pi(\cdot|s)}[Q_{\mathcal{M},\pi,\gamma}(s,a)], \end{aligned} \tag{1.13}$$

$$\begin{aligned} Q_{\mathcal{M},\pi,\gamma}(s,a) &= (1-\gamma)r(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) V_{\mathcal{M},\pi,\gamma}(s') \\ &= (1-\gamma)r(s,a) + \gamma \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V_{\mathcal{M},\pi,\gamma}(s')], \end{aligned} \tag{1.14}$$

for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$.

Proof. Using the definition and the properties of the expected value, we have

$$\begin{aligned} V_{\mathcal{M},\pi,\gamma}(s) &= (1-\gamma)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \\ &= (1-\gamma)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\sum_{a\in\mathcal{A}}\mathbb{1}_{\{a_0=a\}}\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \\ &= (1-\gamma)\sum_{a\in\mathcal{A}}\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\mathbb{1}_{\{a_0=a\}}\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \\ &= (1-\gamma)\sum_{a\in\mathcal{A}}\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\mathbb{1}_{\{a_0=a\}}\right]\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\middle| a_0=a\right] = \\ &= (1-\gamma)\sum_{a\in\mathcal{A}}\mathbb{P}(a_0=a|s_0=s)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s,a_0=a)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \\ &= \sum_{a\in\mathcal{A}}\pi(a|s)Q_{\mathcal{M},\pi,\gamma}(s,a), \end{aligned}$$

where we used that $\sum_{a\in\mathcal{A}}\mathbb{1}_{\{a_0=a\}} \equiv 1$ and $\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\mathbb{1}_{\{a_0=a\}}\right] = \mathbb{P}(a_0 = a|s_0 = s)$, thanks to the properties of indicator functions. Similarly

$$
\begin{aligned}
Q_{\mathcal{M},\pi,\gamma}(s,a) &= (1-\gamma)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s,a_0=a)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \\
&= (1-\gamma)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s,a_0=a)}\left[\sum_{s'\in\mathcal{S}}\mathbb{1}_{\{s_1=s'\}}\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \\
&= (1-\gamma)\sum_{s'\in\mathcal{S}}\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s,a_0=a)}\left[\mathbb{1}_{\{s_1=s'\}}\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \\
&= (1-\gamma)\sum_{s'\in\mathcal{S}}\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s,a_0=a)}\left[\mathbb{1}_{\{s_1=s'\}}\right]\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s,a_0=a)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\middle|s_1=s'\right] = \\
&= (1-\gamma)\sum_{s'\in\mathcal{S}}\mathbb{P}(s_1=s'|s_0=s,a_0=a)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s,a_0=a,s_1=s')}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \\
&= (1-\gamma)\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s,a_0=a,s_1=s')}\left[r(s_0,a_0)+\gamma\sum_{t=1}^{\infty}\gamma^{t-1}r(s_t,a_t)\right] = \\
&= (1-\gamma)\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\left(r(s,a)+\gamma\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_1=s')}\left[\sum_{t=1}^{\infty}\gamma^{t-1}r(s_t,a_t)\right]\right) = \\
&= (1-\gamma)\left(r(s,a)+\gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s')}\left[\sum_{t=0}^{\infty}\gamma^{t}r(s_t,a_t)\right]\right) = \\
&= (1-\gamma)r(s,a)+\gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)V_{\mathcal{M},\pi,\gamma}(s').
\end{aligned}
$$

□

Thanks to Proposition 1.18 we can easily derive the **Bellman equations**.

Proposition 1.19. For every MDP $\mathcal{M}$, policy π and discount factor γ, the value function $V_{\mathcal{M},\pi,\gamma}$ satisfies the Bellman self-consistency equation for every $s\in\mathcal{S}$

$$
\begin{aligned}
V_{\mathcal{M},\pi,\gamma}(s) &= \sum_{a\in\mathcal{A}}\pi(a|s)\left((1-\gamma)r(s,a)+\gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)V_{\mathcal{M},\pi,\gamma}(s')\right) \qquad (1.15)\\
&= \mathbb{E}_{a\sim\pi(\cdot|s)}\left[(1-\gamma)r(s,a)+\gamma\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V_{\mathcal{M},\pi,\gamma}(s')]\right].
\end{aligned}
$$

Proposition 1.20. For every MDP $\mathcal{M}$, policy π and discount factor γ, the state-action value function $Q_{\mathcal{M},\pi,\gamma}$ satisfies the Bellman self-consistency equation for every $s\in\mathcal{S}$, $a\in\mathcal{A}$

$$
\begin{aligned}
Q_{\mathcal{M},\pi,\gamma}(s,a) &= (1-\gamma)r(s,a)+\gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\sum_{a'\in\mathcal{A}}\pi(a'|s')Q_{\mathcal{M},\pi,\gamma}(s',a') \qquad (1.16)\\
&= (1-\gamma)r(s,a)+\gamma\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\mathbb{E}_{a'\sim\pi(\cdot|s')}[Q_{\mathcal{M},\pi,\gamma}(s',a')].
\end{aligned}
$$

Proof. This identities follow straightforwardly from Proposition 1.18. The first follows by replacing (1.14) in (1.13) while the second by replacing (1.13) in (1.14). □

It is also useful to define another type of value function.

Definition 1.21. Given and MDP $\mathcal{M}$, a policy π and a discount factor γ, the **advantage function** $A_{\mathcal{M},\pi,\gamma} : \mathcal{S} \times \mathcal{A} \mapsto [0,1]$ is

$$A_{\mathcal{M},\pi,\gamma}(s,a) := Q_{\mathcal{M},\pi,\gamma}(s,a) - V_{\mathcal{M},\pi,\gamma}(s). \tag{1.17}$$

for every $s \in \mathcal{S}$, $a \in \mathcal{A}$.

The advantage is a measure of how much a local change in the action selection influence the total return, in fact it is exactly defined as the difference between value following π and value taking action a and then following π.

1.2.2 Optimality

Given an MDP, solving a reinforcement learning task consists in, roughly, finding a policy that achieves a lot of reward over the long run. For finite MDPs, we can precisely define an optimal policy in the following way. Value functions define a partial ordering over policies. A policy π is defined to be better than or equal to a policy π' if its expected return is greater than or equal to that of π' for all states. In other words, $\pi \geq \pi'$ if and only if $V_\pi(s) \geq V_{\pi'}(s)$ for all $s \in \mathcal{S}$. It is a partial ordering because one policy may be better in some states and worse in others.

Theorem 1.22. *Given an MDP $\mathcal{M}$ and a discount factor γ, there is always at least one policy $\pi^* \in \Pi_{\mathcal{M}}$ that is better than or equal to all other policies. We call it optimal policy.*

In order to prove Theorem 1.22 we first need some results on *optimal value function.*

Definition 1.23. Given and MDP $\mathcal{M}$ and a discount factor γ, the **optimal value function** $V^*_{\mathcal{M},\gamma} : \mathcal{S} \mapsto [0,1]$ is

$$V^*_{\mathcal{M},\gamma}(s) := \sup_{\pi \in \Pi_{\mathcal{M}}} V_{\mathcal{M},\pi,\gamma}(s) \tag{1.18}$$

Definition 1.24. Given and MDP $\mathcal{M}$ and a discount factor γ, the **optimal state-action value function** $Q^*_{\mathcal{M},\gamma} : \mathcal{S} \times \mathcal{A} \mapsto [0,1]$ is

$$Q^*_{\mathcal{M},\gamma}(s,a) := \sup_{\pi \in \Pi_{\mathcal{M}}} Q_{\mathcal{M},\pi,\gamma}(s,a) \tag{1.19}$$

The supremum being made over a bounded space $\Pi_{\mathcal{M}}$ immediately implies the existence of a maximum.

Note that the policies that achieves the maximums are state and state-action dependant, respectively. There is no guarantees that a single policy achieves the maximum in every state and action, at least for now.

Proposition 1.25. Given an MDP $\mathcal{M}$ and a discount factor γ, the optimal value function $V^*_{\mathcal{M},\gamma}$ and the optimal action-value function $Q^*_{\mathcal{M},\gamma}$ satisfy

$$V^*_{\mathcal{M},\gamma}(s) = \max_{a\in\mathcal{A}} Q^*_{\mathcal{M},\gamma}(s,a), \tag{1.20}$$

for all $s \in \mathcal{S}$

Proof. We prove the equality by prooving separately the two inequalities. The first inequality comes from using (1.13) inside the maximum

$$\begin{aligned} V^*_{\mathcal{M},\gamma}(s) &= \max_{\pi\in\Pi_{\mathcal{M}}} V_{\mathcal{M},\pi,\gamma}(s) = \\ &= \max_{\pi\in\Pi_{\mathcal{M}}} \left(\mathbb{E}_{a\sim\pi(\cdot|s)}[Q_{\mathcal{M},\pi,\gamma}(s,a)]\right) \leq \\ &\leq \max_{\pi\in\Pi_{\mathcal{M}}} \left(\max_{a\in\mathcal{A}} \left(Q_{\mathcal{M},\pi,\gamma}(s,a)\right)\right) = \\ &= \max_{a\in\mathcal{A}} \left(\max_{\pi\in\Pi_{\mathcal{M}}} \left(Q_{\mathcal{M},\pi,\gamma}(s,a)\right)\right) = \max_{a\in\mathcal{A}} \left(Q^*_{\mathcal{M},\gamma}(s,a)\right). \end{aligned}$$

The second inequality comes by defining a deterministic policy $\pi^{\max} : \mathcal{S} \to \mathcal{A}$ by

$$\pi^{\max}(s) :\in \arg\max_{a\in\mathcal{A}} \left(Q^*_{\mathcal{M},\gamma}(s,a)\right)$$

and, slightly abusing notation by writing the expectation with respect to $a \sim \pi^{\max}(s)$, we have

$$\begin{aligned} \max_{a\in\mathcal{A}} \left(Q^*_{\mathcal{M},\gamma}(s,a)\right) &= Q^*_{\mathcal{M},\gamma}(s,\pi^{\max}(s)) = \\ &= \max_{\pi\in\Pi_{\mathcal{M}}} \left(Q_{\mathcal{M},\pi,\gamma}(s,\pi^{\max}(s))\right) = \\ &= \max_{\pi\in\Pi_{\mathcal{M}}} \left(\mathbb{E}_{a\sim\pi^{\max}(s)}[Q_{\mathcal{M},\pi,\gamma}(s,a)]\right) \leq \\ &\leq \max_{\pi\in\Pi_{\mathcal{M}}} \left(\mathbb{E}_{a\sim\pi(\cdot|s)}[Q_{\mathcal{M},\pi,\gamma}(s,a)]\right) = \\ &= \max_{\pi\in\Pi_{\mathcal{M}}} \left(V_{\mathcal{M},\pi,\gamma}(s)\right) = V^*_{\mathcal{M},\gamma}(s). \end{aligned}$$

□

Proposition 1.26. Given an MDP $\mathcal{M}$ and a discount factor γ, the optimal value function $V^*_{\mathcal{M},\gamma}$ and the optimal action-value function $Q^*_{\mathcal{M},\gamma}$ satisfy

$$Q^*_{\mathcal{M},\gamma}(s,a) = (1-\gamma)r(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)}[V^*_{\mathcal{M},\gamma}(s')], \tag{1.21}$$

for all $a \in \mathcal{A}$ and $s \in \mathcal{S}$.

Proof.

$$\begin{aligned}
Q^*_{\mathcal{M},\gamma}(s,a) &= \max_{\pi \in \Pi_{\mathcal{M}}} Q_{\mathcal{M},\pi,\gamma}(s,a) = \\
&= \max_{\pi \in \Pi_{\mathcal{M}}} \left((1-\gamma)r(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)}[V_{\mathcal{M},\pi,\gamma}(s')]\right) = \\
&= (1-\gamma)r(s,a) + \gamma \max_{\pi \in \Pi_{\mathcal{M}}} \left(\mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)}[V_{\mathcal{M},\pi,\gamma}(s')]\right) \leq \\
&\leq (1-\gamma)r(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \left[\max_{\pi \in \Pi_{\mathcal{M}}} V_{\mathcal{M},\pi,\gamma}(s')\right] = \\
&= (1-\gamma)r(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \left[V^*_{\mathcal{M},\gamma}(s')\right]
\end{aligned}$$

The second inequality comes by using again the deterministic policy $\pi^{\max} : \mathcal{S} \to \mathcal{A}$

$$\pi^{\max}(s) :\in \arg\max_{a \in \mathcal{A}} \left(Q^*_{\mathcal{M},\gamma}(s,a)\right)$$

First, note that for any state $s_0 \in \mathcal{S}$ it holds

$$\begin{aligned}
V^*_{\mathcal{M},\gamma}(s_0) &= \max_{a \in \mathcal{A}} Q^*_{\mathcal{M},\gamma}(s_0,a) = \\
&= \mathbb{E}_{a \sim \pi^{\max}(\cdot|s_0)}[Q^*_{\mathcal{M},\gamma}(s_0,a)] = \\
&= \mathbb{E}_{\tau \sim \mathbb{P}(\cdot|\pi^{\max},\mathcal{M},s_0)}[Q^*_{\mathcal{M},\gamma}(s_0,a_0)] \leq \\
&\leq \mathbb{E}_{\tau \sim \mathbb{P}(\cdot|\pi^{\max},\mathcal{M},s_0)}[(1-\gamma)r(s_0,a_0) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s_0,a_0)} \left[V^*_{\mathcal{M},\gamma}(s')\right]] = \\
&= \mathbb{E}_{\tau \sim \mathbb{P}(\cdot|\pi^{\max},\mathcal{M},s_0)}[(1-\gamma)r(s_0,a_0) + \gamma V^*_{\mathcal{M},\gamma}(s_1)],
\end{aligned}$$

where the inequality follows from the first part of the proof while the equalities are from definitions. Iterating this inequality inside itself leads to

$$\begin{aligned}
V^*_{\mathcal{M},\gamma}(s) &\leq \mathbb{E}_{\tau \sim \mathbb{P}(\cdot|\pi^{\max},\mathcal{M},s_0=s)}[(1-\gamma)r(s_0,a_0) + \gamma V^*_{\mathcal{M},\gamma}(s_1)] \leq \\
&\leq \mathbb{E}_{\tau \sim \mathbb{P}(\cdot|\pi^{\max},\mathcal{M},s_0=s)}[(1-\gamma)r(s_0,a_0) + \\
&\qquad + \gamma\big(\,\mathbb{E}_{\tau' \sim \mathbb{P}(\cdot|\pi^{\max},\mathcal{M},s'_0=s_1)}[(1-\gamma)r(s'_0,a'_0) + \gamma V^*_{\mathcal{M},\gamma}(s'_1)]\,\big) = \\
&= \mathbb{E}_{\tau \sim \mathbb{P}(\cdot|\pi^{\max},\mathcal{M},s_0=s)}\left[(1-\gamma)r(s_0,a_0) + \gamma\left((1-\gamma)r(s_1,a_1) + \gamma V^*_{\mathcal{M},\gamma}(s_2)\right)\,\right] = \\
&\leq \mathbb{E}_{\tau \sim \mathbb{P}(\cdot|\pi^{\max},\mathcal{M},s_0=s)}[(1-\gamma)r(s_0,a_0) + \gamma(1-\gamma)r(s_1,a_1) + \gamma^2 V^*_{\mathcal{M},\gamma}(s_2)],
\end{aligned}$$

where $\tau = (s_0, a_0, s_1, ...)$ and $\tau' = (s'_0, a'_0, s'_1, ...)$ and the equality follows thanks to the tower property of the expected value. Iterating for H times leads to

$$V^*_{\mathcal{M},\gamma}(s) \leq \mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi^{\max},\mathcal{M},s_0=s)}\left[(1-\gamma)\sum_{t=0}^{H-1}\gamma^t r(s_t,a_t) + \gamma^H V^*_{\mathcal{M},\gamma}(s_H)\right].$$

In the limit $H \to \infty$ the term $\gamma^H V^*_{\mathcal{M},\gamma}(s_H) \to 0$ thanks to the boundedness of value function and the choice $\gamma < 1$. This finally leads to

$$V^*_{\mathcal{M},\gamma}(s) \leq \mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi^{\max},\mathcal{M},s_0=s)}\left[(1-\gamma)\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = V_{\mathcal{M},\pi^{\max},\gamma}(s) \tag{1.22}$$

Conlcusion is straightforward with

$$\begin{aligned} Q^*_{\mathcal{M},\gamma}(s,a) \geq Q_{\mathcal{M},\pi^{\max},\gamma}(s,a) = \\ &= (1-\gamma)r(s,a) + \gamma\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V_{\mathcal{M},\pi^{\max},\gamma}(s')] \geq \\ &\geq (1-\gamma)r(s,a) + \gamma\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}[V^*_{\mathcal{M},\gamma}(s')] \end{aligned}$$

and this complete the proof. □

We can finally derive the **Bellman optimality equations**.

Proposition 1.27. For every MDP $\mathcal{M}$ and discount factor γ, the optimal value function $V^*_{\mathcal{M},\gamma}$ satisfies the Bellman optimality equation for every $s \in \mathcal{S}$

$$\begin{aligned} V^*_{\mathcal{M},\gamma}(s) &= \max_{a\in\mathcal{A}} \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\left[(1-\gamma)r(s,a) + \gamma V^*_{\mathcal{M},\gamma}(s')\right] \\ &= \max_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a)\left((1-\gamma)r(s,a) + \gamma V^*_{\mathcal{M},\gamma}(s')\right). \end{aligned} \tag{1.23}$$

Proposition 1.28. For every MDP $\mathcal{M}$ and discount factor γ, the optimal state-action value function $Q^*_{\mathcal{M},\gamma}$ satisfies the Bellman optimality equation for every $s \in \mathcal{S}$ and $a \in \mathcal{A}$

$$\begin{aligned} Q^*_{\mathcal{M},\gamma}(s,a) &= \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\left[(1-\gamma)r(s,a) + \gamma\max_{a'\in\mathcal{A}} Q^*_{\mathcal{M},\gamma}(s',a')\right] \\ &= \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a)\left((1-\gamma)r(s,a) + \gamma\max_{a'\in\mathcal{A}} Q^*_{\mathcal{M},\gamma}(s',a')\right). \end{aligned} \tag{1.24}$$

Proof. This identities follow straightforwardly from Proposition 1.25 and Proposition 1.26 combined. □

Remark. A more elegant (and thus more common in literature) proof of Bellman optimality equations uses the right hand side of (1.23) as definition of an operator from value functions to value functions, this operator can be shown to be a contraction in a complete space. Thus existence and uniqueness follows from Banach-Caccioppoli fixed point Theorem. We chose this more constructive demonstration hoping to leave some intuition.

Theorem 1.29. *Given an MDP $\mathcal{M}$ and a discount factor γ, there is always at least one deterministic policy $\pi^* \in \Pi^D_{\mathcal{M}}$ that is better than or equal to all other policies.*

Proof. In the proof of Proposition 1.26 we actually prooved (1.22), that is

$$V_{\mathcal{M},\pi^{\max},\gamma}(s) \geq V^*_{\mathcal{M},\gamma}(s)$$

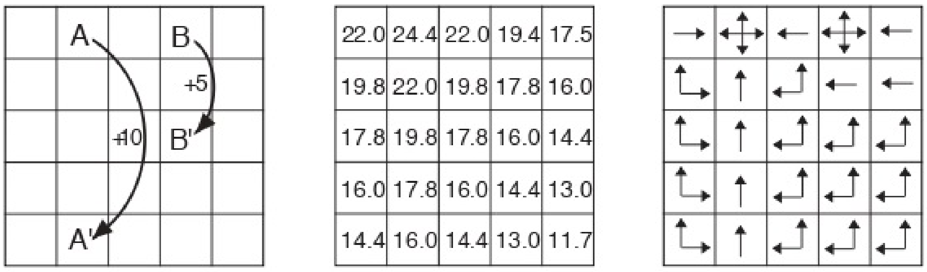
for every state $s \in \mathcal{S}$. Thus $\pi^{\max} = \pi^*$ is an optimal policy. □

This latter result cleary implies also Theorem 1.22. It also implies that all optimal policies share the same value function $V^*_{\mathcal{M},\gamma}$.

Theorem 1.30. *Given a MDP $\mathcal{M}$ and a discount factor γ, a policy $\pi \in \Pi_{\mathcal{M}}$ is an optimal policy if and only if for every $s \in \mathcal{S}$*

$$\pi(a|s) \neq 0 \iff a \in \arg\max_{a' \in \mathcal{A}} Q^*_{\mathcal{M},\gamma}(s, a') \tag{1.25}$$

Example 1.31. Gridworld (part 2) Recall the Gridworld task shown in Figure 1.8(a) and firstly introduced in Example 1.17. State A is followed by a reward of +10 and a transition to state A', while state B is followed by a reward of +5 and a transition to state B'.



22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

(a) (b) (c)

Figure 1.8: Gridworld example: dynamics (a), optimal value function (b), optimal policy (c).

The optimal value function $V^*_{\mathcal{M},\gamma}(s)$, shown in Figure 1.8(b) for every cell $s \in \mathcal{S}$, can be computed exactly solving the system of the $|\mathcal{S}|$ Bellman optimality equations (1.23) since we know explicitely the dynamics $\mathcal{P}(\cdot|s,a)$ of the MDP. Figure 1.8(c) shows the corresponding optimal policies. Where there are multiple arrows in a cell, all of the corresponding action are optimal, this set of optimal actions is exactly the $\arg\max_{a\in\mathcal{A}} Q^*_{\mathcal{M},\gamma}(s,a')$ set of Theorem 1.30, for every cell $s \in \mathcal{S}$.

1.2.3 Error measure

Now that we have defined and, in some way, characterized what an optimal policy is, it is clear that our aim is to find one of them. Concretely this is unfeasible except for very small or particular MDPs. What we aim to find instead is an *almost optimal policy.* This almostness is quantified in terms of an error parameter ϵ, and we will say ϵ-optimal policies.

We have defined value functions and state visitation distribution according to an initial state $s_0 \in \mathcal{S}$. It is straightforward to extend this definitions to the case where s_0 is not fixed but choosen accordingly to a distribution ρ. We overload the previous notation by defining

$$d^{\mathcal{M},\pi,\gamma}_{\rho}(s,a) := \mathbb{E}_{s_0\sim\rho}[d^{\mathcal{M},\pi,\gamma}_{s_0}(s,a)] \qquad \forall(s,a)\in\mathcal{S}\times\mathcal{A} \tag{1.26}$$

$$V_{\mathcal{M},\pi,\gamma}(\rho) := \mathbb{E}_{s_0\sim\rho}[V_{\mathcal{M},\pi,\gamma}(s_0)] \tag{1.27}$$

$$V^*_{\mathcal{M},\gamma}(\rho) := \mathbb{E}_{s_0\sim\rho}[V^*_{\mathcal{M},\gamma}(s_0)] \tag{1.28}$$

Now we can define the *error function.*

Definition 1.32. Given a MDP $\mathcal{M}$, a discount factor γ and a distribution ρ over the state space $\mathcal{S}$. The **error** is a function $err_{\mathcal{M},\gamma,\rho} : \Pi_{\mathcal{M}} \to \mathbb{R}$, defined by

$$err_{\mathcal{M},\gamma,\rho}(\pi) := V^*_{\mathcal{M},\gamma}(\rho) - V_{\mathcal{M},\pi,\gamma}(\rho) \tag{1.29}$$

Actually we will make use of iterative methods with intrinsic stochasticity, and so the error function of a single policy, *i.e.* on a single iteration, may be arbitrary big and unboundable. This problem is commonly avoided in litterature by considering the sum of the errors over iterations, this is clearly a more stable measure and it is called *regret*.

Definition 1.33. Given a MDP $\mathcal{M}$, a discount factor γ, a distribution ρ over the state space $\mathcal{S}$ and a series of policies $(\pi^{(k)})_{k\in\mathbb{N}}$, the **regret** after K iterations is a function $regret_{\mathcal{M},\gamma,\rho}(\cdot\,;K) : (\Pi_{\mathcal{M}})^K \to \mathbb{R}$, defined by

$$regret_{\mathcal{M},\gamma,\rho}((\pi^{(k)})_{k<K};K) := \sum_{k=0}^{K-1}\left(V^*_{\mathcal{M},\gamma}(\rho) - \mathbb{E}_k[V_{\mathcal{M},\pi^{(k)},\gamma}(\rho)]\right) \tag{1.30}$$

where $\mathbb{E}_k[\cdot]$ is the expectation with respect to the k-th iteration

In our case, the sequence of policies $(\pi^{(k)})_{k\in\mathbb{N}}$ is produced by an iterative method, the regret is then a measure of the error of that iterative method.

1.3 Model

In order to make Definition 1.9 of an **episode** rigorous, we need formal definitions of *sampling models* and *termination conditions.* We stated rigorously how the agent-environment system evolves in successive timesteps, we now have to state how the system starts and how it ends. Once we know exaclty what an episode is, we will also be able to state a formal definition of *sample efficiency.*

1.3.1 Sampling models

A sampling model is intuitively the interface in which the environment is presented to the agent.

Definition 1.34. A **sampling model** for a Markov Decision Process $\mathcal{M}$ is a pair consisting of a boolean and a randomized algorithm. The boolean states when or not the agent has access to launch the randomized algorithm. The randomized algorithm, when launched, returns a state $s_0 \in \mathcal{S}$ and a realization of $\mathcal{M}$ in which the agent is at s_0 and can further on interact with the environment. This randomized algorithm is called *reset.*

There are three types of idealized sampling models ranging from the most general to the most powerful from the point of view of the agent, *i.e.* our point of view, with a lot of possible shadows between them:

- The most general sampling model is the **online simulation model**. Here *reset* always returns the same $s_0 \in \mathcal{S}$ and, thus, the same realization of $\mathcal{M}$. Furthermore, the agent has no access to launch *reset.*
- An intermediate setting is the **μ-reset model**. Here the returned s_0 is randomly drawn according to a fixed distribution μ on the state space $\mathcal{S}$ and the agent has access to *reset* whenever he wants. There are important differences, as we will see in the next Chapters in deeper details, based on the properties of μ.
- A considerably more powerful sampling model is the **generative model**, it was introduced by [4]. Here again the agent has access to *reset* whenever he wants and, furthermore, the *reset* algorithm accepts a state $s \in \mathcal{S}$ as input and returns the realization of $\mathcal{M}$ with the agent in s. In a lot of real application this turns out to be a natural assumption, such as in the case in which we have a physical simulator of the environment.

It is worth noticing, even without further details which are useless for the scope of this work, that the μ-reset model becomes very related to the generative model if μ is a uniform distribution over the state space while it becomes very related to the online simulation model if μ is supported only on one state, meaning $\mu(s_0) = 1$ and $\mu(s) = 0$ for all $s \neq s_0$.

Remark. We want to point out the difference between the distribution μ of the μ-reset model and the distribution ρ involved in Definition 1.32 of the error. They are both distribution on the state space $\mathcal{S}$ but the former is intrinsic in the given sampling model and is thus indepentent from our control, the latter on the other hand can be choosen and changed according to our preferences and to what we aim to measure. The most natural choice is to consider the error given by $\rho = \mu$ and so those concepts are, more often than not, confused together in litterature.

1.3.2 Termination conditions

The choice of the trajectory length H in an episode (Definition 1.9) happens by termination. This means that the agent-environment system evolves in successive timesteps according to the already stated rules until some termination condition is satisfied, this termination condition directly correspond to a call of the *reset* function of the sampling model.

The termination condition can be satisfied by:

- The agent, calling the *reset* function itself, if the sampling model allows him to do so;
- The environment, calling the *reset* function every time the agent reaches a terminal state.

Definition 1.35. In a Markow Decision Process $\mathcal{M}$, the **terminal states** $T \subseteq S$ are special states. If the agent arrives at a terminal state then the environment launches the *reset* algorithm of the sampling model. No matter what the agent do, if he arrives at a terminal state the flow of experience is interrupted and restarted.

Remark. Given a μ-reset model or a generative model, we can assume to have no terminal states in our Markow Decision Processes. This leads to no loss in generality.

Proof. Suppose we have an MDP $\mathcal{M}$ with a non empty set $T \subset S$ of terminal states. We can consider a new MDP $\mathcal{M}'$ that is the same of $\mathcal{M}$ except for the transitions out of the terminal states, that always leads to the terminal state itself, and for the rewards in those transitions, that are zero. This idea is shown in Figure 1.9 in order to help intuition. Namely $\mathcal{M}' := (\mathcal{S}, \mathcal{A}, \mathcal{P}', r')$ such that $\mathcal{P}'(s|s,a) = 1$ and $r'(s,a) = 0$ for every $(s,a) \in T \times \mathcal{A}$. It is straightforward to see that all the value functions and optimal policies remains exaclty the same. □

Assuming to have no terminal states avoids countinuously considering annoying particular cases. This is very helpful for clarity of presentation while, as Remark shows, leads to no loss in the generality.

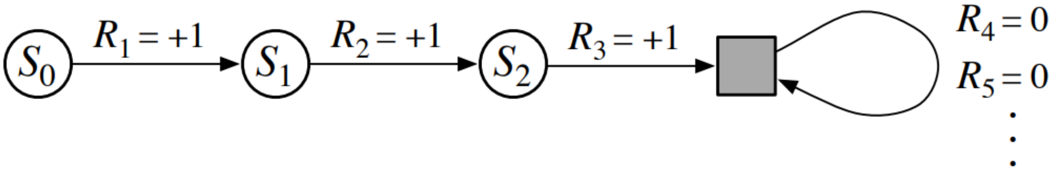


Figure 1.9: Transition graph with a terminal state.

1.3.3 Knowledge of the environment

Another crucial distinction we have to make is on our level of knowledge of the environment. As before, even if there can be infinite possible intermediate shadows, we focus our attention to the extreme cases.

In the **totally known environment** setting we have full access to the environment dynamics, from the rewards $r(s, a)$ to the transition probabilities $\mathcal{P}(s'|s, a)$ for any $s, s' \in \mathcal{S}$ and $a \in \mathcal{A}$. Here, from a supervised learning point of view, since the target function is known then the problem is *already solved*. For finite state and spaces MDPs the Bellman optimality equation

$$V^*_{\mathcal{M},\gamma}(s) = \max_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s, a) \left((1-\gamma)r(s, a) + \gamma V^*_{\mathcal{M},\gamma}(s')\right)$$

is actually a system of equations, one for each state, so there are $|\mathcal{S}|$ equations in $|\mathcal{S}|$ unknowns. In principle one can solve this system of equations for $V^*_{\mathcal{M},\gamma}$ using any one of a variety of methods for solving systems of nonlinear equations. Once one has $V^*_{\mathcal{M},\gamma}$, it is relatively easy to determine an optimal policy.
In big scale MDPs with huge state space, Monte Carlo simulations is often the most tractable way to manipulate models, thus most optimization techniques are simulation based.

In the **totally unknown environment** setting the dynamics of the environment are totally unknown and can only be observed and, at the best, estimated through their realizations during experience flow. In this setting, solving methods split in two main categories:

- In **model-free** methods the agent does not use predictions of the environment dynamics. They rely on real samples from the environment and never use generated predictions of next state and next reward to alter behaviour.
- In **model-based** methods the agents use predictions of the environment dynamics, whilst during learning or acting. Here a trained agent can make approximated predictions about what the next state and reward will be before it takes each action.

1.3.4 Sample efficiency

The sample complexity is, informally, the amount of experience required to satisfy a specific performance criterion. In supervised learning it is well defined the notion of *sample complexity*.

Definition 1.36. In supervised learning the **sample complexity** $N(\epsilon, \delta)$ of an algorithm $\mathfrak{A}$ is the minimum *number of samples* $n \in \mathbb{N}$ such that

$$\mathbb{P}\Big(\text{error}\big(\mathfrak{A} \text{ with } n \text{ samples}\big) \geq \epsilon\Big) < \delta,$$

where the error is usually measured as the expectation of some appropriate loss function.

Unlike in supervised learning, there is as of yet no universal definition in the literature of the sample complexity for reinforcement learning. We already stated two error measures (1.29) and (1.30), those are commonly accepted standards in literature. It comes out that the definition of sample complexity in reinforcement learning must depend on the sampling model by the way we count the *number of samples*. This is because definitions that are good for some sampling models are less meaningful for others. Once defined this counting strategy, Definition 1.36 applies also to our Reinforcement Learning context.

Here we present two standards proposed by [3] for counting the number of samples:

- In the online simulation model the agent must follow one unbroken chain of experience for some number of *decision epochs*, a decision epoch is just a timestep in which a state is observed and an action is taken. The most natural choice here is exactly the number of decision epochs, the number of action we make.

- In the generative model, in which we can always choose when to stop and where to restart from, aiming at minimizing the number of decision epochs would lead to one-timestep-long trajectories because they maintain the same complexity while giving more choice, and more power along, to the agent. Here a more natural definition is the number of calls to the *reset* function, *i.e.* the number of episodes.

In the intermediate μ-reset model both the previous definitions can be used, without relevant differences. We choose to focus on the second one in this work.

Finally, the **sample efficiency** is how the sample complexity scales with the relevant problem dependent parameters. In the case of Reinforcement Learning with stationary policies those relevant parameters are: the size of the state space $\mathcal{S}$, the size of the action space $\mathcal{A}$, the discount factor γ and others, like the *distribution mismatch coefficient* (3.6).

Chapter 2

Learning methods

Here and in the following we always assume the setting with a μ-reset model, on various distributions μ, and totally unknown environment dynamics. We consider only iterative methods, since other optimization techniques require some sort of knowledge of the environment dynamics, which we do not have, or particular structure hypotheses, which we avoid in order to maintain the generality of the results. Other special non-iterative techniques may exist but are beyond the scope of this work.

The key idea of reinforcement learning is to use value functions to organize and structure the search of good policies. This usage techniques in iterative methods mainly split in two macro categories: value based methods and policy based methods. In this Chapter we give a brief introduction of both.

