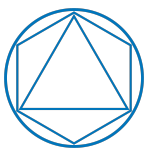
DEPARTMENT OF MATHEMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis

Safe Reinforcement Learning with Safety Critic Policy Optimization



DEPARTMENT OF MATHEMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis

Safe Reinforcement Learning with Safety Critic Policy Optimization

Author:	Mhamed Jaafar
Supervisor:	Prof. Dr. -Ing. Alois Knoll
Advisor:	Shangding Gu
Submission Date:	15.03.2023

Abstract

Incorporating safety is a critical requirement for extending the practical applications of reinforcement learning to real-world scenarios. Constrained Markov Decision Processes (CMDP) have been utilized to address this, introducing a separate cost function to represent safety violations. This approach eliminates the need for designing a reward function that considers safety. The Lagrangian relaxation technique has been used in prior algorithms to transform constrained optimization problems into unconstrained dual problems. However, predicting unsafe behavior in these algorithms can be inaccurate, leading to instability in learning the Lagrange multiplier.
This thesis presents a novel safe reinforcement learning algorithm. We define the safety critic, which enables nullifying rewards gained by violating safety constraints. The algorithm automatically manages the trade-off between adhering to safety constraints and maximizing the return. The effectiveness of our algorithm is empirically demonstrated by benchmarking it against five other safe reinforcement learning techniques.

Contents

1 Introduction

The field of reinforcement learning (RL) involves an agent learning to take actions in an environment to maximize a long-term reward signal [25]. It has been widely adopted in several fields such as finance [1, 14], transportation schedule [15, 3] and autonomous driving [10, 16, 17].
However, the absence of safety assurance significantly hinders the practical application of RL algorithms to real-world problems. RL agents rely on reward signals to make decisions, which can disregard safety restrictions. For instance, an RL agent managing a self-driving vehicle may receive a substantial reward for driving at high speeds, but this behavior could increase the risk of collision with other objects; RL agents may prioritize maximizing rewards instead of adhering to safe behavior, which could result in dangerous or disastrous consequences [9].
Moreover, RL requires exploration to learn an optimal policy, which makes designing safe RL algorithms challenging. Instead of changing the reward function to include a safety term, Ray et al. propose that safety specification should be separate from task performance [21]. This motivates the definition of a cost function separately from the reward function and gives rise to constrained Markov decision process (CMDP). The additional constraint component of CMDPs enhances the flexibility of modelling problems with trajectory-based constraints compared to other approaches that customize immediate costs in MDPs to handle constraints.
The existing model-free on-policy algorithms suffer from low sample efficiency [12], which causes slow conversion and more unsafe behaviors while training. Most approaches use Lagrange multiplier to transform the safety constraints problem into its unconstrained counterpart.
To address the above issues, we propose a new algorithm, safety critic policy optimization (SCPO). Improving the return mostly contradicts respecting safety constraints. Therefore, balancing return and cost during training is not trivial. The core idea behind our method is to nullify the reward obtained from visiting unsafe states. We use a safety critic to approximate the safety of a state action pair and reduce its reward accordingly. The safety critic can be initialized pessimistically; every state is unsafe until proven otherwise. This leads to safer policies throughout training. SCPO aligns both objectives of maximizing return and adhering to safety constraints which leads to greater sample efficiency; training time is shorter, and unsafe trajectories are rarely generated.

In this thesis, we motivate the introduction of the safety critic and provide a theoretical analysis inspired by the trust region paper [24]. The Lagrange multiplier method is a special case of our algorithm when specific hyperparameters are chosen. The effectiveness of our algorithm is empirically demonstrated by benchmarking it against five other popular safe RL techniques. We use the ball agent from safety bullet gym [8] on four tasks: circle, reach, gather, run.