2.1 Action-value methods

The main idea behind the action-value methods can conceptually be splitted in two parts:

- Policy evaluation, or *prediction*;
- Policy improvement, or *control.*

The learning process happens through alternation of those two phases, each using the other to guarantee improvement in performance. *Control* use $V_{\mathcal{M},\pi,\gamma}$ to yield a better policy π', then *prediction* compute $V_{\mathcal{M},\pi',\gamma}$ from π' and then *control* improve it again to yield an even better policy. We can thus obtain a sequence of monotonically improving policies and value functions

$$\pi_0 \xrightarrow{P} V_{\mathcal{M},\pi_0,\gamma} \xrightarrow{C} \pi_1 \xrightarrow{P} V_{\mathcal{M},\pi_1,\gamma} \xrightarrow{C} \pi_2 \xrightarrow{P} \quad ... \quad \xrightarrow{C} \pi^* \xrightarrow{P} V_{\mathcal{M},\pi^*,\gamma}.$$

Once this process arrives at a stable point, it is guaranteed to be an optimum thanks to Theorem 2.1, the Policy Improvement Theorem. Intuition is given by Figure 2.1.

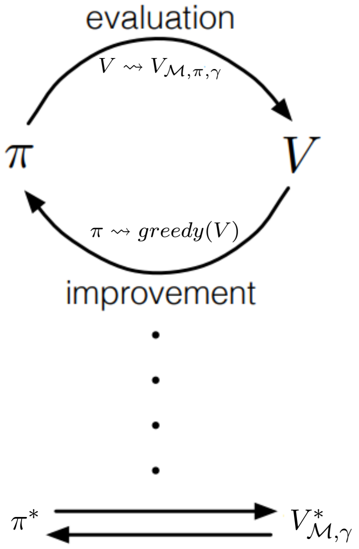

evaluation
$V \rightsquigarrow V_{\mathcal{M},\pi,\gamma}$
π
V
$\pi \rightsquigarrow greedy(V)$
improvement
π^*
$V^*_{\mathcal{M},\gamma}$

Figure 2.1: Value-based methods structure.

Those two phases are better explained when seen independently one from each other. This separation also helps developing an intuition on the link between value functions and action selection that will be useful in the next Chapters.

2.1.1 Prediction

First we consider how to estimate the state value function $V_{\mathcal{M},\pi,\gamma}$ for an arbitrary policy $\pi \in \Pi_{\mathcal{M}}$. In literature this process is called policy prediction.

There are several different ways this can be done, starting from solving the system of $|\mathcal{S}|$ non-linear equations as we have shown in Section 1.3.3 when the environment dynamics is known. We are not interest in covering all of them, despite the fact that there would be a lot to say.

For our purpose, iterative methods are the most suitable, and for completeness we show one of them. The initial approximation $V_0 : \mathcal{S} \to \mathbb{R}$ is chosen arbitrarily (except that the terminal state, if any, must be given value 0), and each successive approximation is obtained by using the Bellman equation (1.15) as an update rule

$$V_{k+1}(s) : = \mathbb{E}_{a \sim \pi(\cdot|s)} \left[(1-\gamma) r(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} [V_k(s')] \right],$$

for all $s \in \mathcal{S}$. Clearly $V_k = V_{\mathcal{M},\pi,\gamma}$ is a fixed point for this udate rule because of the Bellman equation for $V_{\mathcal{M},\pi,\gamma}$. Indeed, the sequence $\{V_k\}$ can be shown in general to converge to $V_{\mathcal{M},\pi,\gamma}$ as $k \to \infty$, for the complete proof we refer to [7]. This algorithm is called *iterative policy* *evaluation.* Notice that all the updates are based on an expectation over all possible next states rather than on a sampled next state.

Here a key observation is that, except for particular cases, an exact evaluation of a policy requires infinitely many samples.

2.1.2 Control

The reason for computing the value function for a policy is to help find better policies. Indeed knowing the exact value function easily leads to a better policy. This improvement is guaranteed by the Policy Improvement Theorem.

Theorem 2.1. ***Policy Improvement Theorem***
Given a MDP $\mathcal{M}$ and a discount factor γ, let π and π' be two policies such that, for all $s \in \mathcal{S}$,

$$\mathbb{E}_{a\sim\pi'(\cdot|s)}[Q_{\mathcal{M},\pi,\gamma}(s,a)] \geq V_{\mathcal{M},\pi,\gamma}(s). \tag{2.1}$$

Then the policy π' must be as good as, or better than, π. That is, it must obtain greater or equal expected return from all states $s \in \mathcal{S}$:

$$V_{\mathcal{M},\pi',\gamma}(s) \geq V_{\mathcal{M},\pi,\gamma}(s) \tag{2.2}$$

Moreover, if there is a strict inequality in (2.1) for at least one state, then there must be a strict inequality in (2.5) at at least one state.

Proof. Thanks to the hypothesis, we can relate the value function of π in a state s with the value function of π in the successor states s' sampled according to π', that is for every $s_0 \in \mathcal{S}$

$$\begin{aligned} V_{\mathcal{M},\pi,\gamma}(s_0) \leq \mathbb{E}_{a\sim\pi'(\cdot|s_0)}[Q_{\mathcal{M},\pi,\gamma}(s_0,a)] = \\ &= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot,\pi',\mathcal{M},s_0)}\left[Q_{\mathcal{M},\pi,\gamma}(s_0,a_0)\right] = \\ &= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot,\pi',\mathcal{M},s_0)}\left[(1-\gamma)r(s_0,a_0) + \gamma\mathbb{E}_{s'\sim\mathcal{P}(\cdot|s_0,a_0)}[V_{\mathcal{M},\pi,\gamma}(s')]\right] = \\ &= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot,\pi',\mathcal{M},s_0)}\left[(1-\gamma)r(s_0,a_0) + \gamma V_{\mathcal{M},\pi,\gamma}(s_1)\right], \end{aligned}$$

where we used respectively: the hypothesis, the definition of probability distribution over trajectories, the Bellman equation (1.14) and the tower property of the expected value. Recall for clarity that the trajectory $\tau = (s_0, a_0, s_1, ...)$ always stands for the sequence of state-action pairs.

We can then apply the same argument to $V_{\mathcal{M},\pi,\gamma}(s_1)$ because the hypothesis holds true for every state.

$$
\begin{aligned}
\mathbb{E}_{\tau\sim\mathbb{P}(\cdot,\pi',\mathcal{M},s_0)}\left[\gamma V_{\mathcal{M},\pi,\gamma}(s_1)\right] &\leq \mathbb{E}_{\tau\sim\mathbb{P}(\cdot,\pi',\mathcal{M},s_0)}\left[\gamma \mathbb{E}_{a\sim\pi'(\cdot|s_1)}[Q_{\mathcal{M},\pi,\gamma}(s_1,a)]\right] = \\
&= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot,\pi',\mathcal{M},s_0)}\left[\gamma Q_{\mathcal{M},\pi,\gamma}(s_1,a_1)\right] = \\
&= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot,\pi',\mathcal{M},s_0)}\left[\gamma(1-\gamma)r(s_1,a_1) + \gamma^2 \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s_1,a_1)}[V_{\mathcal{M},\pi,\gamma}(s')]\right] = \\
&= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot,\pi',\mathcal{M},s_0)}\left[\gamma(1-\gamma)r(s_1,a_1) + \gamma^2 V_{\mathcal{M},\pi,\gamma}(s_2)\right].
\end{aligned}
$$

Iterating this argument infinitely many times leads to

$$
\begin{aligned}
V_{\mathcal{M},\pi,\gamma}(s_0) &\leq \mathbb{E}_{\tau\sim\mathbb{P}(\cdot,\pi',\mathcal{M},s_0)}\left[(1-\gamma)r(s_0,a_0) + \gamma V_{\mathcal{M},\pi,\gamma}(s_1)\right] \leq \\
&\leq \mathbb{E}_{\tau\sim\mathbb{P}(\cdot,\pi',\mathcal{M},s_0)}\left[(1-\gamma)r(s_0,a_0) + \gamma(1-\gamma)r(s_1,a_1) + \gamma^2 V_{\mathcal{M},\pi,\gamma}(s_2)\right] \\
&\leq \mathbb{E}_{\tau\sim\mathbb{P}(\cdot,\pi',\mathcal{M},s_0)}\Big[(1-\gamma)r(s_0,a_0) + \gamma(1-\gamma)r(s_1,a_1) + \gamma^2(1-\gamma)r(s_2,a_2) + \\
&\quad \vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad + \gamma^3 \underbrace{V_{\mathcal{M},\pi,\gamma}(s_3)}_{\cdots}\Big] \leq \\
&\leq \mathbb{E}_{\tau\sim\mathbb{P}(\cdot,\pi',\mathcal{M},s_0)}\left[(1-\gamma)\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = V_{\mathcal{M},\pi',\gamma}(s_0),
\end{aligned}
$$

for every $s_0 \in \mathcal{S}$. This completes the proof. □

Remark. The power of the Policy Improvement Theorem resides in the passage from a *local* property (2.1), namely where the difference in action selection is made only in the first step, to a *non local* property (2.5), namely where the action selection is different at every future timestep.

With this result it is quite natural to define the greedy control, selecting *deterministically* at each state the action that appears the best according to the action value function of π. Formally, in the control step we produce a new deterministic policy $\pi' = greedy(Q_{\mathcal{M},\pi,\gamma})$, given by

$$
\pi'(s) :\in \arg\max_{a\in\mathcal{A}} Q_{\mathcal{M},\pi,\gamma}(s,a). \tag{2.3}
$$

The greedy policy takes the action that looks the best in the short term - after one step of lookahead - according to $Q_{\mathcal{M},\pi,\gamma}$.

By construction, the greedy policy meets the conditions of the Policy Improvement Theorem 2.1, so we know that it is as good as, or better than, the previous policy.

Suppose that the new greedy policy π' is as good as, but not better than, the old policy π. Then $V_{\mathcal{M},\pi,\gamma} = V_{\mathcal{M},\pi',\gamma}$, and from (2.3) it follows that for all $s \in \mathcal{S}$:

$$
V_{\mathcal{M},\pi',\gamma}(s) = \max_{a\in\mathcal{A}} \mathbb{E}_{s'\sim\mathcal{P}(\cdot|s,a)}\left[(1-\gamma)r(s,a) + \gamma V_{\mathcal{M},\pi',\gamma}(s')\right].
$$

But this is the same as the Bellman optimality equation (1.23), and therefore $V_{\mathcal{M},\pi',\gamma}$ must be $V^*_{\mathcal{M},\gamma}$, and both π and π' must be optimal policies. Policy improvement thus gives a strictly better policy except when the original policy is already optimal.

One may also think of the interaction between the evaluation and improvement process in terms of two costrained goals. For example, as two lines in two-dimensional space as suggested by the diagram in Figure 2.2. Although the real geometry is much more complicated than this, the Figure suggests what happens. Each process drives the value function or policy toward one of the lines representing a solution to one of the two goals. Driving directly toward one goal causes some movement away from the other goal, however, the two processes together achieve the overall goal of optimality even though neither is attempting to achieve it directly. That is, in part, because the intersection of the two goals actually implies a solution of the Bellman optimality equations, as we have seen.

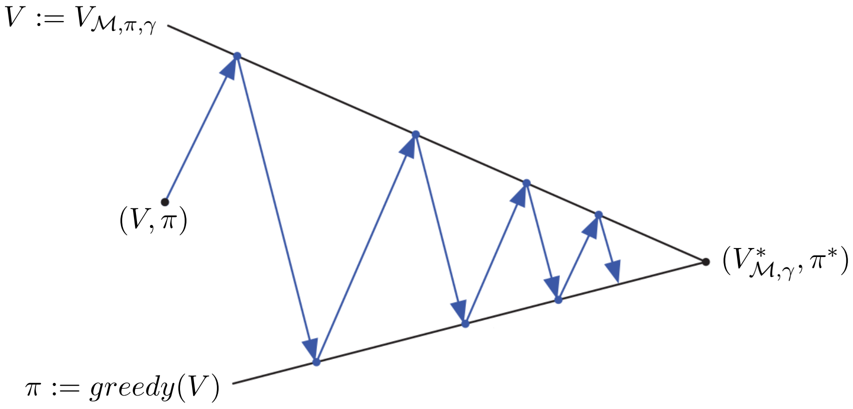

$V := V_{\mathcal{M},\pi,\gamma}$
(V, π)
$(V^*_{\mathcal{M},\gamma}, \pi^*)$
$\pi := greedy(V)$

Figure 2.2: Value-based control iteration

But since we are constrained to a sampled experience, in order to ensure a correct policy evaluation it becomes crucial the hypothesis of visiting every state infinitely many times, so that we can correctly evaluate their values. This hypothesis can be satisfied via a sampling model with *coverage hypothesis* (Definition 3.12) or, in the general case without a powerful sampling model, with a number of different methods. Perhaps the simplest idea to ensure continual exploration is by using ϵ-greedy policies instead of greedy ones, that is

$$[\epsilon\text{-greedy}(Q_{\mathcal{M},\pi,\gamma})]\,(a|s) := \begin{cases} \frac{\epsilon}{|\mathcal{A}|} + (1-\epsilon) & \text{if } a = \bar{a}(s) \\ \frac{\epsilon}{|\mathcal{A}|} & \text{otherwise} \end{cases} \tag{2.4}$$

for every $s \in \mathcal{S}$. Where $\bar{a} : \mathcal{S} \to \mathcal{A}$ is a deterministic function defined as

$$\bar{a}(s) :\in \arg\max_{a' \in \mathcal{A}} Q_{\mathcal{M},\pi,\gamma}(s, a').$$

The improvement Theorem still holds in a slightly modified version.

Theorem 2.2. ***ϵ-greedy Policy Improvement Theorem***
Let π be a policy that is ϵ-greedy with respect to some value function. Let $\pi' := \epsilon\text{-}greedy(Q_{\mathcal{M},\pi,\gamma})$. Then the policy π' must be as good as, or better than, π. That is, it must obtain greater or equal expected return from all states $s \in \mathcal{S}$:

$$V_{\mathcal{M},\pi',\gamma}(s) \geq V_{\mathcal{M},\pi,\gamma}(s). \tag{2.5}$$

Proof.

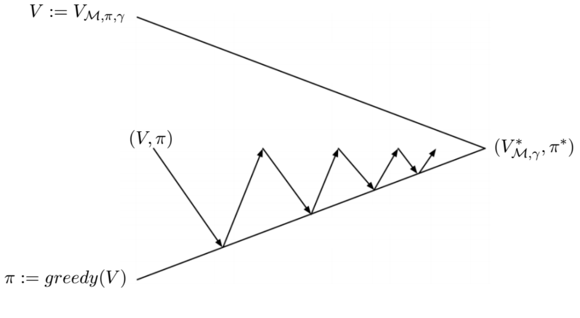
$$\begin{aligned}
\mathbb{E}_{a\sim\pi'(\cdot|s)}[Q_{\mathcal{M},\pi,\gamma}(s,a)] &= \sum_{a\in\mathcal{A}} \pi'(a|s) Q_{\mathcal{M},\pi,\gamma}(s,a) = \\
&= \left(\frac{\epsilon}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} Q_{\mathcal{M},\pi,\gamma}(s,a)\right) + (1-\epsilon)\max_{a\in\mathcal{A}} Q_{\mathcal{M},\pi,\gamma}(s,a) \geq \\
&\geq \frac{\epsilon}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} Q_{\mathcal{M},\pi,\gamma}(s,a) + (1-\epsilon)\sum_{a\in\mathcal{A}} \frac{\pi(a|s) - \frac{\epsilon}{|\mathcal{A}|}}{1-\epsilon} Q_{\mathcal{M},\pi,\gamma}(s,a) = \\
&= \sum_{a\in\mathcal{A}} \pi(a|s) Q_{\mathcal{M},\pi,\gamma}(s,a) = V_{\mathcal{M},\pi,\gamma}(s),
\end{aligned}$$

where we used the fact that $\frac{\pi(a|s)-\frac{\epsilon}{|\mathcal{A}|}}{1-\epsilon} \geq 0$ for every ϵ-greedy policy and $\sum_{a\in\mathcal{A}} \frac{\pi(a|s)-\frac{\epsilon}{|\mathcal{A}|}}{1-\epsilon} = 1$, so the max is greater than an expectation.

Thus we can apply the Policy Improvement Theorem which leads to $V_{\mathcal{M},\pi',\gamma}(s) \geq V_{\mathcal{M},\pi,\gamma}(s)$ for every $s \in \mathcal{S}$. This completes the proof. □

2.1.3 Generalized policy iteration

The evaluation and improvement processes can be viewed as both competing and cooperating. They compete in the sense that they pull in opposite directions. Making the policy greedy with respect to the value function typically makes the value function incorrect for the changed policy, and making the value function consistent with the policy tipically causes that policy no longer to be greedy. In the long run, however, these two processes interact to find a single joint solution: the optimal value function $V^*_{\mathcal{M},\gamma}$ and an optimal policy π^*.


$V := V_{\mathcal{M},\pi,\gamma}$
(V,π)
$(V^*_{\mathcal{M},\gamma},\pi^*)$
$\pi := greedy(V)$

The control cyclus consisting in a full evaluation and greedy improvement is called *policy iteration.* Since a true evaluation of a policy takes an infinite amount of steps, one can speed up policy iteration by stopping the evaluation after a fixed number of steps, or by estimating V or Q by any means. In the same way the improvement step can be made in a way different from greedy, for instance with the ϵ-greedy described before. In all this cases, one talks about *generalized policy iteration.* Every value-based methods works this way. The most famous examples are Sarsa and Q-learning, explained in details in [7]. The generalized policy iteration leads to better practical convergence behaviour but way worse theoretical convergence guarantees.

2.2 Policy gradient methods

Almost all the action-value methods learn the value of actions and then select actions based on their estimated action value. Their policies would not even exists without the action value estimates. In this Section we consider methods that, instead, learn a *parametrized policy* that can select actions without consulting a value function. A value function may still be used to learn the policy parameter, but is not required for action selection.

In particular we consider methods for learning the policy parameters based on the gradient of some scalar performance measure $J(\theta)$ with respect to the policy parameter. These methods seek to maximize performance, so their updates will be in general an approximate gradient ascent in J. This is defined in its most general form by the update rule

$$\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J(\theta_t)},$$

where $\widehat{\nabla J(\theta_t)}$ is a random variable whose expectation approximates the gradient of the performance measure $J(\theta_t)$ with respect to its argument θ. We use a stochastic estimate $\widehat{\nabla J(\theta_t)}$ of the gradient because, in general, the true gradient $\nabla J(\theta_t)$ is unknown.

The choice of the performance measure J is, in principle, totally arbitrary. But, recalling that we are interested in minimizing the error (1.29) and thus in maximizing the value function, it is reasonable that the performance measure contains the value function $V_{\mathcal{M},\pi,\gamma}$. If J also has other terms we will talk about *surrogate objective function*, see (4.1) for details. For the moment, we use as performance the value of a fixed, single state:

$$J(\theta) = V_{\mathcal{M},\pi_\theta,\gamma}(s_0).$$

2.2.1 Policy representation

A policy is a collection of distributions and we need a way to represent such distributions with a collection of parameters θ. Doing so, we will search our optimum in the class of parametric policies $\{\pi_\theta | \theta \in \Theta\}$.

A parametrization is **complete** if any stochastic policy can be represented, or **incomplete** otherwise. Incomplete policy classes are beyond the scope of this thesis since we want to focus on the optimal use of samples instead of approximation hypothesis. In fact, with incomplete parametrization the best we may hope is for an *agnostic* result where we do as well as the best policy in this class. The most commonly used incomplete parametrization are linear, log-linear and neural policy classes, those are explained in details in [2] but are beyond the scope of this work. Using complete parameteizations means that we are in the framework of tabular policy learning.

Here and in the following we always assume the MDPs to have finite state and action spaces.

We are going to focus on two types of complete parametrization:

- **Direct parametrization**: the policies are parametrized by
$$\pi_\theta(a|s) = \theta_{s,a},$$
where $\theta \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is subject to $\theta_{s,a} \geq 0$ and $\sum_{a\in\mathcal{A}} \theta_{s,a} = 1$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$.

- **Softmax parametrization**: the policies are parametrized by
$$\pi_\theta(a|s) = \frac{e^{\theta_{s,a}}}{\sum_{a'\in\mathcal{A}} e^{\theta_{s,a'}}},$$
where $\theta \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ is uncostrained.

Remark. Theoretically in policy gradient methods policies can be parametrized in any way as long as π_θ is differentiable with respect to its parameter. We focus in particular on the softmax parametrization because, as we will see, it is the most natural in some sense. We choose to state some results also with the direct parametrization for the intuition they give and for the problems that emerges from them.

It will be useful in the next Chapters to explicit the gradient of a policy π_θ with respect to its parameter θ. While for the direct parametrization this is trivial, for the softmax parametrization this is not. In the latter case, in fact, we have

$$\begin{aligned}
\frac{\partial \pi_\theta(a|s)}{\partial \theta_{s',a'}} &= \frac{\partial}{\partial \theta_{s',a'}} \frac{e^{\theta_{s,a}}}{\sum_{b\in\mathcal{A}} e^{\theta_{s,b}}} = \\
&= \mathbb{1}_{s=s'} \left(\frac{\partial}{\partial \theta_{s,a'}} \frac{e^{\theta_{s,a}}}{\sum_{b\in\mathcal{A}} e^{\theta_{s,b}}} \right) = \\
&= \mathbb{1}_{s=s'} \left(\frac{\frac{\partial}{\partial \theta_{s,a'}} e^{\theta_{s,a}}}{\sum_{b\in\mathcal{A}} e^{\theta_{s,b}}} + e^{\theta_{s,a}} \frac{\partial}{\partial \theta_{s,a'}} \frac{1}{\sum_{b\in\mathcal{A}} e^{\theta_{s,b}}} \right) = \\
&= \mathbb{1}_{s=s'} \left(\mathbb{1}_{a=a'} \frac{e^{\theta_{s,a}}}{\sum_{b\in\mathcal{A}} e^{\theta_{s,b}}} - e^{\theta_{s,a}} \frac{e^{\theta_{s,a'}}}{(\sum_{b\in\mathcal{A}} e^{\theta_{s,b}})^2} \right) = \\
&= \mathbb{1}_{s=s'} \pi_\theta(a|s) \left(\mathbb{1}_{a=a'} - \pi_\theta(a'|s) \right).
\end{aligned}$$

This immediately translates to a vectorial expression

$$\begin{aligned}\nabla_\theta(\pi_\theta(a|s)) &= \pi_\theta(a|s)\left(\mathbf{e}_{s,a} - \sum_{a'\in\mathcal{A}} \pi_\theta(a'|s)\mathbf{e}_{s,a'}\right) \\ &= \pi_\theta(a|s)\left(\mathbf{e}_{s,a} - \mathbb{E}_{a'\sim\pi_\theta(\cdot|s)}[\mathbf{e}_{s,a'}]\right), \qquad (2.6)\end{aligned}$$

where the vector $\mathbf{e}_{s,a} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ has all zero entries apart from the one corresponding to the state-action pair (s, a), in which it has 1.

Remark. Benefits of policy based over value based methods.
In addition to the practical advantages of policy parametrization over ϵ-greedy action selection, there is also an important theoretical advantage. With continuous policy parametrization the action probabilities change smoothly as a function of the learned parameter, whereas in ϵ-greedy selection the action probabilities may change dramatically for an arbitrary small change in the estimated action values, if that change results in a different action having the maximal value.

It is also worth noting that, while in action-value methods the existence itself of the policy is dependent on the current estimation of the value function, in policy gradient methods an estimate of the value function is not even needed. Nevertheless an estimation can be made in order to improve convergence rate and, in particular, in order to reduce variance. This idea goes under the name of Actor-Critic, where the *actor* is the current policy while the *critic* is the estimated value function.

2.2.2 Policy Gradient Theorem

In the policy learning framework it may seem challenging to change the policy parameters in a way that ensures improvement. The problem is that the performance depends on both the action selection and the distribution of states in which those selection are made, and that both of these are affected by the policy parameters. Given a state, the effect of the policy parameters on the action, and thus on the reward, can be computed in a relatively straightforward way from knowledge of the parametrization. But the effect of the policy on the state distribution is a function of the environment dynamics and, in our setting, those are unknown.

Fortunately, thanks to the *policy gradient theorem* there is an analytic expression for the gradient of the value function with respect to the policy parameters that does *not* involve the derivatives of the state distribution nor the environment transition probabilities $\mathcal{P}(s'|s, a)$.

Theorem 2.3. ***Policy Gradient Theorem***
Given a MDP $\mathcal{M}$ and a discount factor γ, let $\pi_\theta \in \Pi_\mathcal{M}$ be a parametrized policy with differ-

entiable parametrization. Then it holds

$$\nabla_\theta \left(V_{\mathcal{M},\pi_\theta,\gamma}(s_0)\right) = \mathbb{E}_{s\sim d_{s_0}^{\mathcal{M},\pi_\theta,\gamma}}\left[\sum_{a\in\mathcal{A}} Q_{\mathcal{M},\pi_\theta,\gamma}(s,a)\nabla_\theta\pi_\theta(a|s)\right] \tag{2.7}$$

for every state $s_0 \in \mathcal{S}$.

Proof. To keep notation simple, since $\mathcal{M}$ and γ are fixed throughout the proof, we leave them implicit when not needed. We write V_{π_θ} in place of $V_{\mathcal{M},\pi_\theta,\gamma}$ and similarly Q_{π_θ} in place of $Q_{\mathcal{M},\pi_\theta,\gamma}$. In the same spirit we write $d_{s_0,H}^{\pi_\theta}$ in place for $d_{s_0,H}^{\mathcal{M},\pi_\theta}$ and $d_{s_0}^{\pi_\theta,\gamma}$ in place of $d_{s_0}^{\mathcal{M},\pi_\theta,\gamma}$.

$$\begin{aligned}
\nabla_\theta V_{\pi_\theta}(s) = {} & \nabla_\theta\left[\sum_{a\in\mathcal{A}}\pi_\theta(a|s)Q_{\pi_\theta}(s,a)\right] = \\
= {} & \sum_{a\in\mathcal{A}}\left[\nabla_\theta\pi_\theta(a|s)Q_{\pi_\theta}(s,a) + \pi_\theta(a|s)\nabla_\theta Q_{\pi_\theta}(s,a)\right] = \\
= {} & \sum_{a\in\mathcal{A}}\left[\nabla_\theta\pi_\theta(a|s)Q_{\pi_\theta}(s,a) + \pi_\theta(a|s)\nabla_\theta\left((1-\gamma)r(s,a) + \gamma\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)V_{\pi_\theta}(s')\right)\right] = \\
= {} & \sum_{a\in\mathcal{A}}\left[\nabla_\theta\pi_\theta(a|s)Q_{\pi_\theta}(s,a) + \gamma\pi_\theta(a|s)\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\nabla_\theta V_{\pi_\theta}(s')\right] = \\
= {} & \sum_{a\in\mathcal{A}}\left[\nabla_\theta\pi_\theta(a|s)Q_{\pi_\theta}(s,a) + \gamma\pi_\theta(a|s)\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\right. \\
& \qquad \left.\sum_{a'\in\mathcal{A}}\left[\nabla_\theta\pi_\theta(a'|s')Q_{\pi_\theta}(s',a') + \gamma\pi_\theta(a'|s')\sum_{s''\in\mathcal{S}}\mathcal{P}(s''|s',a')\nabla_\theta V_{\pi_\theta}(s'')\right]\right] = \\
= {} & \sum_{a\in\mathcal{A}}\left[\nabla_\theta\pi_\theta(a|s)Q_{\pi_\theta}(s,a) + \gamma\pi_\theta(a|s)\sum_{s'\in\mathcal{S}}\mathcal{P}(s'|s,a)\right. \\
& \qquad \sum_{a'\in\mathcal{A}}\left[\nabla_\theta\pi_\theta(a'|s')Q_{\pi_\theta}(s',a') + \gamma\pi_\theta(a'|s')\sum_{s''\in\mathcal{S}}\mathcal{P}(s''|s',a')\right. \\
& \qquad\quad \left.\left.\sum_{a''\in\mathcal{A}}\left[\nabla_\theta\pi_\theta(a''|s'')Q_{\pi_\theta}(s'',a'') + \gamma\pi_\theta(a''|s'')\sum_{s'''\in\mathcal{S}}\mathcal{P}(s'''|s'',a'')\nabla_\theta V_{\pi_\theta}(s''')\right]\right]\right].
\end{aligned}$$

Here notice that we can iterate the last step as many times as we wish, always replacing $\nabla_\theta V_{\pi_\theta}(s''')$ with the same expression we are evaluating, recursively.

With this step iterated infinitely many times we can rearrange the terms and then notice that, in each addend, the last factor $\sum_{a\in\mathcal{A}}\left[\nabla_\theta\pi_\theta(a|s)Q_{\pi_\theta}(s,a)\right]$ is exactly the same despite the summation is made over a differently indicized action space, i.e. $a' \in \mathcal{A}$, $a'' \in \mathcal{A}$ and so on.

$$
\begin{aligned}
\nabla_\theta V_{\pi_\theta}(s) = \sum_{a\in\mathcal{A}} \Bigg[& \nabla_\theta \pi_\theta(a|s) Q_{\pi_\theta}(s,a) + \gamma \pi_\theta(a|s) \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \\
& \sum_{a'\in\mathcal{A}} \Bigg[\nabla_\theta \pi_\theta(a'|s') Q_{\pi_\theta}(s',a') + \gamma \pi_\theta(a'|s') \sum_{s''\in\mathcal{S}} \mathcal{P}(s''|s',a') \\
& \qquad \sum_{a''\in\mathcal{A}} \Bigg[\nabla_\theta \pi_\theta(a''|s'') Q_{\pi_\theta}(s'',a'') + \gamma \pi_\theta(a''|s'') \sum_{s'''\in\mathcal{S}} \mathcal{P}(s'''|s'',a'') \underbrace{\nabla_\theta V_{\pi_\theta}(s''')}_{\dots} \Bigg] \Bigg] \Bigg] = \\
= & \left(\sum_{a\in\mathcal{A}} \left[\nabla_\theta \pi_\theta(a|s) Q_{\pi_\theta}(s,a) \right] \right) + \\
& \gamma \sum_{a\in\mathcal{A}} \pi_\theta(a|s) \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \left(\sum_{a'\in\mathcal{A}} \left[\nabla_\theta \pi_\theta(a'|s') Q_{\pi_\theta}(s',a') \right] \right) + \\
& \gamma^2 \sum_{a\in\mathcal{A}} \pi_\theta(a|s) \sum_{s'\in\mathcal{S}} \mathcal{P}(s'|s,a) \sum_{a'\in\mathcal{A}} \pi_\theta(a'|s') \sum_{s''\in\mathcal{S}} \mathcal{P}(s''|s',a') \left(\sum_{a''\in\mathcal{A}} \left[\nabla_\theta \pi_\theta(a''|s'') Q_{\pi_\theta}(s'',a'') \right] \right) + \\
& \dots = \\
= & \sum_{t=0}^{\infty} \gamma^t \sum_{\tilde{s}\in\mathcal{S}} d_{s,t+1}^{\pi_\theta}(\tilde{s}) \left(\sum_{a\in\mathcal{A}} \left[\nabla_\theta \pi_\theta(a|\tilde{s}) Q_{\pi_\theta}(\tilde{s},a) \right] \right) = \\
= & \sum_{\tilde{s}\in\mathcal{S}} d_s^{\pi_\theta,\gamma}(\tilde{s}) \left(\sum_{a\in\mathcal{A}} \left[\nabla_\theta \pi_\theta(a|\tilde{s}) Q_{\pi_\theta}(\tilde{s},a) \right] \right) = \\
= & \mathbb{E}_{\tilde{s}\sim d_s^{\pi_\theta,\gamma}} \left[\sum_{a\in\mathcal{A}} \nabla_\theta \pi_\theta(a|\tilde{s}) Q_{\pi_\theta}(\tilde{s},a) \right],
\end{aligned}
$$

and this completes the proof. □

As said, we need a calculable random variable $\widehat{\nabla J(\theta)}$ whose expectation approximates the gradient of the performance measure.The Policy Gradient Theorem explicits the gradient exactly as an expected value of an *almost* calculable expression. Here it is crucial to note that, despite the distribution $d_s^{\mathcal{M},\pi_\theta,\gamma}$ with respect to which the expectation is made being unknown (because it depends directly on the environment dynamics), we can still visit states in proportions given by $d_\mu^{\mathcal{M},\pi_\theta,\gamma}$ by sampling trajectories with a μ-reset model. In fact sampling trajectories we will encounter states according to the desired (unknown) distribution, by Definition 1.12 of the discounted state visitation distribution.

It is sort of straightforward now, given a trajectory τ of lenght H, to set

$$\widehat{\nabla J(\theta)} := \sum_{t=0}^{H-1} \gamma^t \sum_{a \in \mathcal{A}} Q_{\mathcal{M},\pi_\theta,\gamma}(s_t, a) \nabla_\theta \pi_\theta(a|s_t),$$

where s_t is the state visited at timestep t.

But that estimator is *almost* a calculable expression in the sense that we need to know (an estimation of) the state-action value function $Q_{\mathcal{M},\pi_\theta,\gamma}$ for every state encountered along τ and, more difficultly, for every action $a \in \mathcal{A}$, and clearly we can not encounter them in one trajectory. We are not assuming access to a generative model and so we are costrained to a single trajectory flow. This clearly means that we can not estimate also Q from τ but we need a previous computed estimator and this, as we saw, may require infinite samples to be obtained exactly.

It is actually quite easy to avoid that problem, in fact we can write the gradient as expectation of a quantity related to a single trajectory flow, at least in expected value.

Theorem 2.4. *Given a MDP $\mathcal{M}$ and a discount factor γ, let $\pi_\theta \in \Pi_\mathcal{M}$ be a parametrized policy with differentiable parametrization, such that $\pi_\theta(a|s) > 0$ for every $s \in \mathcal{S}$, $a \in \mathcal{A}$. Then it holds*

$$\nabla_\theta \left(V_{\mathcal{M},\pi_\theta,\gamma}(s_0)\right) = \mathbb{E}_{s,a \sim d_{s_0}^{\mathcal{M},\pi_\theta,\gamma}} \left[Q_{\mathcal{M},\pi_\theta,\gamma}(s, a) \nabla_\theta \log \pi_\theta(a|s)\right], \tag{2.8}$$

for every state $s_0 \in \mathcal{S}$.

Proof. From Theorem 2.3, with little algebraic manipulation we get

$$\begin{aligned}
\nabla_\theta \left(V_{\mathcal{M},\pi_\theta,\gamma}(s_0)\right) &= \mathbb{E}_{s \sim d_{s_0}^{\mathcal{M},\pi_\theta,\gamma}} \left[\sum_{a \in \mathcal{A}} Q_{\mathcal{M},\pi_\theta,\gamma}(s, a) \nabla_\theta \pi_\theta(a|s)\right] = \\
&= \mathbb{E}_{s \sim d_{s_0}^{\mathcal{M},\pi_\theta,\gamma}} \left[\sum_{a \in \mathcal{A}} \pi_\theta(a|s) Q_{\mathcal{M},\pi_\theta,\gamma}(s, a) \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}\right] = \\
&= \mathbb{E}_{s,a \sim d_{s_0}^{\mathcal{M},\pi_\theta,\gamma}} \left[Q_{\mathcal{M},\pi_\theta,\gamma}(s, a) \frac{\nabla_\theta \pi_\theta(a|s)}{\pi_\theta(a|s)}\right] = \\
&= \mathbb{E}_{s,a \sim d_{s_0}^{\mathcal{M},\pi_\theta,\gamma}} \left[Q_{\mathcal{M},\pi_\theta,\gamma}(s, a) \nabla_\theta \log \pi_\theta(a|s)\right].
\end{aligned} \tag{2.9}$$

The hypothesis of strict positivity of $\pi_\theta(a|s)$ is fundamental in order to avoid division by zero. □

We have assumed no hypothesis on the policy parametrization, other than differentiability, and so these results hold for any parametrization, not only the complete ones. Recalling the gradient expression (2.6) for the softmax parametrization it becomes clear how natural that parametrization choice is, in fact we have

$$\begin{aligned}\nabla_\theta \log \pi_\theta(a|s) &= \frac{1}{\pi_\theta(a|s)}\pi_\theta(a|s)\left(\mathbf{e}_{s,a} - \mathbb{E}_{a'\sim\pi_\theta(\cdot|s)}[\mathbf{e}_{s,a'}]\right) = \\ &= \mathbf{e}_{s,a} - \mathbb{E}_{a'\sim\pi_\theta(\cdot|s)}[\mathbf{e}_{s,a'}].\end{aligned}$$

We can go even further in the manipulation of the gradient.

Corollary 2.5. *Given a MDP $\mathcal{M}$ and a discount factor γ, let $\pi_\theta \in \Pi_{\mathcal{M}}$ be a parametrized policy with differentiable* ***uncostrained*** *parametrization. Then it holds*

$$\nabla_\theta \left(V_{\mathcal{M},\pi_\theta,\gamma}(s_0)\right) = \mathbb{E}_{s\sim d_{s_0}^{\pi,\gamma}}\left[\sum_{a\in\mathcal{A}} A_{\mathcal{M},\pi_\theta,\gamma}(s,a)\nabla_\theta\pi_\theta(a|s)\right]. \tag{2.10}$$

Moreover, if $\pi_\theta(a|s) > 0$ for every $s \in \mathcal{S}$, $a \in \mathcal{A}$, it holds

$$\nabla_\theta \left(V_{\mathcal{M},\pi_\theta,\gamma}(s_0)\right) = \mathbb{E}_{s,a\sim d_{s_0}^{\pi,\gamma}}\left[A_{\mathcal{M},\pi_\theta,\gamma}(s,a)\nabla_\theta\log\pi_\theta(a|s)\right]. \tag{2.11}$$

Proof. By definition of the Advantage function (1.17), linearity of the expected value and Theorem 2.3, we have

$$\begin{aligned}\mathbb{E}_{s\sim d_{s_0}^{\pi,\gamma}}\left[\sum_{a\in\mathcal{A}} A_{\mathcal{M},\pi_\theta,\gamma}(s,a)\nabla_\theta\pi_\theta(a|s)\right] &= \mathbb{E}_{s\sim d_{s_0}^{\pi,\gamma}}\left[\sum_{a\in\mathcal{A}} \left(Q_{\mathcal{M},\pi,\gamma}(s,a) - V_{\mathcal{M},\pi,\gamma}(s)\right)\nabla_\theta\pi_\theta(a|s)\right] = \\ &= \nabla_\theta\left(V_{\mathcal{M},\pi_\theta,\gamma}(s_0)\right) - \mathbb{E}_{s\sim d_{s_0}^{\pi,\gamma}}\left[\sum_{a\in\mathcal{A}} V_{\mathcal{M},\pi,\gamma}(s)\nabla_\theta\pi_\theta(a|s)\right].\end{aligned}$$

Now considering the residual term

$$\begin{aligned}\mathbb{E}_{s\sim d_{s_0}^{\pi,\gamma}}\left[\sum_{a\in\mathcal{A}} V_{\mathcal{M},\pi,\gamma}(s)\nabla_\theta\pi_\theta(a|s)\right] &= \mathbb{E}_{s\sim d_{s_0}^{\pi,\gamma}}\left[V_{\mathcal{M},\pi,\gamma}(s)\sum_{a\in\mathcal{A}}\nabla_\theta\pi_\theta(a|s)\right] = \\ &= \mathbb{E}_{s\sim d_{s_0}^{\pi,\gamma}}\left[V_{\mathcal{M},\pi,\gamma}(s)\nabla_\theta\left(\sum_{a\in\mathcal{A}}\pi_\theta(a|s)\right)\right] = \\ &= \mathbb{E}_{s\sim d_{s_0}^{\pi,\gamma}}\left[V_{\mathcal{M},\pi,\gamma}(s)\nabla_\theta 1\right] = \\ &= \mathbb{E}_{s\sim d_{s_0}^{\pi,\gamma}}\left[V_{\mathcal{M},\pi,\gamma}(s)\,\mathbf{0}\,\right] = \mathbf{0},\end{aligned} \tag{2.12}$$

where $\mathbf{0}$ is the vector of zeros with the same size as θ. This completes the proof of the first part of the Corollary. The second part of the Corollary follows straightforwardly with the exact same algebraic manipulation made in (2.9). □

Chapter 3

Problems and difficulties

In this Chapter we underline the challenges we have to face within the policy gradient reinforcement learning optimization problem. This is not mean to be neither an exhaustive list nor a complete formalization, it should just give intuition of what can slow down convergence and, hopefully, suggests solutions or improvements.

3.1 Non concavity

Since we are considering a gradient ascent algorithm, a basic property we may hope for is the concavity of the objective function. This would give access to a number of properties of gradient ascent, but unfortunately this is not the case.

Theorem 3.1. *There exists a MDP $\mathcal{M}$ such that the optimization problem $V_{\mathcal{M},\pi_\theta,\gamma}(s)$ is not concave in θ for both the direct and softmax parametrization.*

Proof. The proof is made by building a counterexample $\mathcal{M}$. It has 5 states, 2 action and its transition model $\mathcal{P}$ is deterministic. Whose transition graph is shown in Figure 3.1.
Note that since actions in terminal states s_3, s_4 and s_5 do not change the expected reward, we only consider actions in states s_1 and s_2. Denote the "up/above" action by a_1 and "right" action by a_2. Note that

$$V_{\mathcal{M},\pi,\gamma}(s_1) = \pi(a_2|s_1)\pi(a_1|s_2)\,\gamma r.$$

Consider two policies $\pi^{(1)}$ and $\pi^{(2)}$ parametrized respectively by

$$\theta^{(1)} = (\log 1, \log 3, \log 3, \log 1), \qquad \theta^{(2)} = (-\log 1, -\log 3, -\log 3, -\log 1),$$

where θ is written as a tuple $(\theta_{s_1,a_1}, \theta_{s_1,a_2}, \theta_{s_2,a_1}, \theta_{s_2,a_2})$.

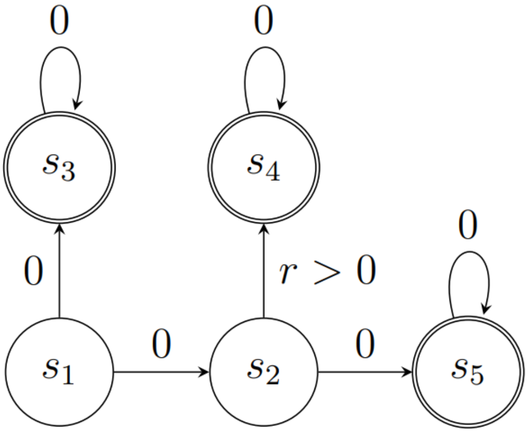


Figure 3.1: Transition graph of a MDP with non-concave value function

Then, for the softmax parametrization, we have

$$\pi^{(1)}(a_2|s_1) = \frac{3}{4} \qquad \pi^{(1)}(a_1|s_2) = \frac{3}{4} \qquad \Longrightarrow \qquad V_{\mathcal{M},\pi^{(1)},\gamma}(s_1) = \frac{9}{16}\gamma r$$

and

$$\pi^{(2)}(a_2|s_1) = \frac{1}{4} \qquad \pi^{(2)}(a_1|s_2) = \frac{1}{4} \qquad \Longrightarrow \qquad V_{\mathcal{M},\pi^{(2)},\gamma}(s_1) = \frac{1}{16}\gamma r.$$

Also, for $\theta^{(\text{mid})} := \frac{\theta^{(1)}+\theta^{(2)}}{2}$,

$$\pi^{(\text{mid})}(a_2|s_1) = \frac{1}{2} \qquad \pi^{(\text{mid})}(a_1|s_2) = \frac{1}{2} \qquad \Longrightarrow \qquad V_{\mathcal{M},\pi^{(\text{mid})},\gamma}(s_1) = \frac{1}{4}\gamma r.$$

This gives

$$V_{\mathcal{M},\pi^{(1)},\gamma}(s_1) + V_{\mathcal{M},\pi^{(2)},\gamma}(s_1) = \frac{9}{16}\gamma r + \frac{1}{16}\gamma r > \frac{2}{4}\gamma r = 2V_{\mathcal{M},\pi^{(\text{mid})},\gamma}(s_1)$$

which shows that $V_{\mathcal{M},\pi_\theta,\gamma}(s_1)$ is non-concave in θ. □

3.1.1 Vanishing gradient at deterministic policies

Related to the non concavity problem we note that every deterministic policy has zero gradient.

Lemma 3.2. Given a MDP $\mathcal{M}$ and a discount factor γ, let $\pi_\theta \in \Pi^D_{\mathcal{M}}$ be a deterministic policy parametrized with softmax parametrization. Then for every $s_0 \in \mathcal{S}$ it holds

$$\nabla_\theta \left(V_{\mathcal{M},\pi_\theta,\gamma}(s_0)\right) = \mathbf{0},$$

where the vector $\mathbf{0} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}|}$ has all zero entries.

Proof. Recall the gradient expression (4.5) of the policy parametrized with softmax parametrization:

$$\nabla_\theta(\pi_\theta(a|s)) = \pi_\theta(a|s)\left(\mathbf{e}_{s,a} - \mathbb{E}_{a'\sim\pi_\theta(\cdot|s)}[\mathbf{e}_{s,a'}]\right) = \begin{cases} 0\left(\mathbf{e}_{s,a} - \mathbb{E}_{a'\sim\pi_\theta(\cdot|s)}[\mathbf{e}_{s,a'}]\right) & \text{if } \pi_\theta(a|s) = 0 \\ 1\left(\mathbf{e}_{s,a} - \mathbf{e}_{s,a}\right) & \text{if } \pi_\theta(a|s) = 1 \end{cases}$$

this is the vector $\mathbf{0}$ for every $s \in \mathcal{S}$, $a \in \mathcal{A}$. Now, by Theorem 2.3:

$$\nabla_\theta\left(V_{\mathcal{M},\pi_\theta,\gamma}(s_0)\right) = \mathbb{E}_{s\sim d_{s_0}^{\mathcal{M},\pi_\theta,\gamma}}\left[\sum_{a\in\mathcal{A}} Q_{\mathcal{M},\pi_\theta,\gamma}(s,a)\nabla_\theta\pi_\theta(a|s)\right] = \mathbb{E}_{s\sim d_{s_0}^{\mathcal{M},\pi_\theta,\gamma}}[\mathbf{0}] = \mathbf{0}.$$

□

Since every deterministic policy is a stationary point, we want to avoid them during our learning. Despite this problem, the following theorem states that gradient ascent asymptotically converges to the global optimum for the softmax parametrization.

Theorem 3.3. *(Global convergence for* ***exact*** *gradient ascent without regularization)*
Let be given an MDP $\mathcal{M}$, a discount factor $\gamma \in (0,1)$ and a μ-reset model with $\mu(s) > 0$ for all states $s \in \mathcal{S}$. Assume to use the softmax parametrization and set initial weights $\theta^{(0)}$ such that $\pi_{\theta^{(0)}}(a|s) > 0$ for every $s \in \mathcal{S}$, $a \in \mathcal{A}$. Assume to follow the gradient ascent update rule

$$\theta^{(k+1)} := \theta^{(k)} + \eta\nabla_\theta V_{\mathcal{M},\pi^{(k)},\gamma}(\mu),$$

where the learning rate $\eta \le \frac{(1-\gamma)^3}{8}$. Then

$$\lim_{k\to\infty} V_{\mathcal{M},\pi^{(k)},\gamma}(s) = V^*_{\mathcal{M},\gamma}(s)$$

for any state $s \in \mathcal{S}$.

An elegant proof of this Theorem can be found in [2].

Due to the exponential scaling with the parameters θ, policies can rapidly become near deterministic, when optimizing under the softmax parametrization, which can result in slow convergence. Indeed a key challenge in the asymptotic analysis in the proof of the previous theorem was to handle the growth of the absolute values of the parameters as they go to infinity.

A common practical remedy is to consider a new objective function that is the value function, as before, plus a **regularization term** $R(\theta)$ to keep probabilities from getting too small.

$$L_\lambda(\theta) := V_{\mathcal{M},\pi_\theta,\gamma}(\mu) - \lambda R(\theta)$$

where λ is a regularization parameter. The most commonly used family of regularization are the *entropy-based* ones, proposed and discussed by [8] and then used, for example, by [5] with asynchronous methods.

Remark. Recall that the *Shannon entropy* for a distribution p is defined as

$$\mathcal{H}(p) := \mathbb{E}_{x \sim p}\left[-\log p(x)\right].$$

Recall that the *relative entropy*, also called Kullback–Leibler divergence, for distribution p and q is defined as

$$\mathrm{KL}(p, q) := \mathbb{E}_{x \sim p}\left[-\log \frac{q(x)}{p(x)}\right].$$

Also, denote the uniform distribution over a set $\mathcal{X}$ by $\mathrm{Unif}_{\mathcal{X}}$.

Definition 3.4. Given a MDP $\mathcal{M}$, the **entropy regularizer** $R^E : \Theta \to \mathbb{R}$ is the mean over states of the Shannon entropy of the policy π_θ, that is:

$$\begin{aligned} R^E(\theta) &:= \mathbb{E}_{s \sim \mathrm{Unif}_{\mathcal{S}}}\left[\mathcal{H}(\pi_\theta(\cdot|s))\right] \\ &= \frac{1}{|\mathcal{S}|}\sum_{s \in \mathcal{S}}\sum_{a \in \mathcal{A}} -\pi_\theta(a|s)\log \pi_\theta(a|s). \end{aligned}$$

Definition 3.5. Given a MDP $\mathcal{M}$, the **log-barrier regularizer** $R : \Theta \to \mathbb{R}$ is the mean over states of the Kullback–Leibler divergence of π_θ from the uniform distribution over actions:

$$\begin{aligned} R(\theta) &:= \mathbb{E}_{s \sim \mathrm{Unif}_{\mathcal{S}}}\left[\mathrm{KL}(\mathrm{Unif}_{\mathcal{A}}, \pi_\theta(\cdot|s)\right] = \\ &= \frac{1}{|\mathcal{S}||\mathcal{A}|}\sum_{s \in \mathcal{S}}\sum_{a \in \mathcal{A}} \log \pi_\theta(a|s) + \log|\mathcal{A}| \end{aligned} \tag{3.1}$$

Notice that in (3.1) the constant $\log|\mathcal{A}|$ is not relevant with regards to optimization since we will use this surrogate objective function only through its gradient

$$\nabla_\theta R(\theta) = \frac{1}{|\mathcal{S}||\mathcal{A}|}\sum_{s \in \mathcal{S}}\sum_{a \in \mathcal{A}} \nabla_\theta \log \pi_\theta(a|s).$$

Furthermore, notice that the entropy is far less aggressive in penalizing small probabilities, in comparison to the log-barrier. In particular, the entropy regularizer is always bounded between 0 and $\log|\mathcal{A}|$, while the log-barrier is bounded between 0 and infinity, where it goes to infinity as probabilities go to 0.

Theorem 4.5, Theorem 5.1 and Theorem 5.3 crucially rely on this aggressive behaviour of the log-barrier entropy in preventing small probabilities. For this reason, in the following we consider only log-barrier (3.1) as regularization term.

3.1.2 Vanishing gradient at suboptimal parameter

We have seen that the gradient is zero for deterministic policies. Another big non-concavity problem we have to face with is that the gradient can still be very close to 0 at suboptimal parameter, as we see in Example 3.6.

Example 3.6. Consecutive crossroads traps Fix some number $H \in \mathbb{N}$ and consider the MDP $\mathcal{M}$ the transition graph of which is shown in Figure 3.2.

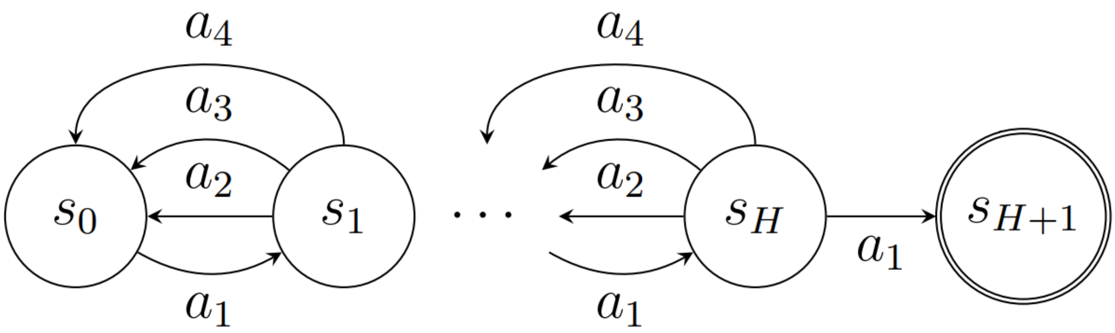


Figure 3.2: Transition graph of the consecutive crossroads traps MDP.

- the state space $\mathcal{S} = \{s_0, s_1, ..., s_{H+1}\}$ has size $H + 2$;
- the action space $\mathcal{A} = \{a_0, a_1, a_2, a_3\}$ has size 4;
- the only positive reward is when arriving at the terminal state s_{H+1}, that is $r(s_H, a_1) = 1$ and $r(s, a) = 0$ for any $(s, a) \neq (s_H, a_1)$;
- the transition model is deterministic, *i.e.* $\mathcal{P}(s'|s, a) \in \{0, 1\}$. For each pair of consecutive states there are 3 actions that go backwards, moving away from the reward, and 1 action that goes forward, approaching the reward.

Assume to have access to a μ-reset model with μ supported only on s_0, that is $\mu(s_0) = 1$ and $\mu(s) = 0$ for any $s \neq s_0$.

Proposition 3.7. Consider the chain MDP $\mathcal{M}$ of Example 3.6 shown in Figure 3.2. Assume a discount factor $\gamma = \frac{H}{H+1}$, and a policy $\pi_\theta \in \Pi_{\mathcal{M}}$ such that $\pi_\theta(a_1|s) \leq \frac{1}{4}$ for every state $s \in \mathcal{S}$. Then, for the direct parametrization, we have

$$\left\|\nabla_\theta V_{\mathcal{M},\pi_\theta,\gamma}(s_0)\right\|_2 \leq \left(\frac{1}{3}\right)^{\frac{H}{4}}.$$

For the complete proof we refer to [2], which proves a similar exponentially scaling bound also for the operator norm of the tensor of the k^{th} order derivatives of $V_{\mathcal{M},\pi_\theta,\gamma}(s_0)$. Despite the technical details, the intuition behind the exponential scaling is clear: in order to recieve a non-zero return the agent has to choose the only good action among four possibilities, and he has to do it *independently* H times in a row. Thus, for $H >> 0$ the gradient will be almost 0, and learning, if any, will be very slow. Section 6.2 illustrate this effect numerically.

This MDP is a common example where *sample* based estimates of the gradient will be close to 0 under random exploration strategies; there is an exponentially small in H chance of hitting the goal state under a random exploration strategy. This suggests that even with exact computations we might expect numerical instabilities. This issue can actually be reframed as an exploration problem.

3.2 Exact vs approximate gradient

In real usage cases we will never have access to the exact gradient $\nabla_\theta L_\lambda$, because it depends on the model transition probabilities $\mathcal{P}(s'|s,a)$ and in our totally unknown environment setting those are unknown.

Instead what we can do is to use episodes. That is, sample M trajectories $\{\tau_m\}_{m=1,\dots,M}$ following our policy π. These trajectories will be distributed accordingly to the unknown model, moreover they will be independent and identically distributed random variables. With this simple observation it is clear that with M episodes, called the *minibatch size*, we can obtain a gradient approximation that tends to the exact gradient for $M \to \infty$. But, as Example 3.6 shows explicitely, a non zero gradient estimation may require exponentially many episodes, since the reward is *sparse* and we almost never obtain it.

A gradient estimator $\hat{\nabla}_\theta$ is a function that, similarly to the gradient ∇_θ, maps scalar-valued functions $L : \Theta \mapsto \mathbb{R}$ to vector-valued function $\hat{\nabla}_\theta L : \Theta \mapsto \mathbb{R}^T$, where T is the vectorial dimension of Θ, in our case $T = |\mathcal{S}||\mathcal{A}|$. We will restrict ourselves to gradient estimators that only use a fixed number M of episodes $\{\tau_m\}_{m=1,\dots,M}$, in particular we focus on the case $M = 1$. We expect (and want) these gradient estimator to be not so bad in the worst case and good on average, this informal statement is quantified by the following assumption.

Assumption 3.8. Gradient Approximation
Given a trajectory space $\mathcal{T}$, a minibatch size $M > 0$, a vector space Θ of dimension T and a differentiable function $L : \Theta \mapsto \mathbb{R}$, let $\nabla_\theta L : \Theta \mapsto \mathbb{R}^T$ be its gradient. A gradient approximator $\hat{\nabla}_\theta : \mathbb{R}^\Theta \times \mathcal{T}^M \mapsto (\mathbb{R}^T)^\Theta$ is said to *satisfy Approximation Assumption* if and only if there exist constants $C_1, C_2, M_1, M_2 \in \mathbb{R}$ such that it holds:

- gradient estimation boundedness

$$\left\|\hat{\nabla}_\theta(L;\{\tau_m\})(\theta)\right\|_2 \leq C_1 \text{ almost surely;} \tag{3.2}$$

- nearly unbiasedness

$$\langle \nabla_\theta L(\theta), \mathbb{E}_{\mathcal{T}}[\hat{\nabla}_\theta(L; \{\tau_m\})(\theta)]\rangle \geq C_2 \left\|\nabla_\theta L(\theta)\right\|_2^2 - \delta_k \text{ with } \delta_k \in \ell^2; \tag{3.3}$$

- bounded second-order moment growth

$$\mathbb{E}_{\mathcal{T}}\left[\left\|\hat{\nabla}_\theta(L; \{\tau_m\})(\theta)\right\|_2^2\right] \leq M_1 + M_2 \left\|\nabla_\theta L(\theta)\right\|_2^2. \tag{3.4}$$

The stochasticity of the approximator is inherited from the stochasticity in the trajectory space $\mathcal{T}$, and thus the expectation with respect to it. We omit the dependence from trajectories when clear from context. Here and in the following $\|\cdot\|_2$ is a short for $\|\cdot\|_{\ell^2(\mathbb{R}^T)}$.

Notice that this Assumption immediately holds if $\hat{\nabla}_\theta(L, \{\tau_m\})$ is unbiased and has a bounded second-order moment instead of just a bounded second-order moment growth.

3.3 Exploration

Example 3.9. Imagine you have two doors in front of you. You open the first door and receive a reward of 0. You open the second and receive a reward of +3, you open the second again and receive +1, you open it again and receive +2. Which is the best door to open?

This concept can be reframed also as: when selecting a restaurant, do you choose your favorite one or do you try something new? Or, similarly: during game playing, do you play the move you believe is best or do you play an experimental move? Basically you always have to choose between (a), making the best decision given current information, and (b), gather more information.

The **exploration-exploitation dilemma** has no formal definition but it may be divided in two aspects:

- the need of visiting every state at least once, *i.e.* the *exploration*;
- the need of continuing to visit those state frequently, *i.e.* the *exploitation.*

The former is, at our best, isolated and underlined by Example 3.6: there is an exponential scaling of the state visitation distribution for the uniform policy. The latter is underlined by the following example.

Example 3.10. Multipath Consider the MDP $\mathcal{M}$ the transition graph of which is shown in Figure 3.3. As for Example 3.6, here the environment dynamics are deterministic and thus action nodes are not shown in the transition graph. In this MDP there are 3 main paths,

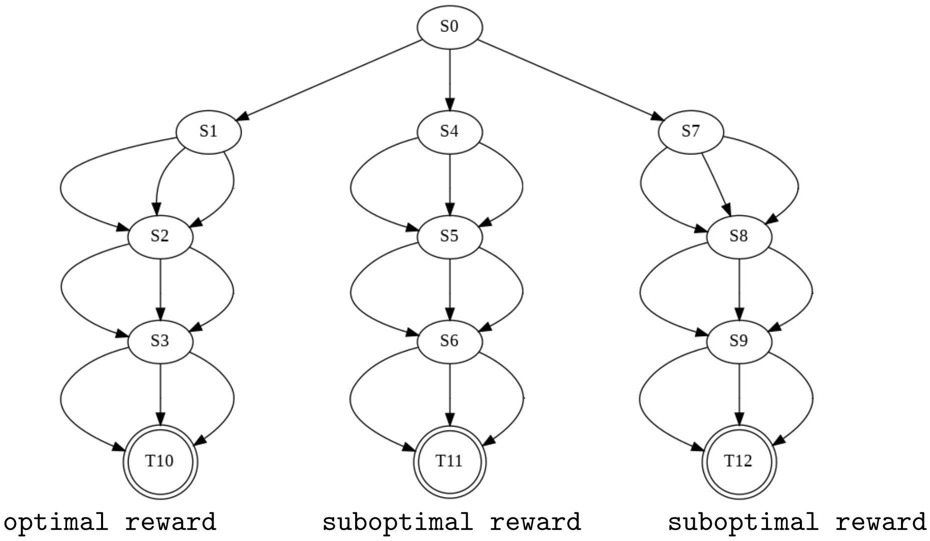


Figure 3.3: Transition graph of the multipath MDP.

the choice of which to follow is made on the first state s_0, and each of these paths leads to a reward at the end. The difference is that one path has a slightly greater total return, namely this is the *optimal path*, while other paths still have positive but smaller return, namely those are *suboptimal paths*. The challenge is to continue visiting all the paths until they are all well evaluated.

Furthermore, a suboptimal path may have positive rewards in the first steps while optimal path may have zero rewards for an arbitrarly long initial number of steps. In other words: the best long-term strategy may involve short-term sacrifices.

In the totally unknown environment setting, at the beginning or our learning we have no information about which path is the optimal one. We wish to evaluate the *cost* of that missing information or, which is the same, the *cost* of gathering that information. Since we are interested in sample efficiency, that cost is measured in terms of number of samples.

In this specific Example 3.10 that cost is clearly a factor 3 on the number of required samples, meaning that - not knowing which the optimal path is - we need to perform the same strategy on each of the 3 possible paths. This factor is easily seen only in toy examples like this. Despite of this, we can give a formal definition that is strictly related to this factor.

Definition 3.11. Exploitation coefficient

Given a MDP $\mathcal{M}$ and a discount factor γ we refer to

$$\sum_{s \in \mathcal{S}} \max_{\pi \in \Pi_{\mathcal{M}}} d_{\mu}^{\mathcal{M},\pi,\gamma}(s)$$

as the *exploitation coefficient*.

This is indeed a characteristic property of a Markow Decision Process and it strongly depends on its graph structure. Intuitively, the bigger this coefficient is, the harder the reinforcement learning task become. In fact, having several different states with a high state visitation number under some policy means having several “paths” in need of being exploited independetly one from each other. We quantified this as an upper bound in (5.30).

The *exploration-exploitation dilemma* is (in some sense) avoided by a number of reinforcement learning optimization algorithms by assuming a strong hypothesis on the μ-reset sampling model

Definition 3.12. Coverage hypothesis

Let $\mathcal{M}$ be a MDP. A μ-reset sampling model for $\mathcal{M}$ satisfy the *coverage hypothesis* if

$$\mu(s) > 0 \quad \text{for every } s \in \mathcal{S}. \tag{3.5}$$

This hypothesis avoids the problems in the sense that the requirements of exploration and exploitation *along states visited during trajectories* are satisfied by the initial state of the trajectory alone.

With $N >> 1$ calls to the sampling model each state is visited at least $N\mu(s)$ times. This suggests that, within the coverage hypothesis, we must further distinguish: a μ-reset with μ close to zero in some state is clearly less powerful than a μ-reset with μ close to the uniform distribution.

In literature it is very common to quantify this power of the μ-reset sampling model when maximizing the objective $V_{\mathcal{M},\pi,\gamma}(\rho)$ through the following notion of *distribution mismatch coefficient.*

Definition 3.13. Distribution mismatch coefficient

Let $\mathcal{M}$ be a MDP. Given a policy $\pi \in \Pi_{\mathcal{M}}$ and two distributions ρ, μ over the state space $\mathcal{S}$, we refer to

$$\left\| \frac{d_\rho^{\mathcal{M},\pi,\gamma}}{\mu} \right\|_\infty = \max_{s\in\mathcal{S}} \frac{d_\rho^{\mathcal{M},\pi,\gamma}(s)}{\mu(s)} \tag{3.6}$$

as the *distribution mismatch coefficient* of π relative to μ.

The policy π is often chosen to be an optimal policy π^*. The distribution mismatch coefficient, as we will see in the next Chapter, comes out very naturally in bounding the suboptimality of a policy. This becames the core of the exploration problem when we drop the coverage hypothesis on μ and assume access only to a weak s_0-sampling model. In this latter case, in fact, the distribution mismatch coefficient becomes infinite and the convergence results invoving it do not hold anymore.

It will come in handy in (5.25) in Chapter 5 and in experiments in Chapter 6 to define a policy that is the best at exploring among all the policies, where the ordering is made on the minumum of the state visitation distribution.

Definition 3.14. Given a MDP $\mathcal{M}$ and a discount factor γ, we refer to the **optimal explorative policy** as the policy $\pi^E \in \Pi_{\mathcal{M}}$ such that

$$\pi^E :\in \arg\max_{\Pi_{\mathcal{M}}} \left(\min_{s \in \mathcal{S}} d_\mu^{\mathcal{M},\pi,\gamma}(s) \right).$$

Moreover we set the constant $\tilde{d} \in \mathbb{R}$ as

$$\tilde{d} := \min_{s \in \mathcal{S}} d_\mu^{\mathcal{M},\pi^E,\gamma}(s)$$

The following toy problem shows that the two issues introduced by Example 3.6 and Example 3.10 can in fact appear together. This is commonly used in literature as a very difficult example to test methods [1].

Example 3.15. Bidirectional Diabolical Combination Lock
Consider the MDP shown in Figure 3.4. It consists of two (or more) paths, the access at which is determined entirely on the first action, as in the Multipath Example. In each path there are a series of actions (traps) which bring the agent back towards the start, only one action makes the agent go further, towards the end of the path, similarly to the Consecutive Crossroad Traps Example. The reward is always zero but for the final states of each path, where it is positive. Furthermore, one path has a greater reward and is thus optimal, the other path still have rewards but smaller and thus are sub-optimal.

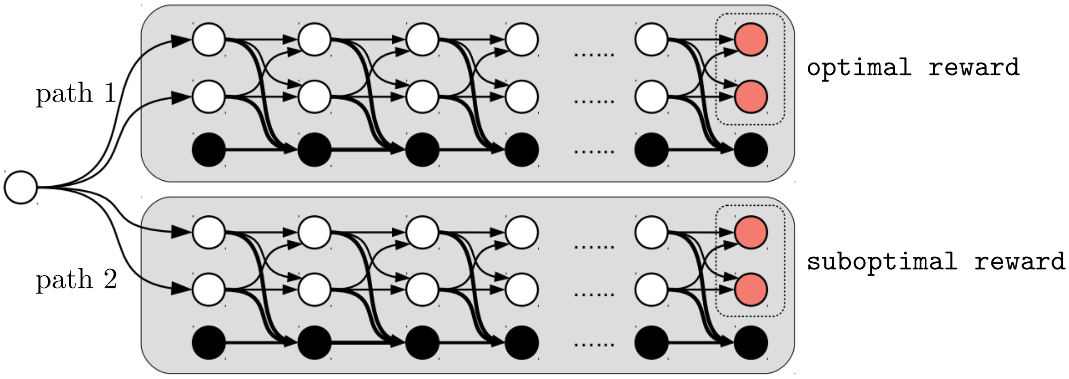


Figure 3.4: Transition graph of the bidirectional diabolical combination lock MDP.