2 Related work

By defining a discounted cumulative cost, safe RL aims to ensure safety constraint satisfaction while maximizing return. Most methods utilize Lagrangian relaxation to transform the constrained problem it its unconstrained counterpart. For instance, Joshua Achiam et al. [2] uses the Lagrangian method to develop an actor-critic algorithm. Ray et al. [21] utilizes the update rules from trust region optimization TRPO [24] and Proximal Policy Optimization PPO [23] to obtain two Lagrangian-based safe RL algorithms named TRPO-Lagrangian and PPO-Lagrangian. The authors of [21] also provide a safe RL environment implementation named safety gym, which was used to demonstrate the validity of the proposed algorithms.
Despite the widespread use of Lagrangian relaxation in solving the safe RL problem, some approaches choose not to utilize this method. For example, Joshua Achiam et al. [2] introduce the CPO algorithm, which approximates the constrained optimization problem using a quadratic constrained optimization to handle the constraints. Nonetheless, the computational cost of CPO exceeds that of PPO-Lagrangian because it involves the computation of the Fisher information matrix and utilizes the second Taylor expansion to optimize objectives. Furthermore, the approximation and sampling errors associated with CPO may adversely affect the overall performance, and the convergence analysis might be challenging. In addition, implementing an additional recovery policy may require a larger number of samples.
Inspired by CPO, Projection-based Constrained Policy Optimisation (PCPO) [26] is a 2-stage algorithm. It employs Trust Region Policy Optimization [24] to maximize the reward and then projects the policy to a feasible region to satisfy the safety constraints. Nonetheless, second-order proximal optimization is used in both steps, which increases the computation cost of this algorithm.
Pham et al. introduce a technique called OptLayer [19], in which they utilize stochastic control policies to maximize rewards, while integrating a neural network layer to ensure safety during deployment. The practical applications of OptLayer show promising results in enhancing safety. The authors of A-CRL (State Augmented Constrained Reinforcement Learning) [4] present a solution for a CMDP problem in which the optimal policy cannot be obtained solely through regular rewards. The proposed method aims to solve the monitor problem in CMDP, and dual gradient descent is utilized to identify feasible trajectories and ensure safety. However, convergence rate analysis is not yet provided for A-CRL and OptLayer. Hasanbeig et al. [11] proposed a

safe RL approach that uses reward shaping and linear temporal logic (LTL). This method ensures safety during exploration by synthesizing policies that satisfy the LTL constraints. The LTL formula acts as a constraint during exploration, enabling the search for safe policies. Although it demonstrates impressive safety performance, determining the logical constraints is crucial to balance the trade-off between safety performance and reward values.
Lyapunov function is a new approach to solving safe RL. It constrains the agent's actions by implementing the control law of Lyapunov functions, eliminating unsafe actions from the action set. The experiments conducted using this method have shown that it can effectively produce safe actions for the control problems [18]. Furthermore, Yinlam Chow et al. [6] use the Lyapunov approach to propose two classes of policy optimization algorithms for continuous tasks, namely θ-projection and α-projection. However, the Lyapunov approach requires an initial policy that is feasible [5]. While this initial feasible policy can converge to the optimal policy under strict restrictions, creating such a policy can be challenging. This means that the Lyapunov approach must be used in conjunction with another optimizer. This requirement limits the generality of the Lyapunov approach as a general solution to safe RL problems. Please see the appendix 10.1 for a detailed analysis.

3 Preliminaries

To model safe reinforcement learning problems, we use the well-studied framework of constrained Markov decision process (CMDP). A Markov decision process (MDP) is a tuple $(\mathcal{X}, \mathcal{A}, r, P, s_0)$ where $\mathcal{X}$ is the set of states, $\mathcal{A}$ is the action space, $r : \mathcal{X}, \mathcal{A} \to \mathbb{R}$ is the immediate reward function, $P : \mathcal{X}, \mathcal{A}, \mathcal{X} \to [0, 1]$ is the environment transition probability distribution and s_0 is the initial state. A CMDP extends on top of an MDP and is defined as $(\mathcal{X}, \mathcal{A}, r, c, P, s_0, c_0)$, where $c : \mathcal{X} \to \mathbb{R}_+$ is the immediate cost function and c_0 is the maximum allowed cumulative cost.
Let $\Delta = \{\pi : \mathcal{X}, \mathcal{A} \to [0, 1] \mid \forall x \in \mathcal{X} \quad \sum_{a \in \mathcal{A}} \pi(a \mid x) = 1\}$ be the set of all policies.
Given a policy π, most papers define the expected cumulative cost as follows:

$$\mathcal{C}_\pi(s_0) = \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t c(S_t) \mid S_0 = s_0 \right]$$

The safety constraint is defined as $\mathcal{C}(s_0) \leq c_0$. The CMDP goal is to solve the following constrained optimization problem:

$$\pi^* \in \max_{\pi \in \Delta} \{ V_\pi(s_0) \mid \mathcal{C}_\pi(s_0) \leq c_0 \}$$

4 Safety

This chapter introduces the safety critic $V^c(s)$, which represents the probability of generating an unsafe trajectory from a given state s. We also define $Q^c(s, a)$ as the probability of generating an unsafe trajectory from state s after taking action a. These functions enable the efficient search for an optimal safe policy. The rationale is as follows: The agent is in a state s, selecting action a_1 leads to a considerably high reward, but violates the safety constraint. Alternatively, selecting action a_1' does not violate the constraint but offers significantly less reward. The agent must balance two objectives: maximizing return and minimizing cost. We argue that the reward obtained by choosing the unsafe action a_1 should be nullified. It is counterproductive to take into consideration rewards obtained by acting in an unsafe manner. By changing our reward function to $r'(s, a) = r(s, a)\ Q^c(s, a)$, we decrease the effect of unsafe actions. If $Q^c(s, a_1) = 0$, indicating that a_1 always violates safety constraints, the agent is not incentivized to select it. In contrast, if $Q(s, a_1') = 1$, the reward obtained from choosing a_1' is unchanged, e.g. $r'(s, a_1') = Q^c(s, a_1')r(s, a_1') = r(s, a_1')$.

4.1 Augmented states with Cumulative cost

Incorporating safety into reinforcement learning necessitates that the agent behaves differently based on the current cumulative cost. Specifically, the agent can take actions that increase the cumulative cost if the maximum cumulative cost c_0 has not been exceeded.
To provide a concrete example of this principle, we define the following CMDP 4.1 with two states s^0, s^1 and two actions a^0, a^1. The episode length is 10 and $c_0 = 5$.

Let us consider the case where $s_t = \mathbb{1}[s_t = s^1]$. To behave optimally, the agent should select action a^0 and remain in state s^1 for five time steps by taking action a^1. At $t = 5$, the maximum cumulative cost constraint is reached. To avoid violating the safety constraint, the agent must select action a^0, transitioning to state s^0. However, this decision cannot be made because it lacks information regarding the current cumulative cost. In this scenario, the optimal solution is $\pi(a \mid s) = 0.5$ for any given state and action; The agent is forced to guess when the maximum cumulative cost has been reached.
To address this issue, we introduce a new variable q_t representing the cumulative cost at time t. By including q_t in the state representation, we can solve

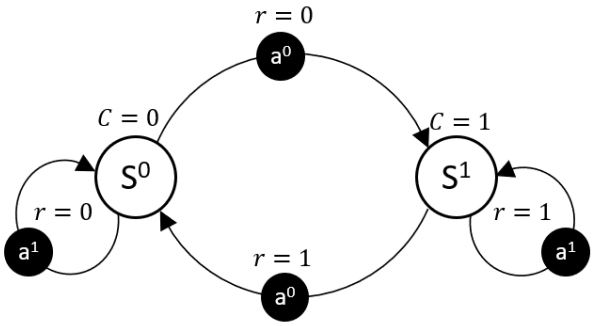


Figure 4.1: CMDP where the optimal policy is stochastic if the state representation does not contain the cumulative cost; the agent has to guess when the constraint violate occurs, e.g. $\pi(a \mid s) = 0.5$. When the cumulative cost is included in the state representation, the agent can make more informed decisions and reach a deterministic policy, e.g. $\pi(a \mid s) = 0$ or 1

the problem discussed above. We define the new state as $s'_t = (\mathbb{1}[s_t = S^1], q_t)$. However, this increases the number of possible states from 2 to 20. When dealing with continuous tasks, the cumulative cost might be unbounded. Hence, we introduce q_t^{clip}:

$$q_t^{\text{clip}} = \text{clip}(\frac{q_t}{c_0}, 0, 1) \tag{4.1}$$

$\frac{q_t}{c_0}$ is clipped because the agent's behavior should remain the same after reaching the maximum allowed cumulative cost. In the upcoming sections, we assume all states are augmented using q_t^{clip}.

4.2 Safety critic

For any state $s \in \mathcal{X}$ and trajectory, we define the function $f : \mathcal{X} \to [0, 1]$ as follows:

$$f(s) = \begin{cases} 1 & \text{if } s \text{ is safe, } q^{\text{clip}} \leq 1 \\ 0 & \text{if } s \text{ is unsafe, } q^{\text{clip}} > 1 \end{cases}$$

For a trajectory $\tau = s_0, a_0, s_1, a_1 ... s_{T-1}$ we define $f(\tau) = \prod_{t=0}^{T-1} f(s_t)$. We call the trajectory safe if $f(\tau) = 1$

Definition 4.2.1 *For any $s \in \mathcal{X}$, action a and policy π*

$$V_\pi^c(s) = \mathbb{E}_\pi \left[\prod_{t=0}^{T-1} f(S_t) \mid S_0 = s \right] \qquad Q_\pi^c(a, s) = \mathbb{E}_\pi \left[\prod_{t=0}^{T-1} f(S_t) \mid S_0 = s, A_0 = a \right]$$