Chapter 4

Results

We use, study and test a Policy Gradient methods with a log-barrier regularization term. Our surrogate objective function is then

$$L_\lambda(\theta) := V_{\mathcal{M},\pi_\theta,\gamma}(\mu) + \lambda R(\theta) \tag{4.1}$$

where $R(\theta) = \frac{1}{|\mathcal{S}||\mathcal{A}|}\sum_{s\in\mathcal{S},a\in\mathcal{A}} \log \pi_\theta(a|s)$ is the regularization term. For the second addend we can compute the exact gradient while for the first one we will need an approximator. For each episode we will estimate the gradient by

$$\hat{\nabla}_\theta L_\lambda(\theta) = \hat{\nabla}_\theta V_{\mathcal{M},\pi_\theta,\gamma}(\mu) + \lambda \nabla R(\theta)$$

and with this we will perform the gradient ascent update

$$\theta^{k+1} = \theta^k + \alpha^k \hat{\nabla}_\theta L_\lambda(\theta^k),$$

for a suitable learning rate sequence $\{\alpha^k\}$. How to build the approximation $\hat{\nabla}_\theta$ will be explained in Section 4.3. In particular, we will explain some of the most common single trajectory gradient approximator for the value function.

4.1 Performance Lemma

In this Section we show a result that quantifies the distance from the optimal value function and, consequently, from an optimal policy. But first it is useful to prove a link between the value function of a policy and its discounted state visitation distribution.
For clarity we recall here the definition of value function (1.11):

$$V_{\mathcal{M},\pi,\gamma}(s) = (1-\gamma)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right],$$

and the definition of discounted state-action visitation distribution (1.6)

$$\begin{aligned} d_{s_0}^{\mathcal{M},\pi,\gamma}(s,a) &= (1-\gamma)\sum_{H=0}^{\infty}\gamma^H d_{s_0,H+1}^{\mathcal{M},\pi}(s,a) \\ &= (1-\gamma)\sum_{H=0}^{\infty}\gamma^H \sum_{\tau\in T_{H+1}} \mathbb{1}_{\{s_H=s\}}\mathbb{1}_{\{a_H=a\}}\mathbb{P}(\tau|\pi,\mathcal{M},s_0). \end{aligned}$$

Lemma 4.1. Given an MDP $\mathcal{M}$, a policy $\pi \in \Pi_{\mathcal{M}}$ and a discount factor $\gamma \in [0,1)$ it holds that

$$V_{\mathcal{M},\pi,\gamma}(s_0) = \sum_{s\in\mathcal{S},a\in\mathcal{A}} d_{s_0}^{\mathcal{M},\pi,\gamma}(s,a)r(s,a), \tag{4.2}$$

where r is the reward function of $\mathcal{M}$.

Proof. Recall that the function $\sum_{s\in\mathcal{S},a\in\mathcal{A}}\mathbb{1}_{\{s_t=s\}}\mathbb{1}_{\{a_t=a\}}$ is constant and equal to 1.

$$\begin{aligned} V_{\mathcal{M},\pi,\gamma}(s_0) &= (1-\gamma)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right] = \\ &= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mathbb{E}_{\tau\sim\mathbb{P}_{t+1}(\cdot|\pi,\mathcal{M},s_0)}\left[r(s_t,a_t)\right] = \\ &= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mathbb{E}_{\tau\sim\mathbb{P}_{t+1}(\cdot|\pi,\mathcal{M},s_0)}\left[\sum_{s\in\mathcal{S},a\in\mathcal{A}}\mathbb{1}_{\{s_t=s\}}\mathbb{1}_{\{a_t=a\}}r(s_t,a_t)\right] = \\ &= \sum_{s\in\mathcal{S},a\in\mathcal{A}}(1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mathbb{E}_{\tau\sim\mathbb{P}_{t+1}(\cdot|\pi,\mathcal{M},s_0)}\left[\mathbb{1}_{\{s_t=s\}}\mathbb{1}_{\{a_t=a\}}r(s_t,a_t)\right] = \\ &= \sum_{s\in\mathcal{S},a\in\mathcal{A}}(1-\gamma)\sum_{t=0}^{\infty}\gamma^t\sum_{\tau\in T_{t+1}}\mathbb{P}_{t+1}(\tau|\pi,\mathcal{M},s_0)\mathbb{1}_{\{s_t=s\}}\mathbb{1}_{\{a_t=a\}}r(s_t,a_t) = \\ &= \sum_{s\in\mathcal{S},a\in\mathcal{A}}r(s,a)(1-\gamma)\sum_{t=0}^{\infty}\gamma^t\sum_{\tau\in T_{t+1}}\mathbb{P}_{t+1}(\tau|\pi,\mathcal{M},s_0)\mathbb{1}_{\{s_t=s\}}\mathbb{1}_{\{a_t=a\}} = \\ &= \sum_{s\in\mathcal{S},a\in\mathcal{A}}r(s,a)d_{s_0}^{\mathcal{M},\pi,\gamma}(s,a). \end{aligned}$$

The first equality is the definition of value function (1.11) while the latter is the definition of discounted state-action visitation distribution (1.6). The only non trivial passage is the use of the linearity of the expected value and the properties of the indicator functions to exchange the summation over timestep and the summation over state-action pairs. □

This result suggests to follow a distributional approach, that is, a visualization not through trajectories but through probabilities of visitation.

We are now ready to state a useful Lemma that quantifies the difference between the value function of two arbitrary policies.

Lemma 4.2. Given a MDP $\mathcal{M}$ and a discount factor γ, let $\pi, \pi' \in \Pi_{\mathcal{M}}$ two policies. Then it holds

$$V_{\mathcal{M},\pi,\gamma}(s_0) - V_{\mathcal{M},\pi',\gamma}(s_0) = \frac{1}{1-\gamma}\mathbb{E}_{s,a\sim d_{s_0}^{\mathcal{M},\pi,\gamma}}[A_{\mathcal{M},\pi',\gamma}(s,a)].$$

Proof. We begin by using the definition of the value function only on the first term and then adding and subtracting the same quantity inside the expected value.

$$\begin{aligned}
V_{\mathcal{M},\pi,\gamma}(s_0) - V_{\mathcal{M},\pi',\gamma}(s_0) &= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\sum_{t=0}^{\infty}\gamma^t(1-\gamma)r(s_t,a_t)\right] - V_{\mathcal{M},\pi',\gamma}(s_0) = \\
&= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\sum_{t=0}^{\infty}\gamma^t\left((1-\gamma)r(s_t,a_t) + V_{\mathcal{M},\pi',\gamma}(s_t) - V_{\mathcal{M},\pi',\gamma}(s_t)\right)\right] - V_{\mathcal{M},\pi',\gamma}(s_0) = \\
&= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\sum_{t=0}^{\infty}\gamma^t\left((1-\gamma)r(s_t,a_t) + \gamma V_{\mathcal{M},\pi',\gamma}(s_{t+1}) - V_{\mathcal{M},\pi',\gamma}(s_t)\right)\right] = \\
&= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\sum_{t=0}^{\infty}\gamma^t\left((1-\gamma)r(s_t,a_t) + \gamma\mathbb{E}_{s_{t+1}\sim\mathcal{P}(\cdot|s_t,a_t)}[V_{\mathcal{M},\pi',\gamma}(s_{t+1})] - V_{\mathcal{M},\pi',\gamma}(s_t)\right)\right] = \\
&= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\sum_{t=0}^{\infty}\gamma^t\left(Q_{\mathcal{M},\pi',\gamma}(s_t,a_t) - V_{\mathcal{M},\pi',\gamma}(s_t)\right)\right] = \\
&= \mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\sum_{t=0}^{\infty}\gamma^t A_{\mathcal{M},\pi',\gamma}(s_t,a_t)\right] = \\
&= \sum_{s\in\mathcal{A},a\in\mathcal{A}} A_{\mathcal{M},\pi',\gamma}(s,a)\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0=s)}\left[\sum_{t=0}^{\infty}\gamma^t\mathbb{1}_{\{s_t=s\}}\mathbb{1}_{\{a_t=a\}}\right] = \\
&= \frac{1}{1-\gamma}\sum_{s\in\mathcal{A},a\in\mathcal{A}} A_{\mathcal{M},\pi',\gamma}(s,a)d_{s_0}^{\mathcal{M},\pi,\gamma}(s,a),
\end{aligned}$$

where the third equality is a rearranging of the terms and the fourth comes from the tower property of the expected value. The conclusion is then easily reached via definitions and properties of the indicator functions. □

Now it is straightforward to evaluate the previous equation with an optimal policy π^*.

Corollary 4.3. ***Performance Lemma***
Given an MDP $\mathcal{M}$, a policy $\pi \in \Pi_{\mathcal{M}}$ and a discount factor $\gamma \in [0,1)$ it holds that

$$V_{\mathcal{M},\pi^*,\gamma}(s_0) - V_{\mathcal{M},\pi,\gamma}(s_0) = \frac{1}{1-\gamma}\mathbb{E}_{s,a\sim d_{s_0}^{\mathcal{M},\pi^*,\gamma}}[A_{\mathcal{M},\pi,\gamma}(s,a)]. \tag{4.3}$$

4.2 Gradient domination

A standard way of dealing with the non concavity problem is via gradient domination lemmas. In particular, these lemmas bound the distance from the optimal value providing that the gradient is sufficiently small.

Theorem 4.4. *Given a MDP $\mathcal{M}$ and a discount factor γ, assume to use direct parametrization. For all distributions ρ, μ over state space $\mathcal{S}$, we have*

$$V^*(\rho) - V^{\pi_\theta}(\rho) \leq \left\| \frac{d_\rho^{\mathcal{M},\pi^*,\gamma}}{d_\mu^{\mathcal{M},\pi_\theta,\gamma}} \right\|_\infty \max_{\hat{\pi}} \langle \hat{\pi} - \pi_\theta, \, \nabla_\theta V^{\pi_\theta}(\mu) \rangle.$$

Proof. The complete proof can be found in [2]. □

For the softmax parametrization we can hope for a similar result but this is not feasible. In fact with this parametrization every near deterministic policy leads to arbitrarly small gradient, and that is also why in this setting the convergence rate can be very small. We can avoid this problem by adding a regularization term, which keeps the policies from being to deterministic. In this setting we have:

Theorem 4.5. *Given a MDP $\mathcal{M}$, a discount factor γ and a distribution μ over the state space. Assume $L_\lambda(\theta) = V_{\mathcal{M},\pi_\theta,\gamma}(\mu) + \lambda R(\theta)$ as defined in (4.1) with softmax parametrization. Suppose the policy parameter θ is such that:*

$$\|\nabla_\theta L_\lambda(\theta)\|_2 \leq \frac{\lambda}{2|\mathcal{S}||\mathcal{A}|}.$$

Then we have that for all distributions ρ on the state space it holds

$$V_{\mathcal{M},\pi^*,\gamma}(\rho) - V_{\mathcal{M},\pi,\gamma}(\rho) \leq 2\lambda \left\| \frac{d_\rho^{\mathcal{M},\pi^*,\gamma}}{d_\mu^{\mathcal{M},\pi_\theta,\gamma}} \right\|_\infty. \tag{4.4}$$

Proof. First we quantify the power of the regularization term, in fact it is guaranteed that every action with positive advantage has probability of being selected of at least $\frac{1}{2|\mathcal{A}|}$. To see this, consider a state action pair (s, a) such that $A_{\mathcal{M},\pi_\theta,\gamma}(s, a) > 0$. Using the value function gradient expression (2.11) for the softmax parametrization we get

$$\frac{\partial L_\lambda(\theta)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d_\mu^{\mathcal{M},\pi_\theta,\gamma}(s, a) A^{\mathcal{M},\pi_\theta,\gamma}(s, a) + \frac{\lambda}{|\mathcal{S}|} \left(\frac{1}{|\mathcal{A}|} - \pi_\theta(a|s) \right). \tag{4.5}$$

The gradient norm assumption $\|\nabla_\theta L_\lambda(\theta)\|_2 \leq \frac{\lambda}{2|\mathcal{S}||\mathcal{A}|}$ implies that:

$$\begin{aligned}\frac{\lambda}{2|\mathcal{S}||\mathcal{A}|} &\geq \frac{\partial L_\lambda(\theta)}{\partial \theta_{s,a}} \\ &= \frac{1}{1-\gamma} d_\mu^{\mathcal{M},\pi_\theta,\gamma}(s,a) A^{\mathcal{M},\pi_\theta,\gamma}(s,a) + \frac{\lambda}{|\mathcal{S}|}\left(\frac{1}{|\mathcal{A}|} - \pi_\theta(a|s)\right) \\ &\geq \frac{\lambda}{|\mathcal{S}|}\left(\frac{1}{|\mathcal{A}|} - \pi_\theta(a|s)\right),\end{aligned}$$

where we have used $A_{\mathcal{M},\pi_\theta,\gamma}(s,a) > 0$. Rearranging the terms,

$$\pi_\theta(a|s) \geq \frac{1}{|\mathcal{A}|} - \frac{|\mathcal{S}|}{\lambda}\frac{\lambda}{2|\mathcal{S}||\mathcal{A}|} = \frac{1}{2|\mathcal{A}|}. \tag{4.6}$$

With this we can now proceed to show that

$$\max_{a \in \mathcal{A}} A^{\mathcal{M},\pi_\theta,\gamma}(s,a) \leq \frac{2\lambda(1-\gamma)}{d_\mu^{\mathcal{M},\pi_\theta,\gamma}(s)|\mathcal{S}|}, \tag{4.7}$$

for every state $s \in \mathcal{S}$. For this, it suffices to bound $A^{\mathcal{M},\pi_\theta,\gamma}(s,a)$ for any state-action pair where $A^{\mathcal{M},\pi_\theta,\gamma}(s,a) \geq 0$, because elsewhere the claim is trivially true. We can then use (4.6) in the proof.

Rearranging the terms in the gradient expression (4.5) we get an expression for the advantage

$$A^{\mathcal{M},\pi_\theta,\gamma}(s,a) = \frac{1-\gamma}{d_\mu^{\mathcal{M},\pi_\theta,\gamma}(s,a)}\left(\frac{\partial L_\lambda(\theta)}{\partial \theta_{s,a}} - \frac{\lambda}{|\mathcal{S}|}\left(\frac{1}{|\mathcal{A}|} - \pi_\theta(a|s)\right)\right),$$

and using (1.7) we obtain

$$\begin{aligned}A^{\mathcal{M},\pi_\theta,\gamma}(s,a) &= \frac{1-\gamma}{d_\mu^{\mathcal{M},\pi_\theta,\gamma}(s)}\left(\frac{1}{\pi_\theta(a|s)}\frac{\partial L_\lambda(\theta)}{\partial \theta_{s,a}} + \frac{\lambda}{|\mathcal{S}|}\left(1 - \frac{1}{\pi_\theta(a|s)|\mathcal{A}|}\right)\right) \leq \\ &\leq \frac{1-\gamma}{d_\mu^{\mathcal{M},\pi_\theta,\gamma}(s)}\left(2|\mathcal{A}|\frac{\partial L_\lambda(\theta)}{\partial \theta_{s,a}} + \frac{\lambda}{|\mathcal{S}|}\right) \leq \\ &\leq \frac{1-\gamma}{d_\mu^{\mathcal{M},\pi_\theta,\gamma}(s)}\left(2|\mathcal{A}|\frac{\lambda}{2|\mathcal{S}||\mathcal{A}|} + \frac{\lambda}{|\mathcal{S}|}\right) = \\ &= 2\frac{1-\gamma}{d_\mu^{\mathcal{M},\pi_\theta,\gamma}(s)}\frac{\lambda}{|\mathcal{S}|},\end{aligned}$$

and so we have proven (4.7). We can then complete the proof by recalling the Performance

Lemma (4.3).

$$\begin{aligned}
V_{\mathcal{M},\pi^*,\gamma}(\rho) - V_{\mathcal{M},\pi,\gamma}(\rho) &= \frac{1}{1-\gamma}\mathbb{E}_{s,a\sim d_\rho^{\mathcal{M},\pi^*,\gamma}}[A_{\mathcal{M},\pi,\gamma}(s,a)] = \\
&= \frac{1}{1-\gamma}\sum_{s\in\mathcal{S},a\in\mathcal{A}} d_\rho^{\mathcal{M},\pi^*,\gamma}(s)\pi^*(a|s)A_{\mathcal{M},\pi,\gamma}(s,a) \leq \\
&\leq \frac{1}{1-\gamma}\sum_{s\in\mathcal{S}} d_\rho^{\mathcal{M},\pi^*,\gamma}(s)\max_{a\in\mathcal{A}} A_{\mathcal{M},\pi,\gamma}(s,a) \leq \\
&\leq \frac{1}{1-\gamma}\sum_{s\in\mathcal{S}} d_\rho^{\mathcal{M},\pi^*,\gamma}(s)\frac{2\lambda(1-\gamma)}{d_\mu^{\mathcal{M},\pi_\theta,\gamma}(s)|\mathcal{S}|} = \\
&= \frac{2\lambda}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\frac{d_\rho^{\mathcal{M},\pi^*,\gamma}(s)}{d_\mu^{\mathcal{M},\pi_\theta,\gamma}(s)} \leq \\
&\leq 2\lambda\left\|\frac{d_\rho^{\mathcal{M},\pi^*,\gamma}}{d_\mu^{\mathcal{M},\pi_\theta,\gamma}}\right\|_\infty .
\end{aligned}$$

□

Previous Theorem shows the importance of balancing how the regularization parameter λ is set relative to the desired accuracy ϵ, as well as the importance of the initial distribution μ, in order to obtain global optimality.

Corollary 4.6. *Let a MDP $\mathcal{M}$, a discount factor γ and a distribution μ over state space be given. Assume $L_\lambda(\theta) = V_{\mathcal{M},\pi_\theta,\gamma}(\mu) + \lambda R(\theta)$ as defined in (4.1). Suppose the policy parameter θ is such that*

$$\|\nabla_\theta L_\lambda(\theta)\|_2 \leq \frac{\lambda}{2|\mathcal{S}||\mathcal{A}|}.$$

Then we have that for all initial distributions ρ

$$err_{\mathcal{M},\gamma,\rho}(\pi;\mu) = V_{\mathcal{M},\pi^*,\gamma}(\rho) - V_{\mathcal{M},\pi,\gamma}(\rho) \leq \frac{2\lambda}{1-\gamma}\left\|\frac{d_\rho^{\mathcal{M},\pi^*,\gamma}}{\mu}\right\|_\infty . \tag{4.8}$$

Proof. The corollary follows from Theorem 4.5 by observing that in the discounted state visitation distribution $d_\mu^{\mathcal{M},\pi,\gamma}$ the initial state s_0 has weight $(1-\gamma)$, hence it holds

$$d_\mu^{\mathcal{M},\pi,\gamma}(s) \geq (1-\gamma)\mu(s)$$

for any $s \in \mathcal{S}$. The infinity-norm inequality follows. □

Remark. Notice that here $err_{\mathcal{M},\gamma,\rho}(\pi;\mu)$ is as defined in (1.29) but with the added parameter μ to underline dependance. In usage, μ will refer to the sampling model distribution since we can calculate the objective function L directly with respect to it.

4.3 Gradient approximation

We now consider some of the most common gradient approximators used in literature ([7], [10]) and we will prove that, under appropriate hypothesis, they all satisfy the *Approximation Assumption.* With this, all of these approximators can be used in the final algorithm and all of them will satify the convergence bound results.

We recall Assumption 3.8: a gradient approximator $\hat{\nabla}_\theta : \mathbb{R}^\Theta \times \mathcal{T}^M \mapsto (\mathbb{R}^T)^\Theta$ is said to **satisfy Approximation Assumption** if and only if there exists constants $C_1, C_2, M_1, M_2 \in \mathbb{R}$ such that it holds:

- $\left\|\hat{\nabla}_\theta(L; \{\tau_m\})(\theta)\right\|_2 \leq C_1$ almost surely;
- $\langle \nabla_\theta L(\theta), \mathbb{E}_\mathcal{T}[\hat{\nabla}_\theta(L; \{\tau_m\})(\theta)]\rangle \geq C_2 \left\|\nabla_\theta L(\theta)\right\|_2^2 - \delta_k$ with $\delta_k \in \ell^2$;
- $\mathbb{E}_\mathcal{T}\left[\left\|\hat{\nabla}_\theta(L; \{\tau_m\})(\theta)\right\|_2^2\right] \leq M_1 + M_2 \left\|\nabla_\theta L(\theta)\right\|_2^2$.

It is worth noting that the computation of the gradient approximator of the objective function $L_\lambda(\theta)$ is non trivial for the value function part $V_{\mathcal{M},\pi,\gamma}$, while for the regularization term $R(\theta)$ it can be easily computed exactly. We choose to include the regularization term in the assumption anyway, because it does not complicate too much next results, but on the other way it extremely simplifies the final convergence result.

4.3.1 REINFORCE

Given a policy π_{θ^k} and a trajectory $\tau = \{(s_t, a_t, r_t)\}_{t=0,..,H}$ with rewards, the classical REINFORCE gradient estimator is

$$\hat{\nabla}_\theta V_{\mathcal{M},\pi_{\theta^k},\gamma} := \hat{Q}(s_0, a_0)\nabla \log \pi_{\theta^k}(a_0, s_0), \tag{4.9}$$

where $\hat{Q}(s_0, a_0) = (1-\gamma)\sum_{t=0}^{H-1} \gamma^t r_t$ is the one-trajectory estimator of the action value function in the starting state-action pair. This follow directly from the Policy Gradient Theorem.

This gradient vector actually has non-zero entries only for the $|\mathcal{A}|$ parameters corresponding to state s_0. Instead of doing this only for the starting state-action pair we could extend this argument to all the visited state-action pairs encountered along τ, provided that the estimator of the action value function is accurate enough. This gives better approximation quality to our gradient estimator, and translates in the following one-trajectory estimator:

$$\hat{\nabla}_\theta(L_\lambda; \tau)(\theta^k) := \sum_{t=0}^{\lfloor \beta H^k \rfloor} \hat{Q}(s_t, a_t)\nabla \log \pi_{\theta^k}(a_t, s_t) + \frac{\lambda}{SA}\sum_{s\in\mathcal{S}, a\in\mathcal{A}} \nabla_\theta \log \pi_{\theta^k}(a|s), \tag{4.10}$$

where

- $H^k \in \mathbb{N}$ is the length of the sampled trajectory;
- $\beta \in (0,1)$;
- $\hat{Q}(s_t, a_t) := (1-\gamma) \sum_{t'=t}^{H^k-1} \gamma^{t'-t} r_{t'}$.

Note that for $\beta = 0$ the gradient estimator is exactly the classical REINFORCE (4.9), because the summation reduces to the $t = 0$ term. Note also that the bigger the β the more inaccurate the action value estimator $\hat{Q}$ will be, at least for the last states appearing in τ.

Theorem 4.7. *Suppose that $H^k \geq \frac{2}{\min\{\beta, 1-\beta\}} \log_{\frac{1}{\gamma}}(k+1)$ for every $k \in \mathbb{N}$. Then the REINFORCE gradient estimator (4.10) satisfy Gradient Approximation Assumption*

Proof. We validate the three groups conditions (3.2), (3.4) and (3.4) in the Gradient Approximation Assumption in order.

Gradient estimation boundedness. Firstly recall the gradient expression of $\log(\pi_\theta)$ when the softmax parametrization is used:

$$\frac{\partial \log(\pi_{\theta^k}(a|s))}{\partial \theta_{s',a'}} = \mathbb{1}_{s=s'} \left(\mathbb{1}_{a=a'} - \pi_{\theta^k}(a'|s) \right).$$

It is clear that $\|\nabla_\theta \log \pi_{\theta^k}(a|s)\|_2 \leq 2$ for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, and we see that

$$\begin{aligned} \left\| \hat{\nabla}_\theta L_\lambda(\theta^k) \right\|_2 &\leq \sum_{t=0}^{\infty} \gamma^t \|\nabla_\theta \log \pi_{\theta^k}(a_t|s_t)\|_2 + \frac{\lambda}{|\mathcal{S}||\mathcal{A}|} \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \|\nabla_\theta \log \pi_{\theta^k}(a|s)\|_2 \leq \\ &\leq \frac{2}{1-\gamma} + 2\lambda. \end{aligned} \tag{4.11}$$

Hence we can take $C_1 := \frac{2}{1-\gamma} + 2\lambda$. Note that the exact same argument works for bounding also the norm of the true gradient $\left\|\nabla_\theta L_\lambda(\theta^k)\right\|_2 \leq C_1$.

Validation of nearly unbiasedness. Secondly, thanks to the non-stochasticity of the regularization term and to the tower property of expected value, notice that

$$\begin{aligned} \mathbb{E}_k[\hat{\nabla}_\theta L_\lambda(\theta^k)] &= \mathbb{E}_k \left[\sum_{t=0}^{\lfloor \beta H^k \rfloor} \gamma^t \mathbb{E}_k[\hat{Q}(s_t, a_t)|s_t, a_t] \nabla \log \pi_{\theta^k}(a_t, s_t) \right] + \frac{\lambda}{SA} \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \nabla_\theta \log \pi_{\theta^k}(a|s) \\ &= J_1 + J_2 + J_3, \end{aligned}$$

where

$$J_1 = \mathbb{E}_k \left[\sum_{t=0}^{\infty} \gamma^t (1-\gamma) \mathbb{E}_k \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'} | s_t, a_t \right] \nabla \log \pi_{\theta^k}(a_t, s_t) \right] + \frac{\lambda}{SA} \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \nabla_\theta \log \pi_{\theta^k}(a|s),$$

$$J_2 = -\mathbb{E}_k \left[\sum_{t=\lfloor \beta H^k \rfloor + 1}^{\infty} \gamma^t (1-\gamma) \mathbb{E}_k \left[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'} | s_t, a_t \right] \nabla_\theta \log \pi_{\theta^k}(a_t, s_t) \right],$$

$$J_3 = -\mathbb{E}_k \left[\sum_{t=0}^{\lfloor \beta H^k \rfloor} \gamma^t (1-\gamma) \mathbb{E}_k \left[\sum_{t'=H^k}^{\infty} \gamma^{t'-t} r_{t'} | s_t, a_t \right] \nabla_\theta \log \pi_{\theta^k}(a_t, s_t) \right].$$

Recall that here $\mathbb{E}_k[\cdot]$ is a short for $\mathbb{E}_{\tau \sim \mathbb{P}(\cdot | \mathcal{M}, \pi_{\theta^k}, \gamma)}[\cdot]$.

By the Policy Gradient Theorem we have that $J_1 = \nabla_\theta L_\lambda(\theta)$ is exactly the gradient. We can see that J_2 quantifies the error made by computing the gradient only on the first $\lfloor \beta H^k \rfloor$ state-action pairs encountered, while J_3 quantifies the error made by evaluating the Q-value only from a limited amount of future steps. Since $r(s,a) \in [0,1]$ we have

$$\begin{aligned} \|J_2\|_2 &\leq \frac{1-\gamma}{1-\gamma} \sum_{t=\lfloor \beta H^k \rfloor + 1}^{\infty} \gamma^t \left\| \nabla_\theta \log \pi_{\theta^k}(a_t, s_t) \right\|_2 \leq \\ &\leq \frac{2}{1-\gamma} \gamma^{\beta H^k}, \end{aligned}$$

and similarly

$$\begin{aligned} \|J_3\|_2 &\leq \sum_{t=0}^{\lfloor \beta H^k \rfloor} \gamma^t (1-\gamma) \left(\sum_{t'=H^k}^{\infty} \gamma^{t'-t} \right) \left\| \nabla_\theta \log \pi_{\theta^k}(a_t, s_t) \right\|_2 \leq \\ &\leq 2 \sum_{t=0}^{\lfloor \beta H^k \rfloor} \gamma^t \gamma^{(1-\beta) H^k} \leq \\ &\leq \frac{2}{1-\gamma} \gamma^{(1-\beta) H^k}. \end{aligned}$$

This leads to

$$\begin{aligned} \left\| \mathbb{E}_k[\hat{\nabla}_\theta L_\lambda(\theta^k)] - \nabla_\theta L_\lambda(\theta^k) \right\|_2 &= \|J_2 + J_3\|_2 \leq \\ &\leq \frac{4}{1-\gamma} \gamma^{\min\{\beta, 1-\beta\} H^k} \leq \frac{4}{(1-\gamma)} \frac{1}{(k+1)^2}, \end{aligned}$$

which implies that

$$\begin{aligned} \langle \nabla_\theta L_\lambda(\theta^k), \mathbb{E}[\hat{\nabla}_\theta L_\lambda(\theta^k)] \rangle &= \left\| \nabla_\theta L_\lambda(\theta^k) \right\|_2^2 + \langle \mathbb{E}[\hat{\nabla}_\theta L_\lambda(\theta^k)] - \nabla_\theta L_\lambda(\theta^k), \nabla_\theta L_\lambda(\theta^k) \rangle \geq \\ &\geq \left\| \nabla_\theta L_\lambda(\theta^k) \right\|_2 - \left\| \nabla_\theta L_\lambda(\theta^k) \right\|_2 \left\| \mathbb{E}[\hat{\nabla}_\theta L_\lambda(\theta^k)] - \nabla_\theta L_\lambda(\theta^k) \right\|_2^2 \geq \\ &\geq \left\| \nabla_\theta L_\lambda(\theta^k) \right\|_2^2 - C_1 \frac{4}{(1-\gamma)} \frac{1}{(k+1)^2}, \end{aligned}$$

where we used the Cauchy inequality. Hence we can take $C_2 := 1$ and $\delta_k := \frac{4C_1}{(1-\gamma)}\frac{1}{(k+1)^2}$. It is clear that $\delta_k \in \ell^2$.

Bounded second-order moment growth. Finally, we bound the second-order moment of the policy gradient. In the following, for a random vector $X = (X_1, ..., X_n) \in \mathbb{R}^n$ we define $Var_\theta(X) := \sum_{i=1}^n Var_\theta(X_i)$, where Var_θ denotes the conditional variance given the policy parameter θ, according to our previous notation

$$Var_\theta(X_i) := \mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\mathcal{M},\pi_\theta,\gamma)}\left[\left(X_i - \mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\mathcal{M},\pi_\theta,\gamma)}[X_i]\right)^2\right].$$

Now define the constant $\tilde{V} \in \mathbb{R}$ as the uniform upper bound on the variance of the policy gradient vector, i.e.,

$$\tilde{V} := \sup_{H\geq 0,\theta\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}} Var_\theta\left(\sum_{t=0}^{\lfloor\beta H\rfloor}\gamma^t\hat{Q}(s_t,a_t)\nabla\log\pi_\theta(a_t,s_t) + \frac{\lambda}{SA}\sum_{s\in\mathcal{S},a\in\mathcal{A}}\nabla_\theta\log\pi_\theta(a|s)\right),$$

where the trajectory $\tau = (s_0, a_0, r_0, ..., s_{H-1}, a_{H-1}, r_{H-1})$ is sampled from $\mathcal{M}$ following policy π_θ, and $\hat{Q}(s_t, a_t) = (1-\gamma)\sum_{t'=t}^{H}\gamma^{t'-t}r_{t'}$ as definition of the REINFORCE approximator. Notice that in the parenthesis there is our gradient estimator with non-fixed length H, in order to take the sup.

Then we have $Var_\theta(\hat{\nabla}_\theta L_\lambda(\theta^k) \leq \tilde{V}$ for any episode $k \geq 0$ by definition. In addition, since for any random vector $X \in \mathbb{R}^n$, we have

$$Var(X) \leq \sum_{i=1}^n \mathbb{E}[X_i^2] = \mathbb{E}[\|X\|_2^2],$$

we obtain by the same argument as (4.11), since it does not depend on specific H of θ, that

$$\tilde{V} \leq \left(\frac{2}{1-\gamma} + 2\lambda\right)^2 = \frac{4(1+\lambda(1-\gamma))^2}{(1-\gamma)^2}.$$

Finally, since for any random vector $X \in \mathbb{R}^n$ we have

$$\mathbb{E}\left[\|X\|_2^2\right] = \mathbb{E}\left[\sum_{i=1}^n X_i^2\right] = \sum_{i=1}^n\left(\mathbb{E}\left[X_i^2\right] + Var(X_i)\right) = \|\mathbb{E}[X]\|_2^2 + Var(X),$$

we obtain

$$\begin{aligned}\mathbb{E}_k\left[\left\|\hat{\nabla}_\theta L_\lambda(\theta^k)\right\|_2^2\right] &\leq \left\|\mathbb{E}_k\left[\hat{\nabla}_\theta L_\lambda(\theta^k)\right]\right\|_2^2 + \tilde{V} \leq \\ &\leq 2\left\|J_1\right\|_2^2 + 2\left\|J_2+J_3\right\|_2^2 + \tilde{V} \leq \\ &\leq 2\left\|J_1\right\|_2^2 + 2\left(\left\|J_2\right\|_2 + \left\|J_3\right\|_2\right)^2 + \tilde{V} \leq \\ &\leq 2\left\|J_1\right\|_2^2 + \frac{32}{(1-\gamma)^2}\gamma^{2\min\{\beta,1-\beta\}H^k} + \tilde{V} \leq \\ &\leq 2\left\|J_1\right\|_2^2 + \frac{32}{(1-\gamma)^2} + \frac{4(1+\lambda(1-\gamma))^2}{(1-\gamma)^2} = \\ &= 2\left\|\nabla_\theta L_\lambda(\theta^k)\right\|_2^2 + \frac{32}{(1-\gamma)^2} + \frac{4(1+\lambda(1-\gamma))^2}{(1-\gamma)^2},\end{aligned}$$

and hence we can take $M_2 := 2$ and $M_1 := \frac{32}{(1-\gamma)^2} + \frac{4(1+\lambda(1-\gamma))^2}{(1-\gamma)^2}$. This complete the proof. □

4.3.2 REINFORCE with baseline

A very standard extension to the REINFORCE gradient estimator is adding term, the *baseline* $b : \mathcal{S} \to \mathbb{R}$. The foundamental property of the baseline is that it is state dependent but action independent and so summing over the action and using the probability distribution constraint of the softmax parametrization we obtain that this summand add no bias to the estimator. On the other side, if chosen properly, it can significantly reduce the variance and so increase the speed of convergence. In this setting the gradient estimator is the following

$$\hat{\nabla}_\theta(L_\lambda;\tau)(\theta^k) := \sum_{t=0}^{\lfloor\beta H^k\rfloor}(\hat{Q}(s_t,a_t)-b(s_t))\nabla\log\pi_{\theta^k}(a_t,s_t) + \frac{\lambda}{SA}\sum_{s\in\mathcal{S},a\in\mathcal{A}}\nabla_\theta\log\pi_{\theta^k}(a|s). \quad (4.12)$$

where

- $H^k \in \mathbb{N}$ is the length of the sampled trajectory;
- $\beta \in (0,1)$;
- $\hat{Q}(s_t,a_t) := \sum_{t'=t}^{H^k-1}\gamma^{t'-t}r_{t'}$;
- $b : \mathcal{S} \to \mathbb{R}$.