$V_\pi^c(s)$ is the probability of generating a safe trajectory starting from state s:
$V_\pi^c(s) = \sum_\tau P_\pi[\tau \mid S_0 = s]\ f(\tau)$

Theorem 4.2.1

$$Q_\pi^c(a,s) = \sum_{s' \in \mathcal{X}} p(s' \mid s,a)\ V_\pi^c(s') \tag{4.2}$$

Proof:

$$\begin{aligned}
Q_\pi^c(a,s) &= \mathbb{E}_\pi \left[\prod_{t=0} f(S_t) \mid S_0 = s, A_0 = a \right] \\
&= \sum_{s' \in \mathcal{X}} p(s' \mid s,a)\, \mathbb{E}_\pi \left[\prod_{t=0} f(S_t) \mid S_0 = s, A_t = a, S_1 = s' \right] \\
&= \sum_{s' \in \mathcal{X}} p(s' \mid s,a) f(s)\ \mathbb{E}_\pi \left[\prod_{t=1} f(S_t) \mid S_0 = s, A_t = a, S_1 = s' \right] \\
&= \sum_{s' \in \mathcal{X}} p(s' \mid s,a)\ f(s)\ \mathbb{E}_\pi \left[\prod_{t=0} f(S_t) \mid S_0 = s' \right] \\
&= \sum_{a \in \mathcal{A}} \pi(a \mid s) \sum_{s' \in \mathcal{X}} p(s' \mid s,a)\ V_\pi^c(s')
\end{aligned}$$

The cumulative cost can only increase when transitioning from one state to another. If $f(s_t) = 0$ then all consecutive states are also unsafe. Therefore, the following holds:
$f(s)p(s' \mid s,a)\, \mathbb{E}_\pi\left[\prod_{t=1} f(S_t) \mid S_1 = s'\right] = p(s' \mid s,a)\, \mathbb{E}_\pi\left[\prod_{t=1} f(S_t) \mid S_1 = s'\right] = 0$. The equality is trivial when $f(s) = 1$.

Theorem 4.2.2

$$\nabla V_\pi^c(s) = \mathbb{E}_\pi \left[\sum_{t=0} \nabla \log(\pi(S_t, A_t)) Q_\pi^c(S_t, A_t) \mid S_0 = s \right]$$

Proof:

$$\begin{aligned}
\nabla V_\pi^c(s) &= \sum_{a \in \mathcal{A}} \nabla(\pi(a \mid s)\ Q_\pi^c(a \mid s)) \\
&= \sum_{a \in \mathcal{A}} \nabla \pi(a \mid s)\ Q_\pi^c(a \mid s) + \sum_{a \in \mathcal{A}} \pi(a \mid s)\ \nabla Q_\pi^c(a \mid s) \\
&= \sum_{a \in \mathcal{A}} \pi(a \mid s) \nabla \log(\pi(a \mid s))\ Q_\pi^c(a \mid s) + \sum_{a \in \mathcal{A}} \pi(a \mid s)\ \sum_{s'} p(s' \mid s,a)\ \nabla V_\pi^c(s') \qquad (4.2) \\
&= \mathbb{E}_\pi \left[\sum_{t=0} \nabla \log(\pi(S_t \mid A_t)) Q_\pi^c(S_t, A_t) \mid S_0 = s \right]
\end{aligned}$$

For an arbitrary state $s \in \mathcal{X}$ and action $a \in \mathcal{A}$ we defined the safety advantage

$A^c_\pi(a,s) = Q^c_\pi(a \mid s) - V^c_\pi(s)$

Corollary 4.2.2.1

$$\nabla V^c_\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0} \nabla \log(\pi(S_t, A_t)) A^c_\pi(S_t, A_t) \mid S_0 = s \right]$$

Proof:

$$\begin{aligned}
&\mathbb{E}_\pi \left[\sum_{t=0} \nabla \log(\pi(S_t, A_t)) Q^c_\pi(S_t, A_t) \mid S_0 = s \right] \\
&= \sum_{t=0} \sum_{s' \in \mathcal{X}} P_\pi[S_t = s' \mid S_0 = s] \sum_{a \in \mathcal{A}} \pi(a \mid s') \frac{\nabla \pi(a \mid s)}{\pi(a \mid s)} A^c_\pi(a \mid s) \\
&= \sum_{t=0} \sum_{s' \in \mathcal{X}} P_\pi[S_t = s' \mid S_0 = s] \sum_{a \in \mathcal{A}} \pi(a \mid s') \frac{\nabla \pi(a \mid s)}{\pi(a \mid s)} Q^c_\pi(s', a) - V(s') \\
&= \sum_{t=0} \sum_{s' \in \mathcal{X}} P_\pi[S_t = s' \mid S_0 = s] \sum_{a \in \mathcal{A}} \pi(a \mid s') \frac{\nabla \pi(a \mid s)}{\pi(a \mid s)} Q^c_\pi(s', a) - \sum_{a \in \mathcal{A}} \nabla \pi(a \mid s') V(s') \\
&= \sum_{t=0} \sum_{s' \in \mathcal{X}} P_\pi[S_t = s' \mid S_0 = s] \sum_{a \in \mathcal{A}} \pi(a \mid s') \frac{\nabla \pi(a \mid s)}{\pi(a \mid s)} Q^c_\pi(s', a) - \nabla 1 \; V(s') \\
&= \sum_{t=0} \sum_{s' \in \mathcal{X}} P_\pi[S_t = s' \mid S_0 = s] \sum_{a \in \mathcal{A}} \pi(a \mid s') \frac{\nabla \pi(a \mid s)}{\pi(a \mid s)} Q^c_\pi(s', a) \\
&= V^c_\pi(s)
\end{aligned}$$

4.3 Q^c and V^cestimate

Let $\tau = (s_0, a_0, s_1, a_1 ... a_{T-1}, s_{T-1})$ be a trajectory. $Q^c_\pi(s_t, a_t)$ can be expressed as follows:

$$\begin{aligned}
Q^{c,(1)}_\pi(s_t, a_t) &= \mathbb{E}_\pi[f(S_t) \; V^c_\pi(S_{t+1}) \mid S_t = s_t, A_t = a_t] \\
Q^{c,(2)}_\pi(s_t, a_t) &= \mathbb{E}_\pi[f(S_t) \; f(S_{t+1}) \; V^c_\pi(S_{t+1}) \mid S_t = s_t, A_t = a_t] \\
&\quad ... \\
Q^{c,(k)}_\pi(s_t, a_t) &= \mathbb{E}_\pi[f(S_t) f(S_{t+1}) ... f(S_{t+k-1}) \; V^c_\pi(S_{t+k})) \mid S_t = s_t, A_t = a_t]
\end{aligned}$$

Let $\hat{Q}^{c,(k)}_\pi$ be the one sample estimate of $Q^{c,(k)}_\pi$:

$$\hat{Q}_\pi^{c,(1)}(s_t, a_t) = f(s_t)\ V_\pi^c(s_{t+1})$$
$$\hat{Q}_\pi^{c,(2)}(s_t, a_t) = f(s_t)\ f(s_{t+1})\ V_\pi^c(s_{t+2}) = f(s_{t+1})\ V_\pi^c(s_{t+2})$$
$$\dots$$
$$\hat{Q}_\pi^{c,(k)}(s_t, a_t) = f(s_t)\ f(s_{t+1}) \dots f(s_{t+k-1})\ V_\pi^c(s_{t+k}) = f(s_{t+k-1})\ V_\pi^c(s_{t+k})$$

Any weighted average of $\hat{Q}_\pi^{c,(k)}$ can be used as an estimate. Let $\bar{Q}_\pi^c$ be such an estimate. We define the estimate of the safety advantage as follows:

$$\hat{A}_\pi^c(s, a) = \bar{Q}_\pi^c(s, a) - V_\pi^c(s)$$

We can also introduce the notion of discount when estimating V^c: Let T be the episode length, we define $\hat{V}_\gamma^c$ as follows:

$$\forall \gamma \in (0, 1] \quad \hat{V}_\gamma^c(s_t) = \begin{cases} (1-\gamma)\ f(s_t) + \gamma \hat{V}_\gamma^c(s_{t+1}) & \text{if } 0 \leq t < T-1 \\ f(s_{T-1}) & \text{otherwise} \end{cases} \tag{4.3}$$

We notice that $V_1^c(s_t) = f(S_{T-1})$, which is the one sample estimate of $V^c(s_t)$. If $f(S_{T-1}) = 1$, then for all $\gamma \in (0, 1]$ $V_\gamma^c(s_t) = 1$