Theorem 4.8. *Suppose that $H^k \geq \frac{2}{\min\{\beta,1-\beta\}}\log_{\frac{1}{\gamma}}(k+1)$ for every $k \in \mathbb{N}$ and suppose that there exists a constant $B > 0$ such that $|b(s)| \leq B$ for any $s \in \mathcal{S}$. Then the REINFORCE gradient estimator with baseline (4.12) satisfy Gradient Approximation Assumption.*

Proof. The proof is very similar to the one of Theorem 4.7, so we only highlight the differences. Firstly, similar to (4.11), we have

$$\left\|\hat{\nabla}_\theta L_\lambda(\theta^k)\right\|_2 \leq \frac{2+2B}{1-\gamma} + 2\lambda,$$

and hence we can take $C_1 := \frac{2+2B}{1-\gamma} + 2\lambda$. Secondly, with the same argument as (2.12), we have

$$\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\mathcal{M},\pi_{\theta^k},\gamma)}\left[\sum_{t=0}^{\lfloor\beta H^k\rfloor}\gamma^t b(s_t)\nabla_\theta \log \pi_{\theta^k}(a_t,s_t)\right] = 0.$$

Hence $\mathbb{E}_k[\hat{\nabla}_\theta L_\lambda(\theta^k)]$ is the same as in the proof of Theorem 4.7, and so we can take $C_2 := 1$ and $\delta_k := \frac{4C_1}{(1-\gamma)}\frac{1}{(k+1)^2}$, as before.

Finally, by similar definition, we can write

$$\tilde{V}_b := \sup_{H\geq 0,\theta\in\mathbb{R}^{|\mathcal{S}||\mathcal{A}|}} Var_\theta\left(\sum_{t=0}^{\lfloor\beta H\rfloor}\gamma^t(\hat{Q}(s_t,a_t)-b(s_t))\nabla\log\pi_\theta(a_t,s_t) + \frac{\lambda}{SA}\sum_{s\in\mathcal{S},a\in\mathcal{A}}\nabla_\theta\log\pi_\theta(a|s)\right),$$

and, with the same argument as before, we have $\tilde{V}_b \leq \left(\frac{2+2B}{1-\gamma}+2\lambda\right)^2 = \frac{4(1+B+\lambda(1-\gamma))^2}{(1-\gamma)^2}$, and thus setting $M_2 := 2$ and $M_1 := \frac{32}{(1-\gamma)^2} + \tilde{V}_b$ completes the proof. □

Those estimators, as said, are single-trajectory based. A natural extension is to consider minibatches with $M > 1$, and compute a gradient estimator based on M different trajectories $\{\tau_m\}_{m=1,\dots,M}$. This can be made with a simple average:

$$\hat{\nabla}_\theta(L_\lambda;\{\tau_m\}_{m=1,\dots,M})(\theta^k) = \frac{1}{M}\sum_{m=1}^{M}\hat{\nabla}_\theta(L_\lambda;\tau_m)(\theta^k) \tag{4.13}$$

where $\hat{\nabla}_\theta(L_\lambda;\tau_m)(\theta^k)$ is an arbitrary single-trajectory gradient estimator. It can be shown that, with the REINFORCE with baseline, Theorem 4.8 still holds with all the same constants excepts for $M_1 = \frac{32}{(1-\gamma)^4} + \frac{\tilde{V}_B}{M}$, that is smaller thanks to the reduction of the variance of the average.

Remark. With further analysis it comes out that the best choice, in terms of variance reduction for the baseline $b(s)$ is the value function $V_{\mathcal{M},\pi,\gamma}(s)$. This can be seen as a *critic* of an actor-critic estimation.

4.3.3 Actor-critic

The general idea of actor-critic methods is to make use of two different estimators, and thus of two set of parameters θ and w, one for the policy π_θ, referred to as the *actor*, and one for the value function V_w, referred to as the *critic*.

The REINFORCE gradient estimator (4.10) is often called also Monte Carlo since in order to get an estimate of value function in state s_t it uses the complete return $\hat{Q}(s_t, a_t)$, at which a baseline critic may or may not be added. Instead, a more value function based critic is the one-step **Temporal Difference** (TD) which estimates the value function in a state s_t using one-step ahead estimates, that is

$$\text{one-step } V(s_t) \approx r_t + \gamma V_w(s_{t+1}).$$

This leads to the actor-critic one-step TD gradient estimator

$$\hat{\nabla}_\theta L_\lambda(\theta^k) := \sum_{t=0}^{H^k-1} (r_t + \gamma V_w(s_{t+1}) - V_w(s_t)) \nabla \log \pi_{\theta^k}(a_t, s_t) + \frac{\lambda}{SA} \sum_{s\in\mathcal{S}, a\in\mathcal{A}} \nabla_\theta \log \pi_{\theta^k}(a|s),$$

where $H^k \in \mathbb{N}$ is the length of the sampled trajectory.

Various types of hypothesis can be made over the estimator V_w in order to guarantee the validity of Approximation Assumption. For example, this result is almost trivial if $V_w(s)$ is an unbiased estimator of $V_{\mathcal{M},\pi_{\theta^k},\gamma}(s)$ with bounded second order moment. Weaker but still sufficient hypotheses may be discussed, but are beyond the scope of this work.

4.3.4 TD(λ)

A very standard extension of the one-step TD is the n-step TD:

$$n\text{-step } V(s_t) \approx \sum_{t'=t}^{t+n-1} \gamma^{t'-t} r_{t'} + \gamma^n V_w(s_{t+n}),$$

which leads to the actor-critic n-step TD gradient estimator

$$[n\text{-TD}\hat{\nabla}_\theta] L_\lambda(\theta^k) := \sum_{t=0}^{H^k-1} (n\text{-step } V(s_t) - V_w(s_t)) \nabla \log \pi_{\theta^k}(a_t, s_t) + \frac{\lambda}{SA} \sum_{s\in\mathcal{S}, a\in\mathcal{A}} \nabla_\theta \log \pi_{\theta^k}(a|s).$$

This further generalizes to the TD(λ) gradient estimator, that is defined as

$$[\text{TD}(\lambda)\hat{\nabla}_\theta] L_\lambda := (1-\lambda) \sum_{n=1}^{\infty} \lambda^{n-1} [n\text{-TD}\hat{\nabla}_\theta] L_\lambda.$$

TD(λ) is a sort of continuous bridge between TD and Monte Carlo estimators. Moreover, there is a number of possibilities in which a gradient estimator can be obtained, despite an exhaustive list is almost impossible to made. We choose to only cite some of them in order to maintain focus on the algorithm convercenge. More details about gradient approximators can be found in [7].

Chapter 5

Algorithms

In this Chapter we state some algorithms and then provide theoretical guarantees on their convergence rate. In the next Chapter we implement them and test in some practical cases.

We focus on two classes of problem, both assuming the μ-reset model as sampling model. In the first case we assume a **coverage hypothesis** of $\mu(s) > 0$ for all $s \in \mathcal{S}$. This strong assumption is very common in literature and almost avoid the exploration problem. In the second case instead we assume a sampling model with only one initial state s_0 for every episode, that is $\mu(s_0) = 1$ and $\mu(s) = 0$ for all $s \neq s_0$.

5.1 Assuming coverage hypothesis

Given a MDP with a μ-reset sampling model satisfying *coverage hypothesis*, in order to find an optimal policy we propose the following algorithm.

Algorithm 1 Policy Gradient Method

```
Input: regularization parameter $\lambda$, step-sizes $\alpha^k$ for $k \geq 0$.
Set $\theta^0$ such that $\pi_{\theta^0}(a|s) \geq \epsilon_{pp}$ for each $s, a \in \mathcal{S} \times \mathcal{A}$.
for $k = 0, 1, ...$ do
    Choose trajectory lenght $H^k$.
    Sample trajectory $\tau^k$ of length $H^k$ from $\mathcal{M}$ following $\pi_{\theta^k}$.
    Compute approximate gradient $\hat{\nabla}_\theta L_\lambda(\theta^k)$ using trajectory $\tau^k$.
    Update $\theta^{k+1} = \theta^k + \alpha^k \hat{\nabla}_\theta L_\lambda(\theta^k)$.
end for
```

We measure the error with respect to an arbitrary distribution ρ over state space.

Since we force trajectories to stop after H^k timestep, the expected value of the measured cumulative discounted reward will not be the value function in its classical definition (1.11), due to the truncation of the summation. What we can actually measure and is involved in the regret function is the following

$$\tilde{V}_{\mathcal{M},\pi_{\theta^k},\gamma}(\rho) := (1-\gamma)\mathbb{E}_{s_0\sim\rho}\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0)}\left[\sum_{t=0}^{H^k-1}\gamma^t r(s_t,a_t)\right]. \tag{5.1}$$

So the formulation of the regret (1.30) in this specific context is

$$\mathit{regret}(K) := \mathit{regret}_{\mathcal{M},\gamma,\rho}((\pi_{\theta^k})_{k<K};K) = \sum_{k=0}^{K-1}\left(V^*_{\mathcal{M},\gamma}(\rho) - \tilde{V}_{\mathcal{M},\pi_{\theta^k},\gamma}(\rho)]\right) \tag{5.2}$$

Theorem 5.1. *Len an MDP $\mathcal{M}$, a discount factor $\gamma\in[0,1)$ and a μ-reset sampling model with $\mu(s)>0$ for any $s\in\mathcal{S}$ be given. Assume to follow Algorithm 1 and suppose that the chosen gradient estimator $\hat{\nabla}_\theta$ satisfies the Gradient Approximation Assumption with constants $C_1, C_2, M_1, M_2\in\mathbb{R}$ and $\{\delta_k\}_{k\in\mathbb{N}}\subset\mathbb{R}$.*
Choose $H^k\geq\log_{\frac{1}{\gamma}}(k+1)$ and choose $\alpha^k = C_\alpha\frac{1}{\sqrt{k+3}\log_2(k+3)}$ for some $C_\alpha\in(0,\frac{C_2}{M_2\beta_\lambda})$.
For any $\epsilon>0$ we choose $\lambda := \epsilon\frac{(1-\gamma)}{2\left\|\frac{d_\rho^{\mathcal{M},\pi^*,\gamma}}{\mu}\right\|_\infty}$. Then, for any $\delta\in(0,1)$, for any $k\in\{0,...,T-1\}$ with probability at least $1-\delta$, we have:

$$\mathit{regret}(k)\leq D_3\left(D_1+D_2\sqrt{\log\left(\frac{2}{\delta}\right)}\right)\frac{1}{\epsilon^2}\sqrt{k+1}\log(k+3)+\epsilon(k+1)+\log(k+1)+1,$$

$$\begin{aligned} \textit{where}\quad D_1 &= V^*_{\mathcal{M},\gamma} - L_\lambda(\theta^0) + C_\alpha^2\|\delta_k\|_{l^2}^2 + M_1\frac{\beta_\lambda}{2}C_\alpha^2;\\ D_2 &= 2C_1C_\alpha\sqrt{4\left(\frac{2}{1-\gamma}+2\lambda\right)^2+\beta_\lambda^2C_1^2C_\alpha^2};\\ D_3 &= \frac{64|\mathcal{S}|^2|\mathcal{A}|^2\left\|\frac{d_\rho^{\pi^*}}{\mu}\right\|_\infty^2}{C_\alpha C_2(1-\gamma)^2}. \end{aligned}$$

Proof. By the strongly smoothness of L_λ it holds

$$\left\|\nabla_\theta L_\lambda(\theta) - \nabla_\theta L_\lambda(\theta')\right\|_2 \leq \beta_\lambda\left\|\theta-\theta'\right\|_2,$$

where $b_\lambda = \frac{8}{(1-\gamma)^2}+\frac{2\lambda}{|\mathcal{S}|}$.
An equivalent condition is that

$$L_\lambda(\theta) - L_\lambda(\theta') \leq \nabla_\theta L_\lambda(\theta')^T(\theta-\theta') - \frac{\beta_\lambda}{2}\left\|\theta-\theta'\right\|_2^2.$$

Using the iteration definition $\theta^{k+1} = \theta^k + \alpha^k \hat{\nabla}_\theta L_\lambda(\theta^k)$ we have

$$\begin{aligned} -L_\lambda(\theta^{k+1}) + L_\lambda(\theta^k) &\leq -\nabla_\theta L_\lambda(\theta^k)^T(\theta^{k+1} - \theta^k) + \frac{\beta_\lambda}{2} \left\| \theta^{k+1} - \theta^k \right\|_2^2 \\ &= \underbrace{-\alpha^k \nabla_\theta L_\lambda(\theta^k)^T \hat{\nabla}_\theta L_\lambda(\theta^k) + \frac{\beta_\lambda(\alpha^k)^2}{2} \left\| \hat{\nabla}_\theta L_\lambda(\theta^k) \right\|_2^2}_{=:Y_k}. \end{aligned}$$

So with this definition of the random variable Y_k we have

$$L_\lambda(\theta^k) - L_\lambda(\theta^{k+1}) \leq Y_k. \tag{5.3}$$

We now define the two random variables

$$Z_k := Y_k - \mathbb{E}_k[Y_k],$$

$$X_k := \sum_{k=0}^{K-1} Z_k, \qquad X_0 := 0.$$

Using the linearity of the expected value and rearranging the terms we get

$$\begin{aligned} Z_k = Y_k - \mathbb{E}[Y_k] = -\alpha^k \nabla_\theta L_\lambda(\theta^k)^T &\left(\hat{\nabla}_\theta L_\lambda(\theta^k) - \mathbb{E}_k \left[\hat{\nabla}_\theta L_\lambda(\theta^k) \right] \right) + \\ &+ \frac{\beta_\lambda(\alpha^k)^2}{2} \left(\left\| \hat{\nabla}_\theta L_\lambda(\theta^k) \right\|_2^2 - \mathbb{E}_k \left[\left\| \hat{\nabla}_\theta L_\lambda(\theta^k) \right\|_2^2 \right] \right); \end{aligned}$$

$$\begin{aligned} |X_k - X_{k-1}| = |Z_k| &\leq \alpha^k \left\| \nabla_\theta L_\lambda(\theta^k) \right\|_2 \left\| \mathbb{E}_k \left[\hat{\nabla}_\theta L_\lambda(\theta^k) \right] - \hat{\nabla}_\theta L_\lambda(\theta^k) \right\|_2 \\ &\quad + \frac{\beta_\lambda(\alpha^k)^2}{2} \left| \mathbb{E}_k \left[\left\| \hat{\nabla}_\theta L_\lambda(\theta^k) \right\|_2^2 \right] - \left\| \hat{\nabla}_\theta L_\lambda(\theta^k) \right\|_2^2 \right| \\ &\leq \underbrace{2\alpha^k C_1 \left(\frac{2}{1-\gamma} + 2\lambda \right) + \beta_\lambda (\alpha^k)^2 C_1^2}_{=:c_k}, \end{aligned}$$

where the last step follows from the Approximation Assumption on $\hat{\nabla}_\theta$ and from the boundedness of $\left\| \nabla_\theta L_\lambda(\theta^k) \right\|_2$ that we have shown in the previous Sections.
This immediately implies

$$\mathbb{E}[|X_K|] \leq \sum_{k=0}^{K} c_k < \infty. \tag{5.4}$$

Then, let $\mathcal{F}_K$ be the filtration up to episode K, *i.e.*, the σ-algebra generated by all iterations $\{\theta^0, ..., \theta^K\}$ up to the K-th one.

$$\mathbb{E}[X_{K+1}|\mathcal{F}_K] = \sum_{k=0}^{K-1} Z_k + \mathbb{E}\left[Y_K + \mathbb{E}_K[Y_K] | \mathcal{F}_K \right] = X_K. \tag{5.5}$$

Notice that the second equality makes use of the fact that given the current policy the corresponding sampled trajectory is conditionally independent of all previous policies and trajectories.

Now (5.4) and (5.5) immediately implies that X_k is a martingale. Hence by the Azuma-Hoeffding inequality, for any $c > 0$:

$$\mathbb{P}(|X_k| \geq c) \leq 2e^{\frac{-c^2}{2\sum_{k=0}^{\infty} c_k^2}}, \tag{5.6}$$

where

$$\begin{aligned}
\sum_{k=0}^{\infty} c_k^2 = \sum_{k=0}^{\infty} \left(2\alpha^k C_1 \left(\frac{2}{1-\gamma} + 2\lambda\right) + \beta_\lambda (\alpha^k)^2 C_1^2\right)^2 \leq \\
\leq 8C_1^2 \left(\frac{2}{1-\gamma} + 2\lambda\right)^2 \sum_{k=0}^{\infty} (\alpha^k)^2 + 2\beta_\lambda^2 C_1^4 \sum_{k=0}^{\infty} (\alpha^k)^4 \leq \\
\leq 8C_1^2 \left(\frac{2}{1-\gamma} + 2\lambda\right)^2 \left\|\alpha^k\right\|_{l^2}^2 + 2\beta_\lambda^2 C_1^4 \left\|\alpha^k\right\|_{l^4}^4 < \infty.
\end{aligned}$$

We now go back to the inequality (5.3) and use the definition of Z_k,

$$L_\lambda(\theta^k) - L_\lambda(\theta^{k+1}) \leq Y_k = Z_k + \mathbb{E}[Y_k],$$

from which we obtain

$$\begin{aligned}
L_\lambda(\theta^k) - L_\lambda(\theta^{k+1}) &\leq Z_k - \alpha^k \nabla_\theta L_\lambda(\theta^k)^T \mathbb{E}\left[\hat{\nabla}_\theta L_\lambda(\theta^k)\right] + \frac{\beta_\lambda (\alpha^k)^2}{2} \mathbb{E}\left[\left\|\hat{\nabla}_\theta L_\lambda(\theta^k)\right\|_2^2\right] \leq \\
&\leq Z_k - \alpha^k \left(C_2 \left\|\nabla_\theta L_\lambda(\theta^k)\right\|_2^2 - \delta_k\right) + \frac{\beta_\lambda (\alpha^k)^2}{2} \left(M_1 + M_2 \left\|\nabla_\theta L_\lambda(\theta^k)\right\|_2^2\right) = \\
&= Z_k + \alpha^k \delta_k + M_1 \frac{\beta_\lambda (\alpha^k)^2}{2} - \alpha^k \left(C_2 - M_2 \frac{\beta_\lambda \alpha^k}{2}\right) \left\|\nabla_\theta L_\lambda(\theta^k)\right\|_2^2 \leq \\
&\leq Z_k + \alpha^k \delta_k + M_1 \frac{\beta_\lambda (\alpha^k)^2}{2} - \alpha^k \frac{C_2}{2} \left\|\nabla_\theta L_\lambda(\theta^k)\right\|_2^2.
\end{aligned}$$

Then by summing up these inequalities for all the T episodes, we obtain that

$$\begin{aligned}
L_\lambda(\theta^T) - L_\lambda(\theta^0) &\leq \sum_{k=0}^{T-1} Z_k - \frac{C_2}{2} \sum_{k=0}^{T-1} \alpha^k \left\|\nabla_\theta L_\lambda(\theta^k)\right\|_2^2 + \sum_{k=0}^{\infty} \alpha^k \delta_k + M_1 \frac{\beta_\lambda}{2} \sum_{k=0}^{\infty} (\alpha^k)^2 \leq \\
&\leq X_T - \frac{C_2}{2} \sum_{k=0}^{T-1} \alpha^k \left\|\nabla_\theta L_\lambda(\theta^k)\right\|_2^2 + \left\|\alpha^k\right\|_{l^2}^2 \|\delta_k\|_{l^2}^2 + M_1 \frac{\beta_\lambda}{2} \left\|\alpha^k\right\|_{l^2}^2.
\end{aligned}$$

From this, rearranging the terms we obtain that for every $0 \leq K \leq T$

$$\begin{aligned}
\sum_{k=0}^{K} \alpha^k \left\| \nabla_\theta L_\lambda(\theta^k) \right\|_2^2 &\leq \sum_{k=0}^{T-1} \alpha^k \left\| \nabla_\theta L_\lambda(\theta^k) \right\|_2^2 \leq \\
&\leq \frac{2}{C_2} \left(\sup_{\theta \in \Theta} L_\lambda(\theta) - L_\lambda(\theta^0) + X_T + \left\| \alpha^k \right\|_{l^2}^2 \|\delta_k\|_{l^2}^2 + M_1 \frac{\beta_\lambda}{2} \left\| \alpha^k \right\|_{l^2}^2 \right) \leq \\
&\leq \frac{2}{C_2} \left(\underbrace{V^*_{\mathcal{M},\gamma} - L_\lambda(\theta^0) + C_\alpha^2 \|\delta_k\|_{l^2}^2 + M_1 \frac{\beta_\lambda}{2} C_\alpha^2}_{=:D_1} + X_T \right),
\end{aligned}$$

where we use the fact that the regularization term $R(\theta) \leq 0$ for all $\theta \in \Theta$.

Hence we have

$$\sum_{k=0}^{K} \alpha^k \left\| \nabla_\theta L_\lambda(\theta^k) \right\|_2^2 \leq \frac{2}{C_2}(D_1 + X_T), \tag{5.7}$$

where D_1 is a constant as defined above and X_T is the martingale previously defined. We now establish the regret bound using (5.7). Fix $K \in \{0, ..., T-1\}$. Let

$$I^+ := \left\{ k \in \{0, ..., K\} \middle| \left\| \nabla_\theta L_\lambda(\theta^k) \right\|_2 \geq \frac{\lambda}{2|\mathcal{S}||\mathcal{A}|} \right\}$$

be the **bad episodes**. Note that during the algorithm we do not know which episodes are bad and which are not since we do not have access to the exact gradient. For simplicity, assume for now that $|I^+| > 0$. Then since α^k is decreasing in k, we have

$$\begin{aligned}
\frac{2}{C_2}(D_1 + X_T) &\geq \frac{\lambda^2}{4|\mathcal{S}|^2|\mathcal{A}|^2} \sum_{k \notin I^+} \alpha^k \geq \\
&\geq \frac{\lambda^2}{4|\mathcal{S}|^2|\mathcal{A}|^2} \sum_{k=K-|I^+|+1}^{K} \alpha^k = \\
&= \epsilon^2 \frac{(1-\gamma)^2}{16|\mathcal{S}|^2|\mathcal{A}|^2 \left\| \frac{d_\rho^{\pi^*}}{\mu} \right\|_\infty^2} \sum_{k=K-|I^+|+1}^{K} \alpha^k \geq \\
&\geq \epsilon^2 \frac{(1-\gamma)^2}{16|\mathcal{S}|^2|\mathcal{A}|^2 \left\| \frac{d_\rho^{\pi^*}}{\mu} \right\|_\infty^2} |I^+| \alpha^K.
\end{aligned}$$

Hence we have a bound on the number of bad episodes

$$|I^+| \leq (D_1 + X_T) \frac{32|\mathcal{S}|^2|\mathcal{A}|^2 \left\| \frac{d_\rho^{\pi^*}}{\mu} \right\|_\infty^2}{C_2(1-\gamma)^2} \frac{1}{\epsilon^2} \frac{1}{\alpha^K}. \tag{5.8}$$

And now thanks to the boundedness of the total discounted reward (1.9) for the bad espisodes and thanks to the Gradient Domination Theorem for the non bad episodes we have:

$$V^*_{\mathcal{M},\gamma}(\rho) - V_{\mathcal{M},\pi_{\theta^k},\gamma}(\rho) \leq \begin{cases} \dfrac{2\lambda}{1-\gamma}\left\|\dfrac{d_\rho^{\mathcal{M},\pi^*,\gamma}}{\mu}\right\|_\infty & \forall k \notin I^+ \\ 1 & \forall k \in I^+ \end{cases} = \begin{cases} \epsilon & \forall k \notin I^+ \\ 1 & \forall k \in I^+ \end{cases}.$$

Hence, summing over the episodes,

$$\begin{aligned}\sum_{k\leq K} V^*_{\mathcal{M},\gamma}(\rho) - V_{\mathcal{M},\pi_{\theta^k},\gamma}(\rho) &= \sum_{k\in I^+} V^*_{\mathcal{M},\gamma}(\rho) - V_{\mathcal{M},\pi_{\theta^k},\gamma}(\rho) + \sum_{k\notin I^+} V^*_{\mathcal{M},\gamma}(\rho) - V_{\mathcal{M},\pi_{\theta^k},\gamma}(\rho) \leq \\ &\leq |I^+|\frac{1}{1-\gamma} + (K+1-|I^+|)\epsilon \leq \\ &\leq (D+X_T)\frac{32|\mathcal{S}|^2|\mathcal{A}|^2\left\|\frac{d_\rho^{\pi^*}}{\mu}\right\|_\infty^2}{C_2(1-\gamma)^2}\frac{1}{\epsilon^2}\frac{1}{\alpha^K} + (K+1)\epsilon. \end{aligned} \tag{5.9}$$

We consider also the difference beetwen the true value function and the truncated one: this can be bound thanks to the logarithmic length assumption we made in the hypothesis.

$$\begin{aligned}\sum_{k\leq K} V_{\mathcal{M},\pi_{\theta^k},\gamma}(\rho) - \tilde{V}_{\mathcal{M},\pi_{\theta^k},\gamma}(\rho) &= \sum_{k\leq K}(1-\gamma)\mathbb{E}_{s_0\sim\rho}\mathbb{E}_{\tau\sim\mathbb{P}(\cdot|\pi,\mathcal{M},s_0)}\left[\sum_{t=H^k}^{\infty}\gamma^t r(s_t,a_t)\right] \leq \\ &\leq \sum_{k\leq K}(1-\gamma)\sum_{t=H^k}^{\infty}\gamma^t = \\ &= \sum_{k\leq K}(1-\gamma)\gamma^{H^k}\sum_{t=0}^{\infty}\gamma^t = \\ &= \sum_{k\leq K}\gamma^{H^k} \leq \\ &\leq \sum_{k\leq K}\frac{1}{k+1} \leq 1+\log(K+1). \end{aligned} \tag{5.10}$$

Combining (5.9) and (5.10) immediately implies that

$$\begin{aligned} regret(K) &= \sum_{k\leq K} V^*_{\mathcal{M},\gamma}(\rho) - V_{\mathcal{M},\pi_{\theta^k},\gamma}(\rho) + \sum_{k\leq K} V_{\mathcal{M},\pi_{\theta^k},\gamma}(\rho) - \tilde{V}_{\mathcal{M},\pi_{\theta^k},\gamma}(\rho) \leq \\ &\leq (D_1+X_T)\frac{32|\mathcal{S}|^2|\mathcal{A}|^2\left\|\frac{d_\rho^{\pi^*}}{\mu}\right\|_\infty^2}{C_2(1-\gamma)^2}\frac{1}{\epsilon^2}\frac{1}{\alpha^K} + (K+1)\epsilon + 1 + \log(K+1). \end{aligned} \tag{5.11}$$

Now if $|I^+| = 0$, then we immediately have that

$$regret(K) \leq (K+1)\epsilon + 1 + \log(K+1),$$

and hence (5.11) always holds.

Finally, by (5.6), we have that with probability at most δ

$$|X_T| \geq \sqrt{2\log\left(\frac{2}{\delta}\right)\sum_{k=0}^{\infty} c_k^2},$$

that combined with (5.11) leads to

$$regret(K) \leq \left(D_1 + \sqrt{2\log\left(\frac{2}{\delta}\right)\sum_{k=0}^{\infty} c_k^2}\right) \frac{32|\mathcal{S}|^2|\mathcal{A}|^2 \left\|\frac{d_\rho^{\pi^*}}{\mu}\right\|_\infty^2}{C_2(1-\gamma)^2} \frac{1}{\epsilon^2}\frac{1}{\alpha^K} + (K+1)\epsilon + 1 + \log(K+1),$$

with probaility at least $1-\delta$ for all $K \in \{0, ..., T-1\}$.

We now make use of the definition of the choosen α^K and of the constants D_2 and D_3 defined in the statement of the Theorem. This gives

$$regret(K) \leq D_3\left(D_1 + D_2\sqrt{\log\left(\frac{2}{\delta}\right)}\right)\frac{1}{\epsilon^2}\sqrt{K+1}\log(K+3) + (K+1)\epsilon + 1 + \log(K+1),$$

with probability at least $1-\delta$ for all $K \in \{0, ..., T-1\}$. And this complete the proof. □

Notice that this result holds for any $\epsilon > 0$, provided that λ is defined accordingly.

Corollary 5.2. *Let an MDP $\mathcal{M}$, a discount factor $\gamma \in [0,1)$ and a μ-reset sampling model with $\mu(s) > 0$ for any $s \in \mathcal{S}$ be given. Suppose that $\hat{\nabla}_\theta$ satifies the Gradient Approximation Assumption.*
Then Algorithm 1 is a function that takes as input the number of episodes $K \in \mathbb{N}$ and returns a sequence of policies $\{\pi^k\}_{k=1,...,K}$ such that

$$\frac{1}{K}\sum_k err_{\mathcal{M},\gamma,\rho}(\pi^k) \leq C\left\|\frac{1}{\mu}\right\|_\infty^2 \left(1 + \sqrt{\log\frac{1}{\delta}}\right) K^{-\frac{1}{6}}\log(K)$$

with probability at least $1-\delta$. Here C is a constant that only depends on $\mathcal{M}$ and γ.

Proof. It suffices to set

- $H^k \geq \log_{\frac{1}{\gamma}}(k+1)$
- $\alpha^k = C_\alpha \frac{1}{\sqrt{k+3}\log_2(k+3)}$ for some $C_\alpha \in (0, \frac{C_2}{M_2\beta_\lambda})$

- $\lambda = K^{-\frac{1}{6}} \frac{(1-\gamma)}{2\left\|\frac{d_\rho^{\pi^*}}{\mu}\right\|_\infty}$

As in the hypothesis of Theorem 5.1, choose $H^k \geq \log_{\frac{1}{\gamma}}(k+1)$ and $\alpha^k = C_\alpha \frac{1}{\sqrt{k+3}\log_2(k+3)}$ for some $C_\alpha \in (0, \frac{C_2}{M_2 \beta_\lambda})$. Then set $\epsilon := K^{-\frac{1}{6}}$ and, consequently, set $\lambda := \epsilon \frac{(1-\gamma)}{2\left\|\frac{d_\rho^{\pi^*}}{\mu}\right\|_\infty}$.

The result follows from Theorem 5.1 by noticing that $d_\rho^{\pi^*}(s) \leq 1$ for every distribution ρ and for every state $s \in \mathcal{S}$. Thus

$$\left\|\frac{d_\rho^{\pi^*}}{\mu}\right\|_\infty \leq \left\|\frac{1}{\mu}\right\|_\infty .$$

□

5.1.1 Anytime regret bound

From Theorem 5.1 one can build a phased algorithm with an anytime regret bound. Here we do not have to specify a tolerance parameter ϵ at the beginning (in order to set the regolarization parameter λ accordingly), but the error will go to zero as the number of episodes goes to infinity, with sublinear rate.

First of all note that every parameter that was k-dependant before, now is (l,k)-dependant, where l is the phase number and k is the episode number in that phase. In particular we now refer to $\theta^{l,k}$ for the parameters, $\tau^{l,k}$ for the sampled trajectory, $\alpha^{l,k}$ for the step size, $H^{l,k}$ for the episode length and so on.
Notice also that the number of episodes in each phase T_l is now phase-dependant, and so are the regularization parameter λ^l and the tolerance parameter ϵ^l.

Algorithm 2 Phased Gradient Method

```
Input: phase lengths $T_l$, regularization parameters $\lambda^l$, step-sizes $\alpha^{l,k}$ for $l, k \geq 0$
Set $\theta^{0,0}$ such that $\pi_{\theta^{0,0}}(a|s) \geq \epsilon_{pp}$ for each $s, a \in \mathcal{S} \times \mathcal{A}$.
for phase $l = 0, 1, 2, ...$ do
    for episode $k = 0, 1, ..., T_l - 1$ do
        Sample trajectory $\tau^{l,k}$ from $\mathcal{M}$ following $\pi_{\theta^{l,k}}$.
        Compute approximate gradient $\hat{\nabla}_\theta L_{\lambda^l}(\theta^{l,k})$ using trajectory $\tau^{l,k}$.
        Update $\theta^{l,k+1} = \theta^{l,k} + \alpha^{l,k} \hat{\nabla}_\theta L_{\lambda^l}(\theta^{l,k})$.
    end for
    Set $\theta^{l+1,0}$ close to $\theta^{l,T_l-1}$ such that $\pi_{\theta^{l+1,0}}(a|s) \geq \epsilon_{pp}$ for each $s, a \in \mathcal{S} \times \mathcal{A}$.
end for
```

The key idea for convergence results in Algorithm 2 is to use a doubling trick in the number of episodes sampled, combined with the right relative rate of variations of the relevant parameter λ and ϵ.

Remark. Note that here the k-th episode in phase l correspond to he n-th episode in the original indexing, which count the total number of call to the sampling model, $n = \sum_{j=0}^{l-1} T_j + k$, so define

$$\begin{aligned} \mathbf{B} &: \mathbb{N} \times \mathbb{N} \to \mathbb{N} \\ (l,k) &\mapsto \sum_{j=0}^{l-1} T_j + k \end{aligned}$$

The mapping $\mathbf{B}$ is a bijection with the phase-episode domain of our algorithm, $l \in \mathbb{N}, k \in \{0, ..., T_l - 1\}$, and we denote it inverse by $\mathbf{G} : \mathbb{N} \to \mathbb{N} \times \mathbb{N}$.

First, we need to reframe the *regret* expression to this phased setting, recalling that, in the end, we are interested in bounding the error with respect to the total number of episodes.

$$regret_l(K) := regret_{\mathcal{M},\gamma,\rho}((\pi_{\theta^{l,k}})_{k<K}; K) = \sum_{k=0}^{K-1} \left(V^*_{\mathcal{M},\gamma}(\rho) - \tilde{V}_{\mathcal{M},\pi_{\theta^{l,k}},\gamma}(\rho)] \right) \qquad \forall l \geq 0,$$

$$regret(N) := \sum_{l=0}^{l_N - 1} regret_l(T_l - 1) + regret_{l_N}(k_N), \tag{5.12}$$

where $(l_N, k_N) = \mathbf{G}(N)$.

Theorem 5.3. *Let an MDP $\mathcal{M}$, a discount factor γ and a μ-reset sampling model with $\mu(s) > 0$ for any $s \in \mathcal{S}$ be given. Assume to follow Algorithm 2 and suppose that the chosen $\hat{\nabla}_\theta$ satifies the Gradient Approximation Assumption with constants $C_1, C_2, M_1, M_2, \{\delta_k\}_{k\in\mathbb{N}}$. Choose $H^{l,k} \geq \log_{\frac{1}{\gamma}}(k+1)$ for any l, k.*

Set $\bar{\lambda} = \frac{1-\gamma}{2\left\|\frac{d^{\pi^*}_\rho}{\mu}\right\|_\infty}$, $T_0 \geq 1$ and $\underline{C}^\alpha \in (0, \frac{C_2}{M_2 \beta_{\bar{\lambda}}})$. Then for any phase $l \geq 0$ set recursively

$$\begin{aligned} &T_l = 2T_{l-1} = 2^l T_0, \\ &\epsilon_l = T_l^{-\frac{1}{6}}, \\ &\lambda^l = \epsilon_l \bar{\lambda}, \\ &\alpha^{l,k} = C_{l,\alpha} \frac{1}{\sqrt{k+3}\log_2(k+3)} \qquad \textit{for some } C_{l,\alpha} \in \left[\underline{C}^\alpha, \frac{C_2}{M_2 \beta_{\lambda^l}}\right). \end{aligned}$$

Then for any $\delta \in (0,1)$, for any $N \geq 0$ with probability at least $1-\delta$, we have:

$$regret(N) \leq \left(\left(\bar{E_1} + \bar{E_2}\sqrt{\log_2(2N+2)\log\left(\frac{2}{\delta}\right)}\right)(N+T_0)^{\frac{5}{6}} + 1\right)(\log(2N+2T_0+4))^2$$

$$\begin{aligned}
\textit{where} \quad \bar{E_1} &= \frac{64|\mathcal{S}|^2|\mathcal{A}|^2\left\|\frac{d_\rho^{\pi^*}}{\mu}\right\|_\infty^2}{\underline{C_\alpha}C_2(1-\gamma)^2}\left(V^*_{\mathcal{M},\gamma} + \bar{\lambda}\log\left(\frac{1}{\epsilon_{pp}}\right) + \bar{C_\alpha}^2\left\|\delta_k\right\|_{l^2}^2 + M_1\frac{\bar{\beta}}{2}\bar{C_\alpha}^2\right) + 1 \\
\bar{E_2} &= \frac{64|\mathcal{S}|^2|\mathcal{A}|^2\left\|\frac{d_\rho^{\pi^*}}{\mu}\right\|_\infty^2}{\underline{C_\alpha}C_2(1-\gamma)^2}C_1\bar{C_\alpha}\sqrt{4\left(\frac{2}{1-\gamma}+2\bar{\lambda}\right)^2 + \beta_{\bar{\lambda}}^2C_1^2\bar{C_{l,\alpha}}^2} \\
\bar{C_\alpha} &= \frac{C_2(1-\gamma)^2}{8M_2}.
\end{aligned}$$

Proof. Fix some $\delta \in (0,1)$. From Theorem 5.1, for any phase l and for any episode $k \in \{0, ..., T_l - 1\}$, with probability at least $1 - \frac{\delta}{2^{l+1}}$, we have

$$\begin{aligned}
regret_l(k) &\leq D_3^{(l)}\left(D_1^{(l)} + D_2^{(l)}\sqrt{\log\left(\frac{2^{l+1}}{\delta}\right)}\right)\frac{1}{\epsilon_l^2}\sqrt{k+1}\log(k+3) + \epsilon_l(k+1) + \log(k+1) + 1 \leq \\
&\leq D_3^{(l)}\left(D_1^{(l)} + D_2^{(l)}\sqrt{\log\left(\frac{2^{l+1}}{\delta}\right)}\right)T_l^{\frac{1}{3}}\sqrt{T_l}\log(T_l+2) + T_l^{-\frac{1}{6}}T_l + \log(T_l+2) + 1 \leq \\
&\leq \left(D_3^{(l)}\left(D_1^{(l)} + D_2^{(l)}\sqrt{\log\left(\frac{2^{l+1}}{\delta}\right)}\right) + 1\right)T_l^{\frac{5}{6}}\log(T_l+2) + \log(T_l+2) + 1
\end{aligned}$$

We are interested in an overall bound, so we wish to sum the regrets of each phase. In order to do this the first step is to bound the constants $D_1^{(l)}, D_2^{(l)}, D_3^{(l)}$ uniformly in l.

Let $x_k = \frac{1}{\sqrt{k+3}\log_2(k+3)}$ for $k \geq 0$. Then we have

$$\sum_{k=0}^{\infty} x_k^4 \leq \sum_{k=0}^{\infty} x_k^2 \leq 1$$

where the first inequality comes from the fact that $x_k \leq 1$ for any $k \geq 0$, while the second inequality can be derived by noticing that

$$\sum_{k=0}^{\infty} x_k^2 \leq \int_0^{\infty} \frac{1}{(y+2)(\log_2(y+2))^2}\mathrm{d}y = 1$$

From this it immediately follows that $\left\|(\alpha^{l,k})_{k\geq 0}\right\|_{l^2}^2 \leq C_{l,\alpha}^2$ and $\left\|(\alpha^{l,k})_{k\geq 0}\right\|_{l^4}^4 \leq C_{l,\alpha}^4$.

The fact that $\epsilon_l \leq 1$ implies that $\lambda^l \leq \bar{\lambda}$ which in turn implies

$$\frac{8}{(1-\gamma)^2} \leq \beta_{\lambda^l} \leq \beta_{\bar{\lambda}} \leq \frac{8}{(1-\gamma)^2} + \frac{2\bar{\lambda}}{|\mathcal{S}|}$$

From that we immediately get a uniform upper bound on β_{λ^l}, but also a uniform upper bound on $C_{l,\alpha}$

$$C_{l,\alpha} \leq \frac{C_2}{M_2 \beta_{\bar{\lambda}}} \leq \underbrace{\frac{C_2(1-\gamma)^2}{8M_2}}_{=:\bar{C_\alpha}}$$

while the lower bound $\underline{C_\alpha} \leq C_{l,\alpha}$ was assumed by hypothesis. Lastly, thanks to the algorithm definition in Line 9, we have that

$$-L_{\lambda^l}(\theta^{l,0} \leq -\lambda^l R(\theta^{l,0}) \leq -\lambda^l \log\left(\epsilon_{pp}\right) \leq \bar{\lambda} \log\left(\frac{1}{\epsilon_{pp}}\right)$$

We can now finally bound the constants from Theorem 5.1 for any phase $l \geq 0$

$$\begin{aligned} D_1^{(l)} &= V^*_{\mathcal{M},\gamma} - L_{\lambda^l}(\theta^0) + C_{l,\alpha}^2 \left\|\delta_k\right\|_{l^2}^2 + M_1 \frac{\beta_{\lambda^l}}{2} C_{l,\alpha}^2 \leq \\ &\leq \underbrace{V^*_{\mathcal{M},\gamma} + \bar{\lambda} \log\left(\frac{1}{\epsilon_{pp}}\right) + \bar{C_\alpha}^2 \left\|\delta_k\right\|_{l^2}^2 + M_1 \frac{\bar{\beta}}{2} \bar{C_\alpha}^2}_{=:\bar{D_1}} \end{aligned}$$

$$\begin{aligned} D_2^{(l)} &= 2C_1 C_{l,\alpha} \sqrt{4\left(\frac{2}{1-\gamma} + 2\lambda^l\right)^2 + \beta_\lambda^2 C_1^2 C_\alpha^2} \leq \\ &\leq \underbrace{2C_1 \bar{C_\alpha} \sqrt{4\left(\frac{2}{1-\gamma} + 2\bar{\lambda}\right)^2 + \beta_{\bar{\lambda}}^2 C_1^2 \bar{C_{l,\alpha}}^2}}_{=:\bar{D_2}} \end{aligned}$$

$$\begin{aligned} D_3^{(l)} &= \frac{64|\mathcal{S}|^2|\mathcal{A}|^2 \left\|\frac{d_\rho^{\pi^*}}{\mu}\right\|_\infty^2}{C_{l,\alpha} C_2 (1-\gamma)^2} \leq \\ &\leq \underbrace{\frac{64|\mathcal{S}|^2|\mathcal{A}|^2 \left\|\frac{d_\rho^{\pi^*}}{\mu}\right\|_\infty^2}{\underline{C_\alpha} C_2 (1-\gamma)^2}}_{=:\bar{D_3}} \end{aligned}$$

This bounds lead to, with probability at least $1 - \frac{\delta}{2^{l+1}}$

$$regret_l(k) \leq \left(\bar{D_3}\left(\bar{D_1} + \bar{D_2}\sqrt{\log\left(\frac{2^{l+1}}{\delta}\right)}\right) + 1\right) T_l^{\frac{5}{6}} \log(T_l + 2) + \log(T_l + 2) + 1 \tag{5.13}$$

We wish to bound the regret after N sampled trajectories, that is

$$\begin{aligned} regret(N) &= \sum_{l=0}^{l_N-1} regret_l(T_l-1) + regret_{l_N}(k_N) \leq \\ &\leq \sum_{l=0}^{l_N} regret_l(T_l-1) \end{aligned} \tag{5.14}$$

where $(l_N, k_N) = \mathbf{G}(N)$. In addition, by the choice of T_l we have that

$$\mathbf{B}(l,k) = (2^l-1)T_0 + k \geq (s^l-1)T_0$$

which implies for any $N \geq 0$ that

$$l_N \leq \log_2\left(\frac{N}{T_0}+1\right) \leq \log_2(N+1) \tag{5.15}$$

and this also implies that

$$T_{l_N} = 2^{l_N}T_0 \leq N + T_0 \tag{5.16}$$

Now combining (5.13) and (5.14), since different phases realizations are indipendent from each other, with probability at least $1 - \sum_{l=0}^{l_N} \frac{\delta}{2^{l+1}} \geq 1-\delta$, for any $N \geq 0$,

$$\begin{aligned} regret(N) &\leq \sum_{l=0}^{l_N} regret_l(T_l-1) \leq \\ &\leq \sum_{l=0}^{l_N} \left(\bar{D_3}\left(\bar{D_1} + \bar{D_2}\sqrt{\log\left(\frac{2^{l+1}}{\delta}\right)}\right)+1\right) T_l^{\frac{5}{6}} \log(T_l+2) + \log(T_l+2) + 1 \leq \\ &\leq (l_N+1)\left(\hat{R}_1(N) + \hat{R}_2(N)\right) \end{aligned} \tag{5.17}$$

where $\hat{R}_1(N)$ and $\hat{R}_2(N)$ are defined as below and bounded using (5.15) and (5.16)

$$\begin{aligned} \hat{R}_1(N) &= \left(\bar{D_3}\left(\bar{D_1} + \bar{D_2}\sqrt{\log\left(\frac{2^{l_N+1}}{\delta}\right)}\right)+1\right) T_{l_N}^{\frac{5}{6}} \log(T_{l_N}+2) \leq \\ &\leq \left(\bar{D_3}\bar{D_1} + 1 + \bar{D_3}\bar{D_2}\sqrt{(l_N+1)\log\left(\frac{2}{\delta}\right)}\right)(N+T_0)^{\frac{5}{6}} \log(N+T_0+2) \leq \\ &\leq \left(\bar{E_1} + \bar{E_2}\sqrt{\log_2(2N+2)\log\left(\frac{2}{\delta}\right)}\right)(N+T_0)^{\frac{5}{6}} \log(N+T_0+2), \end{aligned}$$

$$\begin{aligned} \hat{R}_2(N) &= \log(T_{l_N}+2) + 1 \leq \\ &\leq \log(N+T_0+2) + 1 = \\ &= \log(2N + 2T_0 + 4). \end{aligned}$$

where we set $\bar{E_1} := \bar{D_3}\bar{D_1} + 1$ and $\bar{E_2} := \bar{D_3}\bar{D_2}$. It also holds

$$\begin{aligned} l_N + 1 &\leq \log_2(N+1) + 1 = \\ &= \log_2(2N+2) \end{aligned}$$

Finally, putting all together in (5.17) , we have

$$\begin{aligned} \mathit{regret}(N) \leq \left(\bar{E_1} + \bar{E_2}\sqrt{\log_2(2N+2)\log\left(\frac{2}{\delta}\right)}\right)(N+T_0)^{\frac{5}{6}}(\log(2N+T_0+2))^2 + \\ + (\log(2N+2T_0+4))^2 \end{aligned}$$

This completes the proof. □

Remark. In Theorem 5.3, asimptotically with probability at least $1-\delta$

$$\mathit{regret}(N) = O\left(N^{\frac{5}{6}}\log\left(\frac{N}{\delta}\right)^{\frac{5}{2}}\right),$$

but in the big-O notation above we have hidden the problem dependent quantities, which can be made explicit if we specialize the results to the REINFORCE with baseline gradient estimator.

Corollary 5.4. *Assume to folllow Algorithm 2 in the setting and hypothesis of Theorem 5.3. Assume we use (4.12) as gradient estimator with $|b(s)| \leq B$ for any $s \in \mathcal{S}$ where $B > 0$ is a constant. Set $H^{l,k} \geq \frac{3}{2\min\{\beta, 1-\beta\}}\log_{1/\gamma}\left(\frac{8(k+1)}{(1-\gamma)^3}\right)$ and set $\epsilon_{pp} = \frac{1}{2|\mathcal{A}|}$, $T_0 = 1$. Then for any $N \geq 0$ with probability at least $1-\delta$, we have*

$$\mathit{regret}(N) = O\left(\frac{|\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)^2}\left\|\frac{d_\rho^{\mathcal{M},\pi^*,\gamma}}{\mu}\right\|_\infty^2 N^{\frac{5}{6}}\left(\log\left(\frac{N}{\delta}\right)\right)^{\frac{5}{2}}\right). \tag{5.18}$$

The proof can be done by simply expanding and evaluating the constants in the bounds of Theorem 5.3 combined with the specific choice of the hyper parameters as well as the constants in the Gradient Approximator Assumption.

Notice that here, with those specific choices of hyperparameters we are able to make all the problem dependent parameters explicit in the big-O notation, which is consistent with the convention of the Reinforcement Learning literature. Here the only hidden quantities are some absolute constants.

Lemma 5.5. Suppose that for every $\delta > 0$ with probability at least $1-\delta$, for every $N \in \mathbb{N}$ we have

$$\mathit{regret}(N) \leq d_1(N+d_2)^{d_3}\left(\log\left(\frac{N}{\delta}\right)\right)^{d_4}$$

for some constants $d_1, d_3, d_4 \geq 0$ and $d_2 \in [0,1)$. Then we have

$$\lim_{N\to\infty} \frac{regret(N)}{N} = 0 \quad \text{almost surely.}$$

Proof. Let $\delta_N := 1/N^2$ and define the events $\{\bar{A}_N\}_{N\geq 1}$ as

$$\bar{A}_N := \left\{ regret(N) > d_1(N+d_2)^{d_3} \left(\log\left(\frac{N}{\delta_N}\right)\right)^{d_4} \right\}.$$

Then $\mathbb{P}(\bar{A}_N) \leq \delta_N$ for hypothesis, and hence $\sum_{N=1}^{\infty} \mathbb{P}(\bar{A}_N) \leq \sum_{N=1}^{\infty} 1/N^2 < \infty$. Hence by Borel-Cantelli lemma, we have

$$\mathbb{P}(\bar{A}_N \text{ occurs infinitely often}) = 0.$$

Finally, by noticing that the complement of $\bar{A}_N$ is a subset of A_N, the proof is complete. □

It is also worth noting that all this results extend quite straightforward to the minibatch case, in which the gradient approximation at each (l,k) is made over M different trajectories instead of 1. This idea improves the convergence rate with respect to the number of gradient steps, but it also multiplies the number of calls to the sampling model by a factor of M. This trade-off is better shown by the following result.

Corollary 5.6. *Chose a minibatch size $M \geq 1$. Assume to follow Algorithm 2 in the setting and hypothesis of previous Corollary, but using (4.13) in place of (4.12) as gradient estimator. Then we have that*

$$regret(N;M) = O\left(\frac{|\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)^2} \left\| \frac{d_\rho^{\mathcal{M},\pi^*,\gamma}}{\mu} \right\|_\infty^2 (M^{\frac{1}{6}} + M^{-\frac{5}{6}})(N+M)^{\frac{5}{6}} \left(\log\left(\frac{N}{\delta}\right)\right)^{\frac{5}{2}} \right),$$

where N refers to the effective calls to the sampling model. In particular, when $M = 1$, the bound above reduces to (5.18).

The proof is tedious and not useful for the next results. It can be found in [10].

We can see explicitly that there might be a trade-off between the terms $M^{\frac{1}{6}}$ and $M^{-\frac{5}{6}}$. The intuition behind this is a trade-off between lower variance with larger batch size and more frequent updates with smaller batch size.

Remark. The results of this Section depend on the *distribution mismatch coefficient* $\left\| \frac{d_\rho^{\mathcal{M},\pi^*,\gamma}}{\mu} \right\|_\infty$, where ρ is the arbitrary distribution with respect to which we measure the distance from the optimal value function. If, for example, μ is uniform on the states, the coefficient is bounded from above by $\frac{1}{|\mathcal{S}|}$. If μ is not uniform but everywhere positive, *i.e.* with coverage hypothesis, this coefficient is unbounded but at least finite and constant once the MDP is fixed. The problems with all previous results arise if μ is 0 on some states and, more generally, when μ is supported on only one initial state s_0. With this sampling model assumption, in fact, the distribution mismatch coefficient is infinite and so previous convergence results do not hold anymore.

5.2 Simulating coverage hypothesis

Given a MDP, fix a state $s_0 \in \mathcal{S}$. We assume throughout all this Section to only have access to a s_0-reset model.

In order to solve the exploration problem, we develop a strategy to produce an *explorative policy* that, starting from s_0, visits all the other states with non-zero probability. This requirement alone is not hard, the difficulties arise in trying to bound the mismatch coefficient with known (or at least fixed) quantities, that is our final aim in order to preserve good convergence guarantees.

We basically simulate a new (hopefully better) sampling model using the existing one, then we apply the previously analyzed algorithm with the new sampling model and find the optimal solution. These two stages may be made together, in the same spirit as policy evaluation and improvement in generalized policy iteration, but this is beyond the scope of this Thesis and may be a subject for future work.

$$s_0\text{-reset model} \quad \xrightarrow{\texttt{simulation}} \quad \mu\text{-reset model} \quad \xrightarrow{\texttt{optimization}} \quad \pi^*$$

We start with a simple observation: we call a state $s \in \mathcal{S}$ *visitable* if there exists a trajectory τ from s_0 to s with all transition probabilities $\mathcal{P}(s_{t+1}|s_t, a_t) > 0$ for all t. If a state is not visitable even the optimal policy will never reach it, and so $d_{s_0}^{*;\mathcal{M}}(s) = 0$.

Fact. Given an MDP $\mathcal{M}$ and an initial states s_0 we can assume without loss of generality that there are no non-visitable states.

Proof. If there are some, we can simply consider a new MDP $\mathcal{M}' := (\mathcal{S}', \mathcal{A}, \mathcal{P}', r)$ where $\mathcal{S}' \subset \mathcal{S}$ is the set of visitable states and $\mathcal{P}' := \mathcal{P}|_{\mathcal{S}'}$ is the restriction to those states. Optimal policy in the reduced MDP is exactly the same as the original one because if a state is not visitable it also can not lead to any reward. □

With this Fact we may start considering the uniform policy π^u, such that $\pi(a|s) = \frac{1}{|\mathcal{A}|}$ for every $s, a \in \mathcal{S} \times \mathcal{A}$. With the all-states-visitability assumption this policy, as all policies with strictly positive probabilities in all state action pairs, has a positive state visitation distribution $d_{s_0}^{\mathcal{M},\pi^u}(s) > 0$ for all $s \in \mathcal{S}$.

Proof. Fix a state $s \in \mathcal{S}$. From its visitability there exists a trajectory τ_s from s_0 to s with all positive transition probabilities.

$$\begin{aligned}
d_{s_0}^{\mathcal{M},\pi^u}(s) = (1-\gamma) \sum_{H=0}^{\infty} \gamma^H d_{s_0,H+1}^{\mathcal{M},\pi^u}(s) = \\
&= (1-\gamma) \sum_{H=0}^{\infty} \gamma^H \sum_{\tau \in T_H} \mathbb{1}_{\{s_{H-1}=s\}} \mathbb{P}(\tau | \pi^u, \mathcal{M}, s_0) \geq \\
&\geq (1-\gamma)\gamma^{\text{len}(\tau_s)-1}\, \mathbb{P}(\tau_s | \pi^u, \mathcal{M}, s_0) = \\
&= (1-\gamma)\gamma^{\text{len}(\tau_s)-1}\, \pi^u(a_0|s_0) \prod_{t=1}^{\text{len}(\tau_s)-1} \mathcal{P}(s_t|s_{t-1}, a_{t-1}) \pi^u(a_t|s_t) = \\
&= (1-\gamma)\gamma^{\text{len}(\tau_s)-1}\, \frac{1}{|\mathcal{A}|^{\text{len}(\tau_s)}} \prod_{t=1}^{\text{len}(\tau_s)-1} \mathcal{P}(s_t|s_{t-1}, a_{t-1}) > 0.
\end{aligned}$$

□

We now need to make two key observations:

- Given a policy, we can simulate a sampling model with the policy's visitation distribution as initial distribution;
- Given two or more policies, we can simulate the mean of their distribution.

The first one is made with a simple stochastic stopping algorithm:

Algorithm 3 Visit(π)

```
Launch the true sampling model. Set $s = s_0$.
while true do
    With probability $1 - \gamma$:
        break
    Choice action $a$ according to $\pi(\cdot|s)$.
    Perform $a$ in $\mathcal{M}$ and go to state $s'$ according to the unknown $\mathcal{P}(\cdot|s, a)$.
    Set $s = s'$.
end while
return $s$
```

The second one is made with a policy *ensemble*.

Definition 5.7. Given a set of policies $\{\pi^0, \pi^1, ..., \pi^n\}$, we can define their **ensemble**

$$\mathit{ensemble}(\pi^0, \pi^1, ..., \pi^n)$$

as the agent behaviour that at the start of each new trajectory uniformly select one policy in the set, and then follows it until the end of the trajectory.

By linearity argument it is easy to see that the state visitation distribution of the ensemble is the mean of the state visitation distributions of the policies, that is

$$d_{s_0}^{\mathcal{M},ensemble(\pi^0,\pi^1,...,\pi^n),\gamma}(s) = \frac{1}{n+1}\sum_{i=0}^{n} d_{s_0}^{\mathcal{M},\pi^i,\gamma}(s). \tag{5.19}$$

We are finally ready to define our exploration policy discovery algorithm:

Algorithm 4 Iterative Exploration Strategy

```
Input: .
Set $\pi^0$ the uniform policy such that $\pi^0(a|s) = \frac{1}{|\mathcal{A}|}$ for each $s, a \in \mathcal{S} \times \mathcal{A}$. Set $\mu^0 = \mu$
for step $n = 0, 1, 2, ...$ do
    Define $\tilde{\pi}^n := ensemble(\pi^0, \pi^1, ..., \pi^n)$.
    Set $K^n = \left\{ s \in \mathcal{S} | \hat{d}^{\tilde{\pi}^n} \leq \beta_n \right\}$.
    Set $r^n(s,a) = \mathbb{1}_{\{s \in K^n\}}$ and let $\mathcal{M}^n = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r^n)$.
    Simulate $\mu^{n+1} := d_{s_0}^{ensemble(s_0,\tilde{\pi}^n)}$ as initial distribution using Visit($\tilde{\pi}^n$).
    Find $\pi^{n+1}$ an $\epsilon$-optimal solution in $\mathcal{M}^n$ with simulated $\mu^{n+1}$-reset model using Alg. 1.
end for
```

Single step analysis

We now consider in deeper details a single step in the algorithm. In order to do this we fix a step $N > 0$ through all this section.

The proposed explorative strategy is $\tilde{\pi}^N := ensemble(\pi^0, \pi^1, ..., \pi^N)$ and so we hope that his state visitation distribution $d_{s_0}^{\mathcal{M},\tilde{\pi}^N,\gamma}(s)$ in the original MDP $\mathcal{M}$ with the real sampling model $\mu \sim s_0$ is positive for every state $s \in \mathcal{S}$, more than that we hope that is bounded from below by a fixed quantity. Having said that, it is quite natural to define a set of *poorly visited states.*

$$\mathcal{K}^N := \left\{ s \in \mathcal{S} | d_{\mu^N}^{\mathcal{M},\tilde{\pi}^N,\gamma}(s) < \beta_N \right\}. \tag{5.20}$$

In real usage we do not have access to the true state visitation distribution and we need to use an approximator $\hat{d}$ that will lead to an approximate poorly visited states set $\hat{\mathcal{K}}^N$, but for clarity of presentation we deal with this problem at the end.

The states in $\mathcal{K}^N$ are the states we visit less frequently and thus the ones which leads to poor convergence guarantees in Algorithm 2. Our aim is then to improve visitation probability in these states, it is again quite natural to set positive rewards in these states, independently on the selected action, that is

$$r^N(s,a) := \mathbb{1}_{\{s \in \mathcal{K}^N\}}. \tag{5.21}$$

We then consider an (only theoretically defined) MDP to make our analysis:

$$\mathcal{M}^N := (\mathcal{S}, \mathcal{A}, \mathcal{P}, r^N).$$

Notice that the rewards change while the dynamics remains the same, this mean that we have different value functions but same policy spaces ($\Pi_{\mathcal{M}} = \Pi_{\mathcal{M}^N}$) and same visitation distributions.

We also make another usage, independent of the previous one with rewards definition, of the current explorative policy $\tilde{\pi}^N$: we set $\mu^N := d_{s_0}^{ensemble(s_0, \tilde{\pi}^{N-1})}$ and simulate a μ^N-sampling model from the real sampling model using the Visit Algorithm. In the definition of μ^N we clearly abuse the notation since s_0 is not a policy, what we mean is that half of the times the Visit follows $\tilde{\pi}^N$ while the other half it simply stays in s_0, that is always the initial state of the real sampling model.

Using (4.2) the value function in $\mathcal{M}^N$ is

$$\begin{aligned} V_{\mathcal{M}^N, \pi, \gamma}(\mu^N) &= \sum_{s \in \mathcal{S}, a \in \mathcal{A}} d_{\mu^N}^{\mathcal{M}^N, \pi, \gamma}(s) \pi(a|s) r^N(s, a) = \\ &= \sum_{s \in \mathcal{K}^N} d_{\mu^N}^{\mathcal{M}^N, \pi, \gamma}(s) \left(\sum_{a \in \mathcal{A}} \pi(a|s) \right) = \\ &= \sum_{s \in \mathcal{K}^N} d_{\mu^N}^{\mathcal{M}^N, \pi, \gamma}(s). \end{aligned}$$

We then perform Algorithm 1 until a certain stopping criterion is satisfied and define π^{N+1} as the almost optimal policy found by the algorithm, this policy satisfies

$$\sum_{s \in \mathcal{K}^N} d_{\mu^N}^{\mathcal{M}^N, \pi^{N+1}, \gamma}(s) = V_{\mathcal{M}^N, \pi^{N+1}, \gamma}(\mu^N) \geq V^*_{\mathcal{M}^N, \gamma}(\mu^N) - err(N), \tag{5.22}$$

where $err(N)$ is the error made in step N, stated precisely with all the dependence in (5.31).

The optimal value function, by definition, for every $\pi \in \Pi_{\mathcal{M}} (= \Pi_{\mathcal{M}^N})$ satisfies

$$\begin{aligned} V^*_{\mathcal{M}^N, \gamma}(\mu^N) &\geq V_{\mathcal{M}^N, \pi, \gamma}(\mu^N) = \\ &= \frac{1}{2} V_{\mathcal{M}^N, \pi, \gamma}(s_0) + \frac{1}{2} V_{\mathcal{M}^N, \pi, \gamma}(d_{s_0}^{\mathcal{M}, \tilde{\pi}^{N-1}, \gamma}) \geq \\ &\geq \frac{1}{2} V_{\mathcal{M}^N, \pi, \gamma}(s_0). \end{aligned} \tag{5.23}$$

Combining (5.22) and (5.23) leads to

$$\begin{aligned} \sum_{s \in \mathcal{K}^N} d_{\mu^N}^{\mathcal{M}^N, \pi^{N+1}, \gamma}(s) + err(N) &\geq \frac{1}{2} \max_{\pi \in \Pi_{\mathcal{M}^N}} \sum_{s \in \mathcal{K}^N} d_{s_0}^{\mathcal{M}^N, \pi, \gamma}(s) = \\ &= \frac{1}{2} \max_{\pi \in \Pi_{\mathcal{M}}} \sum_{s \in \mathcal{K}^N} d_{s_0}^{\mathcal{M}, \pi, \gamma}(s). \end{aligned} \tag{5.24}$$

We can now consider the optimal explorative policy π^E from Section 3.3, such that $d_{s_0}^{\mathcal{M},\pi^E,\gamma}(s) \geq \tilde{d}$ for any $s \in \mathcal{S}$, and use it in the max in (5.24). This leads to

$$\sum_{s \in \mathcal{K}^N} d_{\mu^N}^{\mathcal{M}^N,\pi^{N+1},\gamma}(s) + err(N) \geq \tilde{d}|\mathcal{K}^N|. \tag{5.25}$$

It is also worth noting that, since distribution are non-negative, (5.24) also leads to

$$\sum_{s \in \mathcal{K}^N} d_{\mu^N}^{\mathcal{M}^N,\pi^{N+1},\gamma}(s) + err(N) \geq \frac{1}{2} \max_{\pi \in \Pi_{\mathcal{M}}} \sum_{s \in Q} d_{s_0}^{\mathcal{M},\pi,\gamma}(s) \qquad \forall Q \subseteq \mathcal{K}^N. \tag{5.26}$$

Overall analysis

For the overall analysis we use a double counting trick, over states and over algorithm's steps, of the state visitation distributions of our explorative policies. We fix a final step $N > 0$ and analyze steps $n = 0, ..., N$.

We temporarily fix a state $s \in \mathcal{S}$.

$$\sum_{n=0}^{N} \mathbb{1}_{\{s \in \mathcal{K}^n\}} d_{\mu^n}^{\mathcal{M}^n,\pi^{n+1},\gamma}(s) \leq \begin{cases} 0 & \text{if } s \notin \mathcal{K}^n \, \forall n \leq N \\ \displaystyle\sum_{n=0}^{\tilde{n}} d_{\mu^n}^{\mathcal{M}^n,\pi^n,\gamma}(s) + d_{\mu^n}^{\mathcal{M}^n,\pi^{\tilde{n}+1},\gamma}(s) & \text{if } \tilde{n} := \max_{n \leq N}\{n | s \in \mathcal{K}^n\} \end{cases}$$

and by definition ok $\mathcal{K}^n$ and (5.19) we get

$$\begin{cases} 0 & \text{if } s \notin \mathcal{K}^n \, \forall n \leq N \\ \displaystyle\sum_{n=0}^{\tilde{n}} d_{\mu^n}^{\mathcal{M}^n,\pi^n,\gamma}(s) + d_{\mu^n}^{\mathcal{M}^n,\pi^{\tilde{n}+1},\gamma}(s) & \text{if } \tilde{n} := \max_{n \leq N}\{n | s \in \mathcal{K}^n\} \end{cases} \leq \begin{cases} 0 \\ \beta_{\tilde{n}}\tilde{n} + d_{\mu^n}^{\mathcal{M}^n,\pi^{\tilde{n}+1},\gamma}(s) \end{cases}.$$

By setting $\beta_n := \frac{\beta}{n}$ for some constant $\beta \in \mathbb{R}$, we have that $\beta_{\tilde{n}}\tilde{n} = \beta$ for any $\tilde{n}$.
Moreover, it is clear that $d_{\mu^n}^{\mathcal{M}^n,\pi^{\tilde{n}+1},\gamma}(s) = d_{\mu^n}^{\mathcal{M},\pi^{\tilde{n}+1},\gamma}(s)$ since $\mathcal{M}^n$ and $\mathcal{M}$ differ only on rewards and thus on value functions, not on transitions probabilities and thus on the states visitation distributions.

Lemma 5.8. Let μ be a distribution over states that is equal to the states visitation distribution of an ensemble of policies all starting from s_0. Then for any $\pi' \in \Pi_{\mathcal{M}}$, it holds

$$d_{\mu}^{\mathcal{M},\pi',\gamma}(s) \leq \max_{\pi \in \Pi_{\mathcal{M}}} d_{s_0}^{\mathcal{M},\pi,\gamma}(s).$$

Proof. The proof relies on a similar idea to the Policy Improvement Theorem 2.1, that is a local maggioration everywhere implies a global maggioration. □

With this, in any case, we have that

$$\sum_{n=0}^{N} \mathbb{1}_{\{s\in\mathcal{K}^n\}} d_{\mu^n}^{\mathcal{M}^n,\pi^{n+1},\gamma}(s) \leq \beta + \max_{\pi\in\Pi_{\mathcal{M}}} d_{s_0}^{\mathcal{M},\pi,\gamma}(s), \tag{5.27}$$

where the RHS is independent on N.