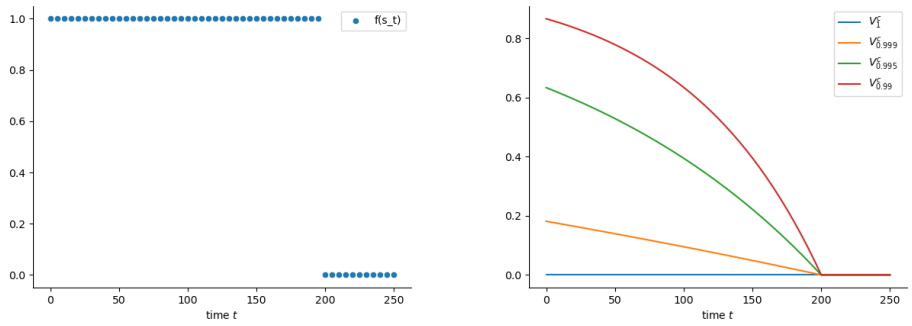


Figure 4.2: The left graph represents the value of $f(s_t)$ for a specific episode. In the right graph we plot V_γ^c for different values of γ

Unrolling the recursion, V_γ^c can be written as follows:

$$V_\gamma^c(s_t) = \begin{cases} \sum_{k=0}^{T-t-1} (1-\gamma)\gamma^k f(s_{t+k}) & \text{if } 0 \leq t < T-1 \\ f(S_{T-1}) & \text{otherwise} \end{cases}$$

Choosing an appropriate γ value can reduce variance when estimating V^c.

If the current state is unsafe, depending on the problem at hand, we can roughly analyze how much previous states led to unsafely. The upper bound of V_γ^c is 1. Therefore, it can still be interpreted as the probability of visiting an unsafe state.

$$\begin{aligned} V_\gamma^c(s_t) &= \sum_{k=0}^{T-t-1} (1-\gamma)\gamma^k f(s_{t+k}) \\ &\leq \sum_{k=0}^{T-t-1} (1-\gamma)\gamma^k \\ &\leq (1-\gamma)\frac{1-\gamma^{T-t}}{1-\gamma} \\ &\leq 1-\gamma^{T-t} \\ &\leq 1 \end{aligned}$$

5 Trust region

To find an optimal policy, we randomly initialize a policy π, then iteratively improve it by collecting trajectories and following the gradient of the objective function. When iterating, stochastic gradient ascent is used with batches of configurable size. After one epoch, the policy changes from π to π'. The trajectories were collected using π, which is different from the intermediate policy we generated after one epoch. The trust region paper [24] (TRPO) addresses this problem and proposes a solution that guarantees the objective function's improvement using samples generated by the initial policy π. In this section, we present new theoretical results inspired by the TRPO paper.

5.1 General results

Theorem 5.1.1 *For arbitrary policies π and π' and state s the following equality holds:*

$$V^c_{\pi'}(s) = V^c_{\pi}(s) + \mathbb{E}_{\pi'}\left[\sum_{t=0}(1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)})Q^c_{\pi}(S_t, A_t) \mid S_0 = s\right] \tag{5.1}$$

Proof:

$$\begin{aligned}
(V^c_{\pi'} - V^c_{\pi})(s) &= \sum_{a\in\mathcal{A}} \pi'(a \mid s)\ Q^c_{\pi'}(s,a) - \sum_{a\in\mathcal{A}} \pi(a \mid s)\ Q^c_{\pi}(s,a) \\
&= \sum_{a\in\mathcal{A}} \pi'(a \mid s)\ Q^c_{\pi'}(s,a) - \sum_{a\in\mathcal{A}} \pi(a \mid s)\ Q^c_{\pi}(s,a) \pm \sum_{a\in\mathbf{A}} \pi'(a \mid s)\ Q^c_{\pi}(s,a) \\
&= \sum_{a\in\mathcal{A}} \pi'(a \mid s)\ (Q^c_{\pi'}(s,a) - Q^c_{\pi}(a,s)) + \sum_{a\in\mathcal{A}} (\pi'(a \mid s) - \pi(a \mid s))\ Q^c_{\pi}(s,a) \\
&= \sum_{a\in\mathcal{A}}\sum_{s'\in\mathcal{X}} \pi'(a \mid s)\ p(s' \mid s,a)\ (V^c_{\pi'} - V^c_{\pi})(s') + \sum_{a\in\mathcal{A}} \pi'(s,a)(1 - \frac{\pi(a \mid s)}{\pi'(a \mid s)})\ Q^c_{\pi}(s,a) \\
&= \mathbb{E}_{\pi'}\left[\sum_{t}(1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)})Q^c_{\pi}(S_t, A_t) \mid S_0 = s\right] \quad (4.2)
\end{aligned}$$

Maximizing $V'_{\pi}(s_0)$ is therefore equivalent to maximizing $\mathbb{E}_{\pi'}\left[\sum_{t=0}(1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)})Q^c_{\pi}(S_t, A_t) \mid S_0 = s\right]$. However, evaluating this term requires sampling trajectories using π', which is unavailable until we explicitly compute it. Therefore, we approximate the term using trajectories sampled using π, following the approach in [24].

We also observe that the below proofs only used shared properties between V^c and V.

Theorem 5.1.2 *For arbitrary policies π and π', state s and $\gamma \in [0,1]$ the following equality holds:*

$$V_{\pi'}(s) = V_\pi(s) + \mathbb{E}_{\pi'}\left[\sum_{t=0} \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) Q_\pi(S_t, A_t) \mid S_0 = s\right] \tag{5.2}$$

Proof:

$$\begin{aligned}
(V_{\pi'} - V_\pi)(s) &= \sum_{a \in \mathcal{A}} \pi'(a \mid s)\ Q_{\pi'}(s,a) - \sum_{a \in \mathcal{A}} \pi(a \mid s)\ Q_\pi(s,a) \\
&= \sum_{a \in \mathcal{A}} \pi'(a \mid s)\ Q_{\pi'}(s,a) - \sum_{a \in \mathcal{A}} \pi(a \mid s)\ Q_\pi(s,a) \pm \sum_{a \in \mathbf{A}} \pi'(a \mid s)\ Q_\pi(s,a) \\
&= \sum_{a \in \mathcal{A}} \pi'(a \mid s)\ (Q_{\pi'}(s,a) - Q_\pi(a,s)) + \sum_{a \in \mathcal{A}} (\pi'(a \mid s) - \pi(a \mid s))\ Q_\pi(s,a) \\
&= \sum_{a \in \mathcal{A}} \sum_{s' \in \mathcal{X}} \pi'(a \mid s)\ p(s' \mid s,a)\ \gamma\ (V_{\pi'} - V_\pi)(s') + \sum_{a \in \mathcal{A}} \pi'(s,a)(1 - \frac{\pi(a \mid s)}{\pi'(a \mid s)})\ Q_\pi(s,a) \\
&= \mathbb{E}_{\pi'}\left[\sum_t \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) Q_\pi(S_t, A_t) \mid S_0 = s\right]
\end{aligned}$$

In the following, we prove properties with respect to V. The results can be generalized to V^c because only common algebraic properties of (5.2) and (5.1) were used.

Corollary 5.1.2.1

$$V_{\pi'}(s) = V_\pi(s) + \mathbb{E}_{\pi'}\left[\sum_{t=0} \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) A_\pi(S_t, A_t) \mid S_0 = s\right] \tag{5.3}$$

Proof:

$$\begin{aligned}
&\mathbb{E}_{\pi'}\left[\sum_{t=0} \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) A_\pi(S_t, A_t) \mid S_0 = s\right] \\
&= \sum_{t=0} \sum_{s' \in \mathcal{X}} P_\pi[S_t = s' \mid S_0 = s] \gamma^t \sum_{a \in \mathcal{A}} (\pi'(a \mid s) - \pi(a \mid s)) A_\pi(s', a) \\
&= \sum_{t=0} \sum_{s' \in \mathcal{X}} P_\pi[S_t = s' \mid S_0 = s] \gamma^t \sum_{a \in \mathcal{A}} (\pi'(a \mid s) - \pi(a \mid s)) Q_\pi(s', a) - V_\pi(s') \sum_{a \in \mathcal{A}} \pi'(a \mid s) - \pi(a \mid s) \\
&= \sum_{t=0} \sum_{s' \in \mathcal{X}} P_\pi[S_t = s' \mid S_0 = s] \gamma^t \sum_{a \in \mathcal{A}} (\pi'(a \mid s) - \pi(a \mid s)) Q_\pi(s', a) - V_\pi(s')\ 0 \\
&= \mathbb{E}_{\pi'}\left[\sum_{t=0} \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) Q_\pi(S_t, A_t) \mid S_0 = s\right]
\end{aligned}$$

5.2 First estimator

We define $H(s) = \mathbb{E}_\pi \left[(\frac{\pi'(A \mid S)}{\pi(A \mid S)} - 1)\ A_\pi(S, A) \mid S = s \right]$.