Now summing (5.27) over $s \in \mathcal{S}$ leads to

$$\sum_{s\in\mathcal{S}}\sum_{n=0}^{N} \mathbb{1}_{\{s\in\mathcal{K}^n\}} d_{\mu^n}^{\mathcal{M}^n,\pi^{n+1},\gamma}(s) \leq \beta|\mathcal{S}| + \sum_{s\in\mathcal{S}} \max_{\pi\in\Pi_{\mathcal{M}}} d_{s_0}^{\mathcal{M},\pi,\gamma}(s), \tag{5.28}$$

while summing (5.26) over $n \in \{0, ..., N-1\}$ leads to

$$\begin{aligned}\sum_{n=0}^{N-1}\sum_{s\in\mathcal{S}} \mathbb{1}_{\{s\in\mathcal{K}^n\}} d_{\mu^n}^{\mathcal{M}^n,\pi^{n+1},\gamma}(s) &= \sum_{n=0}^{N-1}\sum_{s\in\mathcal{K}^n} d_{\mu^n}^{\mathcal{M}^n,\pi^{n+1},\gamma}(s) \geq \\ &\geq -\sum_{n=0}^{N-1} err(n) + \frac{1}{2}\sum_{n=0}^{N-1} \max_{\pi\in\Pi_{\mathcal{M}}} \sum_{s\in Q^n} d_{s_0}^{\mathcal{M},\pi,\gamma}(s) \qquad \forall Q^n \subseteq K^n.\end{aligned} \tag{5.29}$$

The first term of (5.28) and (5.29) is clearly the same, so we deduce that if $Q^n \subseteq \mathcal{K}^n$ $\forall n$ then

$$\begin{aligned}\frac{1}{2}\sum_{n=0}^{N-1} \max_{\pi\in\Pi_{\mathcal{M}}} \sum_{s\in Q^n} d_{s_0}^{\mathcal{M},\pi,\gamma}(s) &\leq \sum_{n=0}^{N-1} err(n) + \beta|\mathcal{S}| + \sum_{s\in\mathcal{S}} \max_{\pi\in\Pi_{\mathcal{M}}} d_{s_0}^{\mathcal{M},\pi,\gamma}(s) \leq \\ &\leq \|err\|_{l^1} + \beta|\mathcal{S}| + \sum_{s\in\mathcal{S}} \max_{\pi\in\Pi_{\mathcal{M}}} d_{s_0}^{\mathcal{M},\pi,\gamma}(s).\end{aligned}$$

With our specific choice of β^n, the sets $\mathcal{K}^n$ are monotonically decreasing, *i.e.*, $\mathcal{K}^{n+1} \subseteq \mathcal{K}^n$ for any n. We can than set $Q^n := \mathcal{K}^N$ for any $n \in \{0, ..., N-1\}$ and this leads to

$$\max_{\pi\in\Pi_{\mathcal{M}}} \sum_{s\in\mathcal{K}^N} d_{s_0}^{\mathcal{M},\pi,\gamma}(s) \leq \frac{2}{N}\left(\|err\|_{l^1} + \beta|\mathcal{S}| + \sum_{s\in\mathcal{S}} \max_{\pi\in\Pi_{\mathcal{M}}} d_{s_0}^{\mathcal{M},\pi,\gamma}(s)\right). \tag{5.30}$$

This bound is crucial in the final result.

If, instead, we use (5.25) in place of (5.26) in evaluating (5.29) we have that

$$\sum_{n=0}^{N-1}\sum_{s\in\mathcal{S}} \mathbb{1}_{\{s\in\mathcal{K}^n\}} d_{\mu^n}^{\mathcal{M}^n,\pi^{n+1},\gamma}(s) \geq -\|err\|_{l^1} + \sum_{n=0}^{N-1} \tilde{d}|\mathcal{K}^n|,$$

which, with the same double counting argument, leads to

$$\sum_{n=0}^{N-1} |\mathcal{K}^n| \leq \frac{1}{\tilde{d}}\left(\|err\|_{l^1} + \beta|\mathcal{S}| + \sum_{s\in\mathcal{S}} \max_{\pi\in\Pi_{\mathcal{M}}} d_{s_0}^{\mathcal{M},\pi,\gamma}(s)\right),$$

and this bound the sum of cardinalities of $\mathcal{K}^n$ up to N with a constant. This immediately implies that $\lim_{n\to\infty} |\mathcal{K}^n| = 0$. Moreover, here we do not need the monotonicity of the sets $\mathcal{K}^n$ and so other, weaker, choices of β_n can be made without loosing this result.

Remark. We need to be more precise about $err(N)$, related to (1.29), we use $err(N)$ in

$$V_{\mathcal{M}^N,\pi^{N+1},\gamma}(\mu^N) \geq V^*_{\mathcal{M}^N,\gamma}(\mu^N) - err(N),$$

where the sampling model is the simulated μ^N-reset model. Then, more explicitely

$$err(N) = err_{\mathcal{M}^N,\gamma,\mu^N}(\pi^{N+1};\mu^N). \tag{5.31}$$

5.2.1 Without coverage hypothesis

Finally, we can use this two previously analyzed techniques, namely Algorithm 1 and Algorithm 4, all together. We propose the following Algorithm 5 in pseudocode, then we analyze it in deeper detail.

This algorithm mainly splits in two parts:

- lines 2-17: aim at finding an explorative policy in order to simulate a sampling model;
- lines 19-25: aim at finding an optimal policy for the original task $\mathcal{M}$.

Theorem 5.9. *Given an MDP $\mathcal{M}$, a discount factor $\gamma \in [0,1)$ and a s_0-reset sampling model for some $s_0 \in \mathcal{S}$. Assume to follow Algorithm 5 and assume to use (4.12) as gradient estimator with $|b(s)| \leq B$ for any $s \in \mathcal{S}$ where $B > 0$ is a constant. Assume choosing H^k, α^k and λ as hypothesis of Corollary of Theorem 5.1.*
Setting $K_n := n^{16}$ for every $n \in \mathbb{N}$ and $N = K^{\frac{1}{18}}$. Then, for any $\delta \in (0,1)$, for any $K \in \mathbb{N}$ with probability at least $1-\delta$, we have:

$$\frac{1}{K}\sum_{k=0}^{K-1} err_{\mathcal{M},\gamma,s_0}(\pi_{\theta^k};\mu^N) = O\left(\frac{|\mathcal{S}|^2|\mathcal{A}|^2}{(1-\gamma)^2} K^{-\frac{1}{18}} \log(K)\sqrt{\log\frac{1}{\delta}}\right)$$

using $K + \sum_{n=0}^{N-1} K_n = O(K + K^{\frac{17}{18}}) = O(K)$ calls to the s_0-reset sampling model.

Proof. The first part of the algorithm finds an explorative policy $\tilde{\pi}^N$ such that, thanks to (5.30) we have

$$\sum_{s\in\mathcal{K}^N} d_{s_0}^{\mathcal{M},\pi^*,\gamma}(s) \leq \max_{\pi\in\Pi_\mathcal{M}} \sum_{s\in\mathcal{K}^N} d_{s_0}^{\mathcal{M},\pi,\gamma}(s) \leq \frac{2}{N}\left(\|err\|_{l^1} + \beta|\mathcal{S}| + \sum_{s\in\mathcal{S}} \max_{\pi\in\Pi_\mathcal{M}} d_{s_0}^{\mathcal{M},\pi,\gamma}(s)\right). \tag{5.32}$$

where $\mathcal{K}^N = \left\{ s \in \mathcal{S} | d_{s_0}^{\mathcal{M},\tilde{\pi}^N,\gamma}(s) < \beta_N \right\}$.

Focus in more detail on the $\|err\|_{l^1}$ term, it is

$$\|\{err(n)\}_{n\in\mathbb{N}}\|_{l^1} = \left\|\{err_{\mathcal{M}^n,\gamma,\mu^n}(\pi^{n+1};\mu^n)\}_{n\in\mathbb{N}}\right\|_{l^1}$$

and it clearly depends on how many episode K_n we spend in the n-th step. Recall that we define $\pi^{n+1} := ensemble(\pi_{\theta^{n,0}}, \pi_{\theta^{n,1}}, ..., \pi_{\theta^{n,K_n-1}})$ and that the value function of an ensemble is the average of the value functions of the policies is made by, this follows from (5.19) and (4.2) combined. From Corollary of Theorem 5.1 we have that for every $n \in \mathbb{N}$

$$\begin{aligned} err_{\mathcal{M}^n,\gamma,\mu^n}(\pi^{n+1};\mu^n) &= \frac{1}{K_n}\left(\sum_{k=0}^{K_n-1} err_{\mathcal{M},\gamma,\mu^n}(\pi_{\theta^{n,k}};\mu^n)\right) \leq \\ &\leq C\left\|\frac{1}{\mu^n}\right\|_\infty^2 \left(1+\sqrt{\log\frac{1}{\delta_n}}\right) K_n^{-\frac{1}{6}}\log(K_n) \end{aligned}$$

with probability at least $1-\delta_n$. If we set $\delta_n := \frac{\delta}{2^{n+2}}$ we immediately get

$$err_{\mathcal{M}^n,\gamma,\mu^n}(\pi^{n+1};\mu^n) \leq C\left\|\frac{1}{\mu^n}\right\|_\infty^2 \left(1+\sqrt{n}\sqrt{\log\frac{4}{\delta}}\right) K_n^{-\frac{1}{6}}\log(K_n) \tag{5.33}$$

Then with probaility at least $1-\sum_{n=0}^{N-1}\delta_n \geq 1-\frac{\delta}{2}$ inequality (5.33) holds for every $n \in \mathbb{N}$ at the same time, and thus the sum holds

$$\begin{aligned} \left\|\{err_{\mathcal{M}^n,\gamma,\mu^n}(\pi^{n+1};\mu^n)\}_{n\in\mathbb{N}}\right\|_{l^1} &= \sum_{n\in\mathbb{N}} err_{\mathcal{M}^n,\gamma,\mu^n}(\pi^{n+1};\mu^n) \leq \\ &\leq \sum_{n\in\mathbb{N}} C\left\|\frac{1}{\mu^n}\right\|_\infty^2 \left(1+\sqrt{n}\sqrt{\log\frac{2}{\delta}}\right) K_n^{-\frac{1}{6}}\log(K_n) \end{aligned} \tag{5.34}$$

with probability at least $1-\frac{\delta}{2}$.

Lemma 5.10. For every $n \geq 0$ it holds

$$\left\|\frac{1}{\mu^n}\right\|_\infty \leq n\left\|\frac{1}{\mu^0}\right\|_\infty$$

.

Proof. Recall that $\mu^n := d_{s_0}^{ensemble(s_0,\tilde{\pi}^n)}$ and $\tilde{\pi}^n := ensemble(\pi^0, \pi^1, ..., \pi^n)$. The inequality follows easily thanks to the linearity of the state visitation distribution and the definition of the ensemble. □

Remark. In real usage case we actually expect $\left\|\frac{1}{\mu^n}\right\|_\infty \leq \left\|\frac{1}{\mu^{n-1}}\right\|_\infty$ for every $n \geq 0$ because at the n-step we are rewarding the states where μ^{n-1} is lower, so we expect π^n to improve visitation on those states.

Then, setting $K_n := n^{16}$, we have that the sum of the exponents of n in (5.34) is equal to $1 + \frac{1}{2} + \frac{16}{6} = -\frac{7}{6}$ and thus $\|\{err(n)\}_{n\in\mathbb{N}}\|_{l^1} < \infty$.

Now the second part of the algorithm, with the ensemble $\tilde{\pi}^N$, simulate a $\mu^N := d_{s_0}^{\mathcal{M},\tilde{\pi}^N,\gamma}$ reset sampling model and thus the objective function (4.1) of which we estimate the gradient during the search will be $L_\lambda(\theta) := V_{\mathcal{M},\pi_\theta,\gamma}(\mu^N) + \lambda R(\theta)$.

For every π_{θ^k} for $k \in \{0, ..., K-1\}$ we can write

$$\begin{aligned} err_{\mathcal{M},\gamma,s_0}(\pi_{\theta^k};\mu^N) &= V^*_{\mathcal{M},\gamma}(s_0) - V_{\mathcal{M},\pi_{\theta^k},\gamma}(s_0) = \\ &= \mathbb{E}_{s,a\sim d_{s_0}^{\pi^*}}\left[A_{\mathcal{M},\pi_{\theta^k},\gamma}(s,a)\right] = \\ &= \underbrace{\mathbb{E}_{s,a\sim d_{s_0}^{\pi^*}}\left[A_{\mathcal{M},\pi_{\theta^k},\gamma}(s,a)\mathbb{1}_{\{s\notin\mathcal{K}^N\}}\right]}_{(1)} + \underbrace{\mathbb{E}_{s,a\sim d_{s_0}^{\pi^*}}\left[A_{\mathcal{M},\pi_{\theta^k},\gamma}(s,a)\mathbb{1}_{\{s\in\mathcal{K}^N\}}\right]}_{(2)} \end{aligned}$$

Averaging (1) **over episode** $k \in \{0, ..., K-1\}$**:**

By Corollary Alg 1 and Gradient domination Theorem (slightly modified) we have that with probability at least $1 - \frac{\delta}{2}$

$$\begin{aligned} \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}_{s,a\sim d_{s_0}^{\pi^*}}\left[A_{\mathcal{M},\pi_{\theta^k},\gamma}(s,a)\mathbb{1}_{\{s\notin\mathcal{K}^N\}}\right] &\leq C\left\|\left.\frac{d_{s_0}^{\mathcal{M},\pi^*,\gamma}}{\mu^N}\right|_{s\notin\mathcal{K}^N}\right\|_\infty^2\left(1+\sqrt{\log\frac{2}{\delta}}\right)K^{-\frac{1}{6}}\log(K) \leq \\ &\leq C\left(\max_{s\notin\mathcal{K}^N}\frac{1}{\mu^N(s)}\right)^2\left(1+\sqrt{\log\frac{2}{\delta}}\right)K^{-\frac{1}{6}}\log(K) \leq \\ &\leq C\left(\frac{1}{\beta_N}\right)^2\left(1+\sqrt{\log\frac{2}{\delta}}\right)K^{-\frac{1}{6}}\log(K) \leq \\ &\leq \frac{C}{\beta^2}\left(1+\sqrt{\log\frac{2}{\delta}}\right)K^{-\frac{1}{6}}\log(K)N^2, \end{aligned}$$

where we used that $\mu^N(s) \geq \beta_N$ for every $s \notin \mathcal{K}^N$ by definition of $\mathcal{K}^N$.

Averaging (2) **over episode** $k \in \{0, ..., K-1\}$**:**

$$\begin{aligned}\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}_{s,a\sim d_{s_0}^{\pi^*}}\left[A_{\mathcal{M},\pi_{\theta^k},\gamma}(s,a)\mathbb{1}_{\{s\in\mathcal{K}^N\}}\right] &= \frac{1}{K}\sum_{k=0}^{K-1}\sum_{s\in\mathcal{K}^N}\sum_{a\in\mathcal{A}} d_{s_0}^{\mathcal{M},\pi^*,\gamma}(s)\pi^*(a|s)A_{\mathcal{M},\pi_{\theta^k},\gamma}(s,a) \leq \\ &\leq \frac{1}{K}\sum_{k=0}^{K-1}\sum_{s\in\mathcal{K}^N}\sum_{a\in\mathcal{A}} d_{s_0}^{\mathcal{M},\pi^*,\gamma}(s)\pi^*(a|s)\,2 = \\ &= 2\sum_{s\in\mathcal{K}^N}\sum_{a\in\mathcal{A}} d_{s_0}^{\mathcal{M},\pi^*,\gamma}(s)\pi^*(a|s) = \\ &= 2\sum_{s\in\mathcal{K}^N} d_{s_0}^{\mathcal{M},\pi^*,\gamma}(s)\sum_{a\in\mathcal{A}}\pi^*(a|s) = 2\sum_{s\in\mathcal{K}^N} d_{s_0}^{\mathcal{M},\pi^*,\gamma}(s),\end{aligned}$$

and then by property (5.32) of $\mathcal{K}^N$ we have

$$\sum_{s\in\mathcal{K}^N} d^{\pi^*}(s) \leq \frac{2}{N}\left(\left\|\{err_{\mathcal{M},\gamma,\mu^n}(\pi^{n+1};\mu^n)\}_{n\in\mathbb{N}}\right\|_{l^1} + \beta|\mathcal{S}| + \sum_{s\in\mathcal{S}}\max_{\pi\in\Pi_{\mathcal{M}}} d_{s_0}^{\mathcal{M},\pi,\gamma}(s)\right)$$

Combining all together, we get with probability at least $1-\delta$

$$\begin{aligned}&\frac{1}{K}\sum_{k=0}^{K-1} err_{\mathcal{M},\gamma,s_0}(\pi_{\theta^k};\mu^N) = \\ &\quad =\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}_{s,a\sim d_{s_0}^{\pi^*}}\left[A_{\mathcal{M},\pi_{\theta^k},\gamma}(s,a)\mathbb{1}_{\{s\notin\mathcal{K}^N\}}\right] + \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}_{s,a\sim d_{s_0}^{\pi^*}}\left[A_{\mathcal{M},\pi_{\theta^k},\gamma}(s,a)\mathbb{1}_{\{s\in\mathcal{K}^N\}}\right] \leq \\ &\quad \leq\frac{C}{\beta^2}\left(1+\sqrt{\log\frac{2}{\delta}}\right)K^{-\frac{1}{6}}\log(K)N^2 + \frac{4}{N}\left(\|\{err(n)\}_{n\in\mathbb{N}}\|_{l^1} + \beta|\mathcal{S}| + \sum_{s\in\mathcal{S}}\max_{\pi\in\Pi_{\mathcal{M}}} d_{s_0}^{\mathcal{M},\pi,\gamma}(s)\right).\end{aligned}$$

By setting $N := K^{-\frac{1}{18}}$ the proof is complete. The problem dependent parameter $|\mathcal{S}|$, $|\mathcal{A}|$ and γ are involved exactly as in Theorem 5.3, where the inequality we used comes from. □

Remark. We still have to deal with the approximation $\hat{\mathcal{K}}^n$ of $\mathcal{K}^n$. The approximation is done by estimating the state visitation distribution $d_{\mu^n}^{\mathcal{M},\tilde{\pi}^n,\gamma}(s)$ empirically for every states, that is, run a certain number D_n of episodes and count occurrences. Notice that we can set this number up to $D_n \sim n^{16}$ without influencing asymptotic convergence rate. The idea is to restate (5.30) with an added n-dependent stochastic error. A more rigorous quantification of this effect should be carried on and it may be a subject for future work.

Algorithm 5 Policy Gradient Method with exploration

```
Input: regularization parameter $\lambda$, step-sizes $\alpha^k$, number of optimization episodes $K$, number of steps $N$, step length $K_n$.
Set $\pi^0$ the uniform policy such that $\pi^0(a|s) = \frac{1}{|\mathcal{A}|}$ for each $s, a \in \mathcal{S} \times \mathcal{A}$.
for step $n = 0, 1, 2, ..., N-1$ do
    Define $\tilde{\pi}^n := ensemble(\pi^0, \pi^1, ..., \pi^n)$.
    Set $\hat{\mathcal{K}}^n = \left\{ s \in \mathcal{S} | \hat{d}^{\tilde{\pi}^n} \leq \beta_n \right\}$.
    Set $r^n(s,a) = \mathbb{1}_{\{s \in \hat{\mathcal{K}}^n\}}$ and let $\mathcal{M}^n = (\mathcal{S}, \mathcal{A}, \mathcal{P}, r^n)$.
    Simulate $\mu^n := d_{s_0}^{ensemble(s_0, \tilde{\pi}^n)}$ as initial distribution using Visit($\tilde{\pi}^n$).
    Set $\theta^{n,0}$ such that $\pi_{\theta^{n,0}}(a|s) \geq \epsilon_{pp}$ for each $s, a \in \mathcal{S} \times \mathcal{A}$.
    for $k = 0, 1, ..., K_n - 1$ do                    # Find an $\epsilon$-opt-solution in $(\mathcal{M}^n, d^{\tilde{\pi}^n})$
        Choose trajectory lenght $H^{n,k}$.
        Sample trajectory $\tau^{n,k}$ from $\mathcal{M}^n$ starting from $d^{\tilde{\pi}^n}$ and following $\pi_{\theta^{n,k}}$.
        Compute approximate gradient $\hat{\nabla}_\theta L_{\lambda^n}(\theta^{n,k})$ using trajectory $\tau^{n,k}$.
        Update $\theta^{n,k+1} = \theta^{n,k} + \alpha^k \hat{\nabla}_\theta L_{\lambda^n}(\theta^{n,k})$.
    end for
    Define $\pi^{n+1} := ensemble(\pi_{\theta^{n,0}}, \pi_{\theta^{n,1}}, ..., \pi_{\theta^{n,K_n-1}})$.
end for
Simulate $d^{\tilde{\pi}^{N-1}}$ as initial distribution using Visit($\tilde{\pi}^{N-1}$).
Set $\theta^0$ such that $\pi_{\theta^0}(a|s) \geq \epsilon_{pp}$ for each $s, a \in \mathcal{S} \times \mathcal{A}$.
for $k = 0, 1, ..., K-1$ do                            # Find an $\epsilon$-opt-solution in $(\mathcal{M}, s_0)$
    Choose trajectory lenght $H^k$.
    Sample trajectory $\tau^k$ of length $H^k$ from $\mathcal{M}$ following $\pi_{\theta^k}$.
    Compute approximate gradient $\hat{\nabla}_\theta L_\lambda(\theta^k)$ using trajectory $\tau^k$.
    Update $\theta^{k+1} = \theta^k + \alpha^k \hat{\nabla}_\theta L_\lambda(\theta^k)$.
end for
```

Chapter 6

Experiments

We implement in Python the algorithms presented in the previous Chapter so that we can use them and, in particular, compare their results with the theoretical bound that we stated. This Chapter is meant to present those comparisons.

We take some examples of MDPs encountered along the Thesis and test our algorithms on them. We test two extreme cases: *with* Coverage Hypothesis, with a uniform distribution over states as μ-reset model, and *without* Coverage Hypothesis, with a single state s_0-reset model.

The first example is the simple Recycling Robot, despite not being interesting in terms of difficulty of the solution, it is useful in order to clear the ideas of what is being plotted. We then test the algorithms on the two toy examples that, at our best, expose the difficulties of exploration and exploitation. Those are the Consecutive Crossroad Traps and the Multipath tasks respectively. After that we analyze the example that combines these two aspects: the Diabolical Combinational Lock.

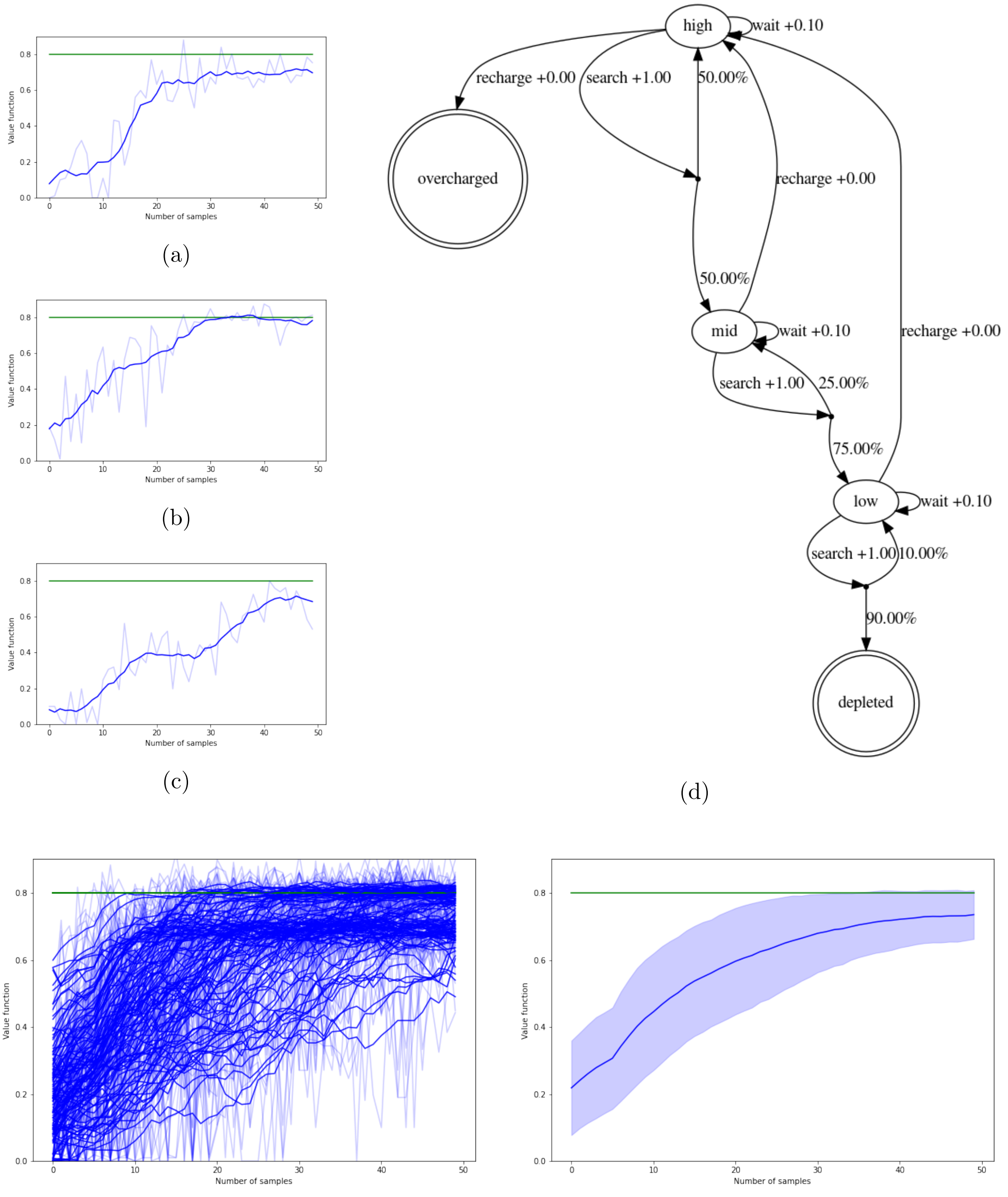
Finally, we run the algorithms on a more realistic and funnier example: the Balancing Pole.

6.1 Recycling robot

We start with a small MDP. We consider a slightly modified version of the recycling robot shown in Example 1.11 still simple but with less trivial optimum. The relative transition graph is shown in Figure 6.1d.

It has 3 non terminal states $\{$`high,mid,low`$\}$ for battery level, and 2 terminal states $\{$`depleted`, `overcharged`$\}$ for clearness of what is going on. Terminal cases are handled as Remark of Definition 1.35 with zero return, we clearly avoid them in any sampling model.

Action space is $\{$`search,wait,recharge`$\}$ as before. Searching leads to reward 1 but may lower the battery level with transition probability of 1/2, 1/4 and 1/10, from full to zero respectively. Waiting leads to a small reward of 0.1 without affecting battery level. Recharge lead to no reward but reduces the risk of depleting the battery.



(a)

(b)

(c)

(d)

(e)

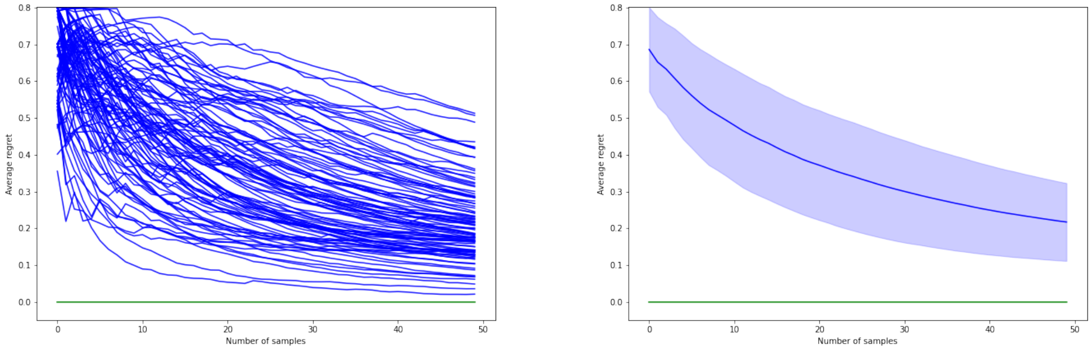
(f)

Figure 6.1: Recycling robot

First of all it is important to address the stochasticity of the process in order to produce meaningful plots. From three runs of Algorithm 1 with the same parameters we get Figure 6.1a, Figure 6.1b and Figure 6.1b, that are the value function in state `high` with respect to the number of episode sampled, each light blue point is an episode. There are two types of stochasticity we need to manage: (1) the strong oscillations in one run of the algorithm, this is due to the stochasticity of both the agent and the environment, even slightly different policies can lead to very different return on a single episode, and (2) the strong differences between one run of the algorithm and another, this is due to the stochasticity of the algorithm itself, in particular of the gradient estimator.

We manage (1) by using moving averages, of length 10 in this case, and we manage (2) by running multiple times the algorithm, 100 times in this case as shown in Figure 6.1e. This last intricate figure can be presented in a cleaner way with a pointwise average (dark blue line) and standard deviation wide confidence interval (light blue area), as in Figure 6.1f. This method of representation is implicitly used from now on in all experiments.

The same approach can be made on the regrets from Figure 6.2a to Figure 6.2b, these clearly do not need the moving average since they already are averages.



(a)

(b)

Figure 6.2: Value function convergence rates on Recycling robot.

We then try Algorithm 2 to test the anytime regret bound. Error and regret over number of episodes for 20 runs of the algorithm are shown in Figure 6.3. We can clearly appreciate the different phase of doubling length.

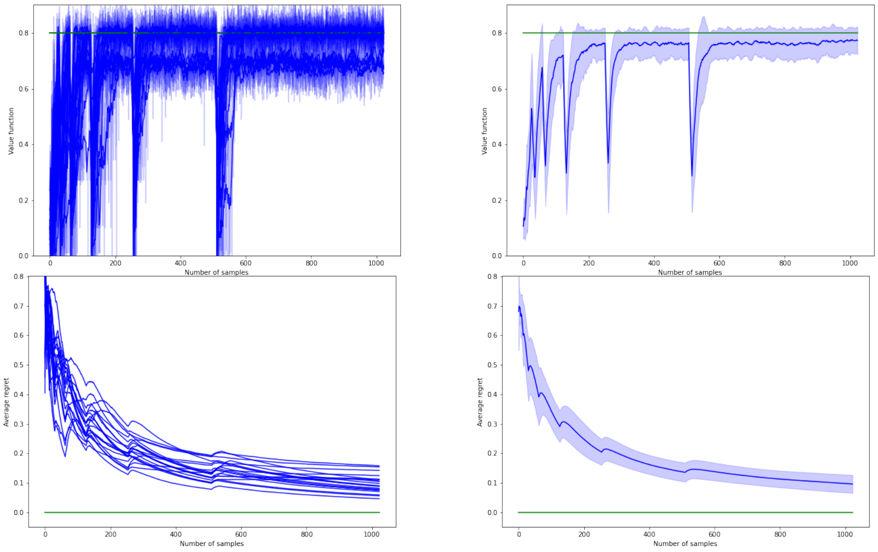


Figure 6.3: Value function and regret convergence rates on Recycling robot

Exploration is definitely not a problem in this task. The differences between assuming or not the Coverage Hypothesis are barely visible, thus exploration algorithm is neither needed nor worth space for analysis.

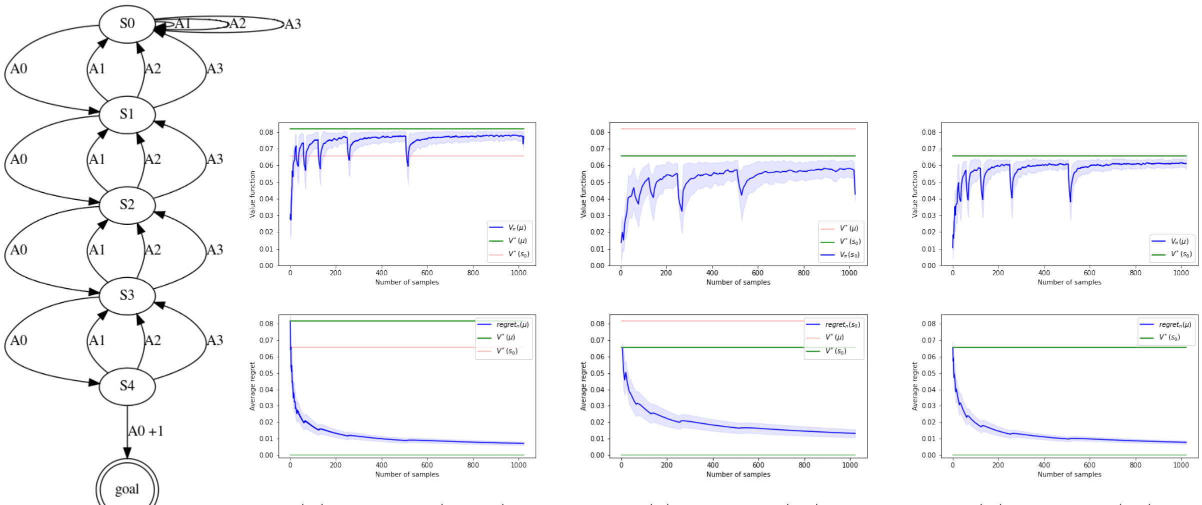
6.2 Consecutive crossroads traps

The second MDP we use for our test is the consecutive crossroads traps shown in Example 3.6. We stick to the analyzed case with 4 possible actions, only one of which makes the agent go forward approaching the only positive reward located at the end of the chain. At each level any action could be the good one, without loss of generality we set it to always be a_0, for clarity purpose.

As we saw in Proposition 3.7, the difficulty of this task is exponentially dependent on the deep H of the graph. Thus, we consider 3 specific tasks of length 5, 10 and 20.

6.2.1 Deep 5

In Figure 6.4a we show the transition graph of the considered MDP.



(b) $V_{\mathcal{M},\pi,0.9}(unif)$

(c) $V_{\mathcal{M},\pi,0.9}(s_0)$

(d) $V_{\mathcal{M},\pi,0.9}(s_0)$

(a) Transition graph

Figure 6.4: Convergence rates on consecutive crossroads traps of deep 5

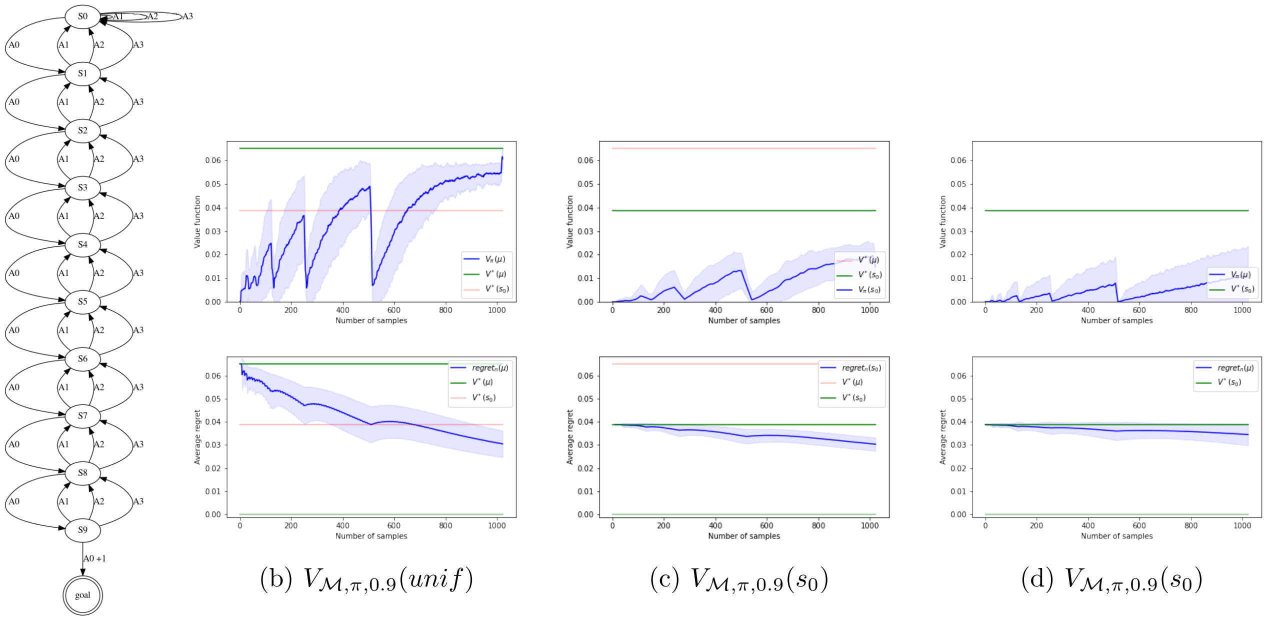
First, we want to point out again the distinction between the distribution μ of the μ-reset model and the distribution ρ involved in the measure of the error. Figures 6.4b and Figures 6.4c are sampled with a μ-sampling model with μ uniform, while Figures 6.4d are sampled with a s_0-sampling model. Figures 6.4b show value function with ρ uniform on the states while Figures 6.4c and Figures 6.4d show value function with $\rho \sim s_0$. Green horizontal bars are at the optimums of the considered value in each case.

Each plot is obtained from 100 independent runs of Algorithm 2. We plot error (up) and regret (down) over 10 phases, roughly 1000 episodes. The difference from assuming (Figure 6.4b and Figure 6.4c) or not (Figure 6.4d) the Coverage Hypothesis is visible but not significant.

6.2.2 Deep 10

In Figure 6.5a we show the transition graph of the considered MDP. All the plots on the right are obtained from 100 independent runs, over 1000 episodes. Again Figures 6.5b and Figures 6.5c are sampled with a μ-sampling model with μ uniform, while Figures 6.5d are sampled with a s_0-sampling model. Figures 6.5b show value function with ρ uniform on the states while Figures 6.5c and Figures 6.5d show value function with $\rho \sim s_0$.

In this case we start seeing the difference when assuming or not the coverage hypothesis. In fact, despite in every case there is a good improvement in average with increasing number of sampled episodes, we can see the higher variance in Figures 6.5d. In those cases with the s_0-sampling model the improvement only begins when, by chance, the uniform policy (the initial one) reaches the goal and earn a positive reward, and thus the algorithm estimates a non zero gradient. In fact some runs of the Algorithm ends with no improvement at all.



(b) $V_{\mathcal{M},\pi,0.9}(unif)$

(c) $V_{\mathcal{M},\pi,0.9}(s_0)$

(d) $V_{\mathcal{M},\pi,0.9}(s_0)$

(a) Transition Graph

Figure 6.5: Convergence rates on CCT of deep 10

Now it becomes useful to have an explorative strategy. First of all notice that we can not aim at simulating a sampling model with uniform distribution over states when we only have access to a s_0-sampling model. The best we can aim for is the state visitation distribution of the optimal explorative policy π^E, explained in Definition 3.14. In this case it is simply computable as the policy that always chose action $A0$, that is $\pi^E(A0|s) = 1$ for every $s \in \mathcal{S}$.

The state visitation distribution $d_{S0}^{\mathcal{M},\pi^E,\gamma}$ of π^E is geometric of factor γ in the depth of the state, *i.e.*

$$d_{S0}^{\mathcal{M},\pi^E,\gamma}(Sn) \propto \gamma^n. \tag{6.1}$$

Despite being far from the uniform distribution is farther from the distribution of the uniform policy π^U. This, choosing randomly, has state visitation distribution

$$d_{S0}^{\mathcal{M},\pi^U,\gamma}(Sn) \propto \left(\frac{1}{4}\right)^n. \tag{6.2}$$

Thus, when we plot the evolution of our simulated sampling model, we can plot our ideal objective (6.1) in green and our starting point simulated sampling model (6.2) in red. This is done in Figure 6.9. On the x-axis we have the 10 ordered states, on the y-axis we have the distribution value for each state. Furthermore, for every step n of our exploration algorithm we plot the simulated μ^n, the points are averages over 10 independent runs of Algorithm 4, the transparent areas are a standard deviation wide. Each run calls the s_0-sampling model approximately 2000 times.

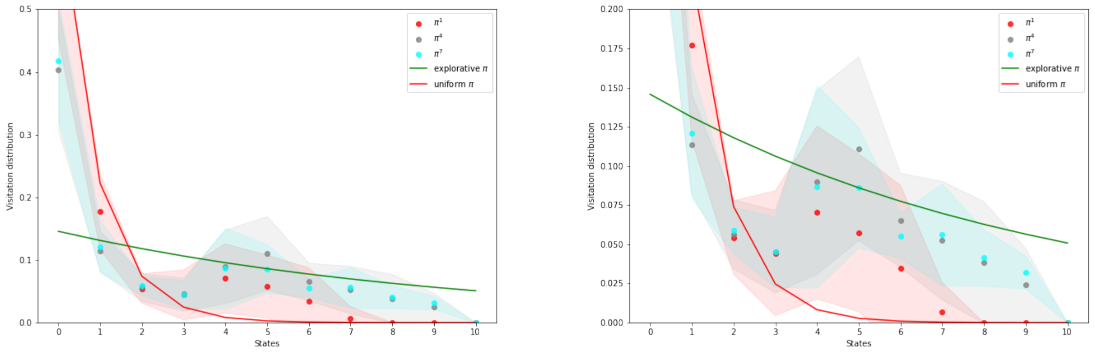
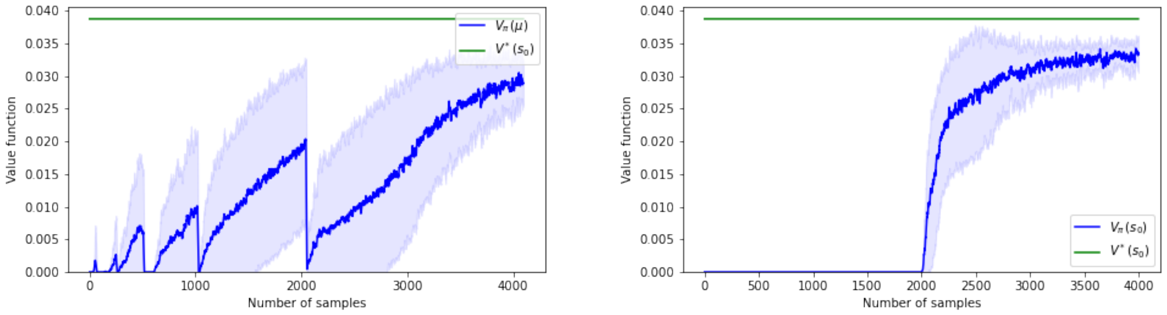


Figure 6.6: Simulated sampling model distribution μ^n for $n = 1, 4, 7$ by Algorithm 4 on consecutive crossroad task of deep 10. Wide view on the left, detail on the right.

For clarity of representation, instead of plotting the simulated sampling model distribution for every step, we only plot $d^{\tilde{\pi}^1}$, $d^{\tilde{\pi}^4}$ and $d^{\tilde{\pi}^7}$. This still shows clearly the improvement in exploration of the deeper states.

We can now compare performance between using the explorative strategy and thus Algorithm 5 (Figure 6.7b) versus using Algorithm 2 (Figure 6.7a), both assuming access only to a s_0-sampling model. Also, for fair comparison, both algorithms are runned for 10 independent times and averaged. Every run uses approximately 4000 calls to the sampling model.



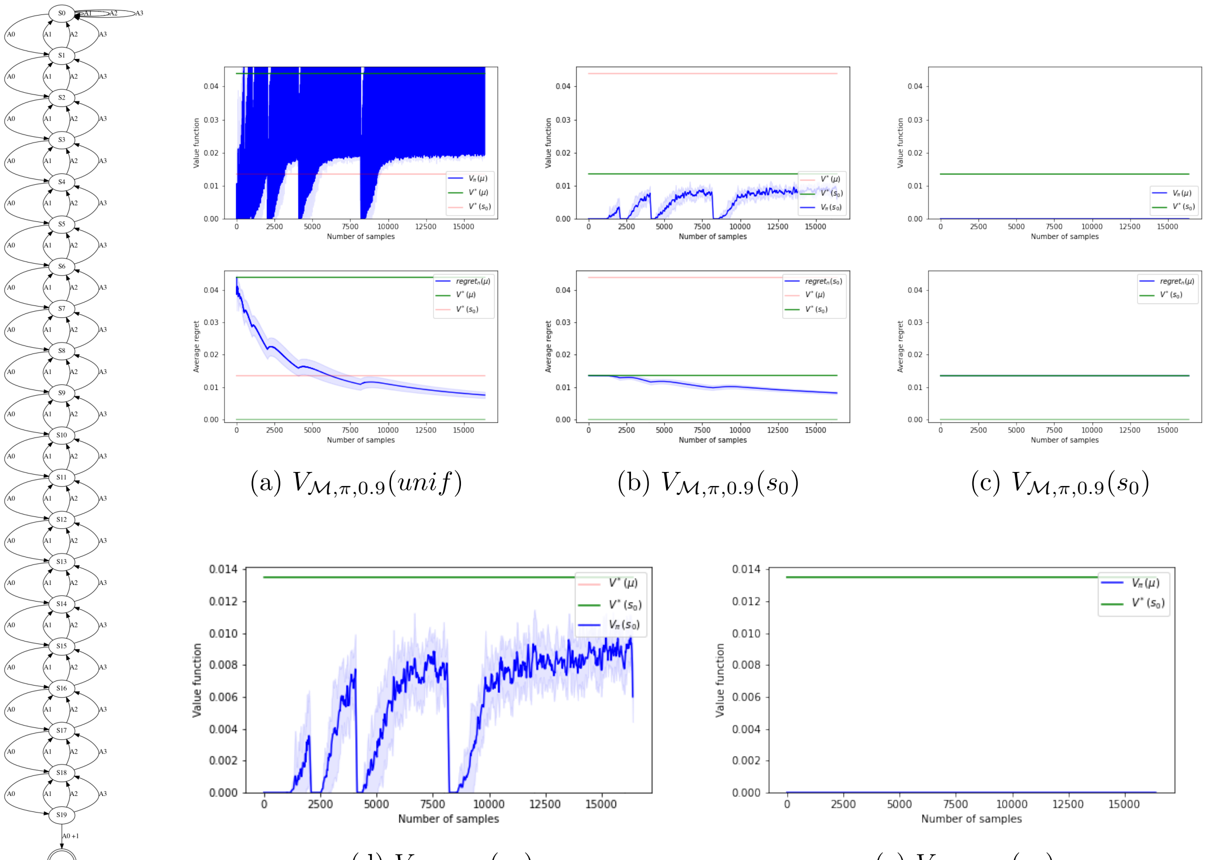
(a) $V_{\mathcal{M},\pi,0.9}(s_0)$ for Algorithm 2

(b) $V_{\mathcal{M},\pi,0.9}(s_0)$ for Algorithm 5

Figure 6.7: Comparison of convergence rates in crossroads traps example of deep 10

6.2.3 Deep 20

When considering the crossroads traps example of deep 20, the exponential behaviour starts playing a crucial role.



(a) $V_{\mathcal{M},\pi,0.9}(unif)$

(b) $V_{\mathcal{M},\pi,0.9}(s_0)$

(c) $V_{\mathcal{M},\pi,0.9}(s_0)$

(d) $V_{\mathcal{M},\pi,0.9}(s_0)$

(e) $V_{\mathcal{M},\pi,0.9}(s_0)$

Figure 6.8: Convergence rates in consecutive crossroads traps of deep 20

We run Algorithm 2 for 14 phases, roughly 16000 episodes. Average result over 20 independent runs are shown in Figure 6.8.

With Coverage Hypothesis, in Figures 6.8a the blue zone is caused by a strong oscillation in the returns, depending on where the uniform sampling model set the start of the trajectory. In Figures 6.8b the improvement is small but consistent, zoomed on the y-axis in Figure 6.8d. **Without** Coverage Hypothesis, in Figures 6.8c the improvement is not even visible, and so it remains when zoomed on the y-axis in Figure 6.8e. The gradient actually never differ from zero.

With the s_0-sampling model the initial policy π^U has a probability of hitting the final state (and thus the reward) of approximately $(1/4)^{20} \approx 10^{-12}$. It will never do that with the number of episodes in the order of the thousands that we are testing on. Now it becomes not only useful but very necessary to have an explorative strategy.

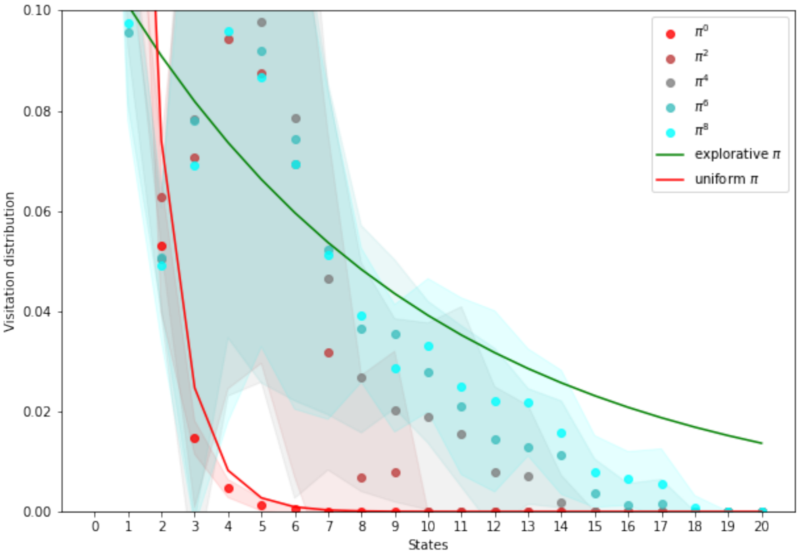


Figure 6.9: Simulated sampling model distribution μ^n for $n = 0, 2, 4, 6, 8$ by Algorithm 4.

The exploration algorithm with $N = 10$, starting from a s_0-sampling model, produce some simulated μ^n-sampling models. Some distributions μ^n are shown in Figure 6.9. This is done with approximately 2000 calls to the sampling model. Repeating this procedure for 10 times allows us to smoothen the results with averages and leave an idea of the stability with standard deviations. Results are pretty strong and consistent with the final simulated distribution (light blue) covering almost every state.

Finally, testing Algorithm 5 in this challenging explorative task leads to the convergence results shown in Figure 6.10. Again, averaged on 10 independent runs. Results are noisy but definitely better than the flatness obtained (in the same context) without an explorative strategy in Figure 6.8e.

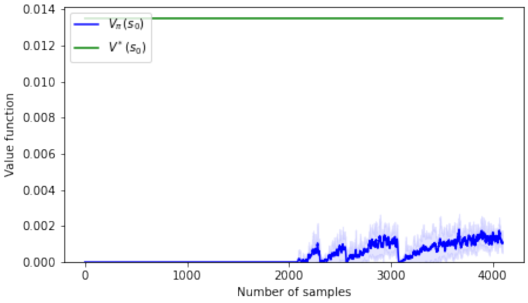
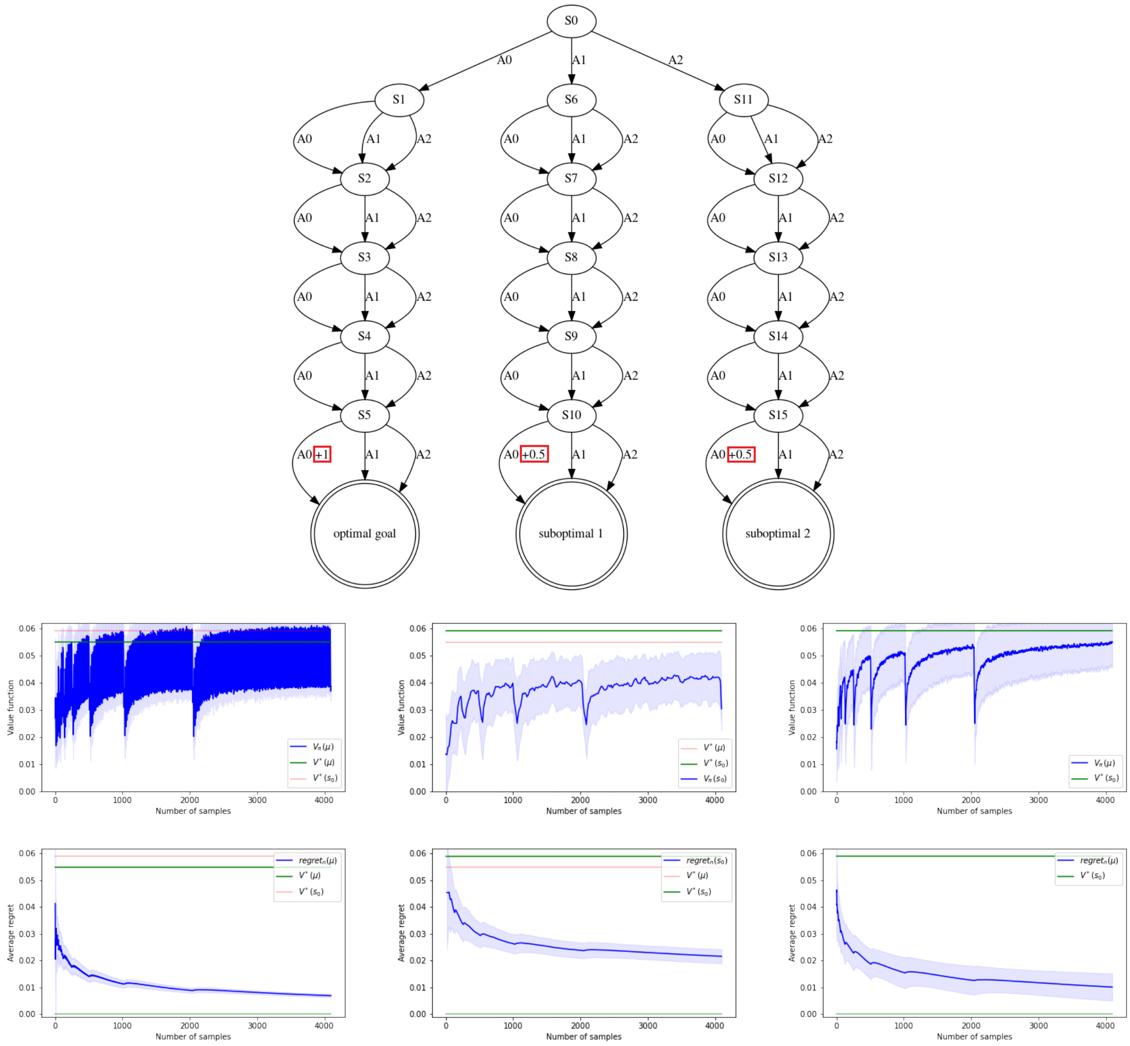


Figure 6.10: Convergence rate in consecutive crossroad task of deep 20 without Coverage Hypothesis and using the exploration strategy.

6.3 Multipath

We now consider the MDP presented in Example 3.10. The relative transition graph is shown in Figure 6.11. This is the toy problem that, at our best, underline the exploitation problem.



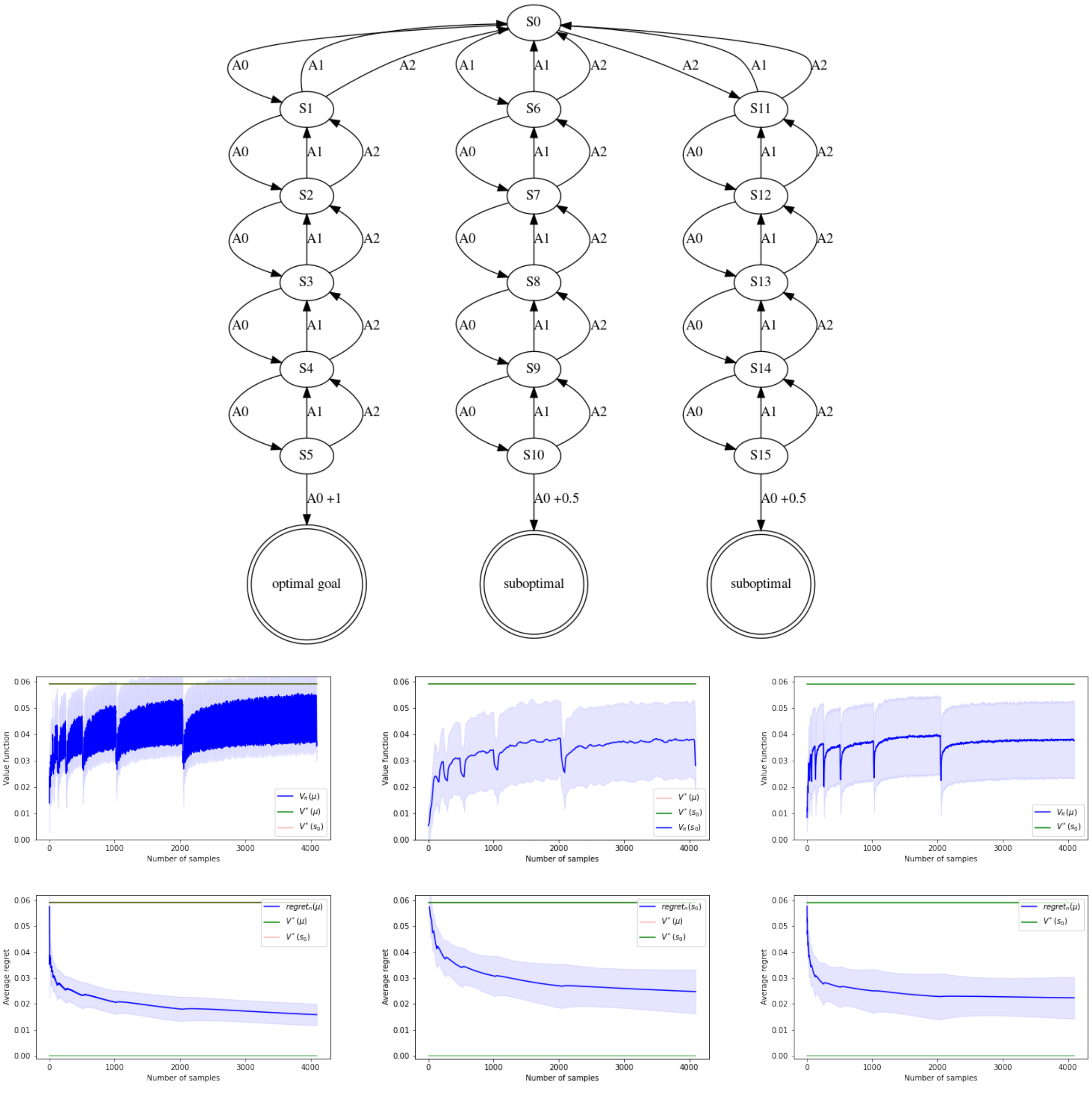
(a) $V_{\mathcal{M},\pi,0.9}(unif)$ (b) $V_{\mathcal{M},\pi,0.9}(s_0)$ (c) $V_{\mathcal{M},\pi,0.9}(s_0)$

Figure 6.11: Convergence rates in multipath task

Again, Figures 6.11a and 6.11b use access to a uniform sampling model while Figures 6.11c use access to a s_0-sampling model. Despite sometimes both methods get stuck in the half value suboptimal local maximum as we can see from the variances, this happens with decreasing probability, as we can see from the averages.

6.4 Diabolical combination lock

We test Algorithm 2 also on the challenging Diabolical Combination Lock, results are shown in Figure 6.12. Testing also Algorithm 5 in this task (and maybe also in bigger variants) would be interesting and we hope in doing this soon.



(a) $V_{\mathcal{M},\pi,0.9}(unif)$ (b) $V_{\mathcal{M},\pi,0.9}(s_0)$ (c) $V_{\mathcal{M},\pi,0.9}(s_0)$

Figure 6.12: Convergence rates in diabolical combination lock task

6.5 Balancing pole

Finally, we can test convergence in a less instructive but more realistic task. The balancing cartpole explained in Example 1.3. We can make use of the OpenAI Gym Environment that implements a cool graphic representation of what is going on. Termination happens when pole angle is greater than 15 degrees from vertical, cart position is greater than 2.4 units, or, in any case, at timestep=200.

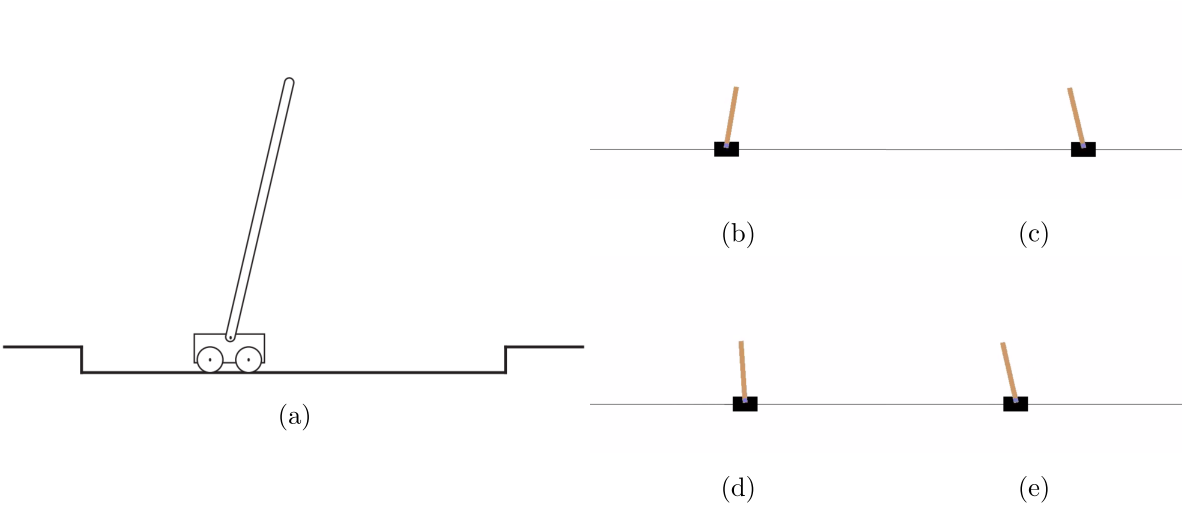


Figure 6.13: Stylized representation (a) and screenshots from simulator (b), (c), (d), (e).

We need a discretization of the state space. More dense partitions lead to better results but slower convergence rate. Moreover, we need a dynamic implementation since velocities of cart and pole are, in principle, unbounded. Details of this implementation are omitted.

Learning rate of Algorithm 1, averaged on 10 independent runs, is shown in Figure 6.14.

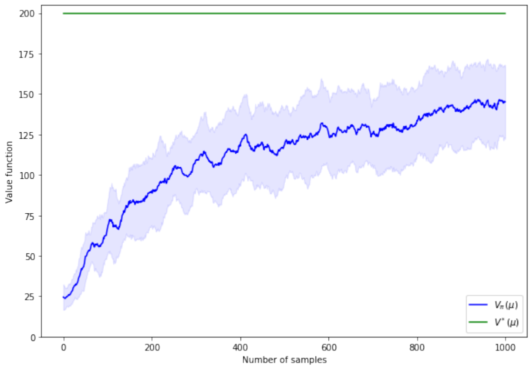


Figure 6.14: Balanced timesteps over number of episodes in the Cartpole task.

Conclusions

In this work we gave a formal description of the setting of Reinforcement Learning. We focused in clarifying some concepts as the sampling model that, despite being often neglected in literature, are key in order to fully frame and understand the exploration problem.

After giving a brief overview of the most common solving strategy families, we deepen in the policy based methods and, in particular, in the stochastic gradient ascent. We described a sublinear anytime convergence result that relies on a strong sampling model assumption and, then, we developed a strategy to ensure exploration. We proved at which extent this exploration strategy can get around the coverage hypothesis and thus leads to stronger solving methods. To the best of our knowledge, this is an original contribution in the field of Reinforcement Learning

Finally, theoretical results are empirically verified with numerical examples.

Further improvements

There is actually one more key distinction between Reinforcement Learning optimization techniques: on-policy or off-policy. The former refers to policies that learn from experience sampled according to the policies themselves, that is basically what we consider in this Thesis. The latter refers to policies (*target policies*) that learn from experience sampled according to other policies (*behaviour policies*). For example the natural off-policy extension to the on-policy ϵ-greedy improvement is the **Q-learning**, which make use of:

- a *target* policy π that is greedy with respect to the state-action value function;
- a *behaviour* policy η that is ϵ-greedy with respect to the state-action value function.

Despite particular cases, with off-policy methods an *importance sampling* is needed. In fact, we sample trajectories according to η, but we need expectation with respect to π in order to correclty improve it, therefore a weight correction is needed inside the expectation, that is the importance sampling correction

$$\frac{\pi(a|s)}{\eta(a|s)},$$

and we have to add it for every state-action pair encountered along the trajectory. More precisely, given a generic function $f : T_H \to \mathbb{R}$, the expectation with respect to trajectories satisfies

$$\mathbb{E}_{\tau \sim \mathbb{P}(\cdot|\pi,\mathcal{M},s_0)}[f(s_0, a_0, ..., s_{H-1}, a_{H-1})] = \mathbb{E}_{\tau \sim \mathbb{P}(\cdot|\eta,\mathcal{M},s_0)} \left[f(s_0, a_0, ..., s_{H-1}, a_{H-1}) \prod_{t=0}^{H-1} \frac{\pi(a_t|s_t)}{\eta(a_t|s_t)} \right]$$

where the first expectation is made with respect to the *target* policy π and used for improvement, while the second expectation is made with respect to the **behaviour** policy η and is the only thing we can estimate following η and not knowing the environment dynamincs.

Thus, in Algorithm 5 an improvement can clearly be made by letting our optimal policy also learn from the experience sampled during the development of the explorative policy $\tilde{\pi}^N$. Otherwise in the first half of the algorithm learning is flat, as shown in Figure 6.7b and Figure 6.10. This improvement would require further analysis and may be a subject for future work.

Furthermore, there may be a memory problem in the algorithm as we stated it. In fact it requires memorizing all the policies tried at every step. This memory abuse can be lightened at the cost of theoretical guarantees. In practical usage a *genetic algorithm* may be a great choice. Without digging into deeper details this procedure represents the survival of the fittest, and it may be implemented with the visitation of the poorly visited states as fit measure.

Bibliography

[1] A. Agarwal, M. Henaff, S. Kakade, and W. Sun. Pc-pg: Policy cover directed exploration for provable policy gradient learning. *arXiv:2007.08459. NIPS 2020*, 2020.

[2] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Proceedings of Machine Learning Research vol 125:1–3*, 2020.

[3] S. M. Kakade. *On the sample complexity of reinforcement learning.* PhD thesis, UCL (University College London), 2003.

[4] M. Kearns, Y. Mansour, and A. Y. Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine learning*, 49(2):193–208, 2002.

[5] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[6] M. L. Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990.

[7] R. S. Sutton, A. G. Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 2nd edition, 2018.

[8] R. J. Williams and J. Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

[9] Y. Yu. Towards sample efficient reinforcement learning. In *IJCAI*, pages 5739–5743, 2018.

[10] J. Zhang, J. Kim, B. O'Donoghue, and S. Boyd. Sample efficient reinforcement learning with reinforce. *arXiv:2010.11364 to appear in AAAI 2021*, 2020.