Corollary 5.2.0.1 *Let $s \in \mathcal{X}$. The following equality holds:*

$$\mathbb{E}_{\pi'} \left[\sum_{t=0} \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) A_\pi(S_t, A_t) \mid S_0 = s \right] = \mathbb{E}_{\pi'} \left[\sum_{t=0} \gamma^t H(s) \mid S_0 = s \right] \tag{5.4}$$

Proof:

$$\begin{aligned}
&\mathbb{E}_{\pi'} \left[\sum_{t=0} \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) A_\pi(S_t, A_t) \mid S_0 = s \right] \\
&= \sum_{t=0} \sum_{s' \in \mathcal{X}} P_\pi[S_t = s' \mid S_0 = s] \gamma^t \sum_{a \in \mathcal{A}} (\pi'(a \mid s) - \pi(a \mid s)) A_\pi(s', a) \\
&= \sum_{t=0} \sum_{s' \in \mathcal{X}} P_\pi[S_t = s' \mid S_0 = s] \gamma^t \sum_{a \in \mathcal{A}} \pi(a \mid s) (\frac{\pi'(a \mid s)}{\pi(a \mid s)} - 1) A_\pi(s', a) \\
&= \sum_{t=0} \sum_{s' \in \mathcal{X}} P_\pi[S_t = s' \mid S_0 = s] \gamma^t \mathbb{E}_\pi \left[(\frac{\pi'(a \mid s)}{\pi(a \mid s)} - 1)\ A_\pi(S, A) \mid S = s' \right] \\
&= \mathbb{E}_{\pi'} \left[\sum_{t=0} \gamma^t H(S_t) \mid S_0 = s \right]
\end{aligned}$$

Because we use trajectories sampled using π, we replace $\mathbb{E}_{\pi'} \left[\sum_{t=0} \gamma^t H(s) \mid S_0 = s \right]$ by $\mathbb{E}_\pi \left[\sum_{t=0} \gamma^t H(s) \mid S_0 = s \right]$. If $\pi = \pi'$, both expected values equal 0. Similar to the trust region paper [24], we investigate the difference between the true value and the approximation.

Definition 5.2.1 *(π', π) is an α-coupled policy pair if it defines a joint distribution $P(a, a' \mid s)$, such that $P(a \neq a' \mid s) \leq \alpha$ for all $s \in \mathcal{X}$. π and π' will denote the marginal distributions of a and a' respectively.*

Theorem 5.2.1 *Let $\alpha = \max_s D_{TV}(\pi'(a \mid .), \pi(a \mid .))$,*
$\epsilon = \max_{s,a} |(\frac{\pi'(s \mid a)}{\pi(s \mid a)} - 1) A_\pi(s, a)|$ and $\gamma \in [0, 1)$

$$V_{\pi'}(s) \geq V_\pi(s) + \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t (\frac{\pi'(A_t \mid S_t)}{\pi(A_t \mid S_t)} - 1)\ A_\pi(S_t, A_t) \mid S_0 = s \right] - \frac{2\alpha\gamma\epsilon}{(1-\gamma)^2} \tag{5.5}$$

Proof:
Instead of sampling trajectories using π' and π independently, we sample a pair of trajectories at the same time using the α coupled policies (π', π):

$$\mathbb{E}_{\pi'}\left[\sum_{t=0}\gamma^t H(S_t)\right] - \mathbb{E}_{\pi}\left[\sum_{t=0}\gamma^t H(S_t)\right] = \mathbb{E}_{(\pi',\pi)}\left[\sum_{t=0}\gamma^t (H(S'_t) - H(S_t))\right]$$

Because the trajectories can diverge, let S'_t and S_t respectively denote the states at time T of the first and second trajectory. The first trajectory is generated using π', while the second trajectory is generated using π.

$$P_{(\pi',\pi)}[S'_t = S_t] \leq (1-\alpha)^t \qquad P_{(\pi',\pi)}[S'_t \neq S_t] \geq 1 - (1-\alpha)^t$$

For two trajectories to agree on a state s_t, they must either agree on all actions up to time t with probability $(1-\alpha)^t$, or they can diverge at some point but still end up at the same state s_t.

$$\begin{aligned}
\mathbb{E}_{(\pi',\pi)}\left[\gamma^t (H(S'_t) - H(S_t))\right] &= \mathbb{E}_{(\pi',\pi)}\left[\gamma^t (H(S'_t) - H(S_t)) \mid S'_t = S_t\right]\ P[S'_t = S_t] \\
&\quad + \mathbb{E}_{(\pi',\pi)}\left[\gamma^t (H(S'_t) - H(S_t)) \mid S'_t \neq S_t\right]\ P[S'_t \neq S_t] \\
&= \mathbb{E}_{(\pi',\pi)}\left[\gamma^t (H(S'_t) - H(S_t)) \mid S'_t \neq S_t\right]\ P[S'_t \neq S_t] \\
&\geq \mathbb{E}_{(\pi',\pi)}\left[\gamma^t (H(S'_t) - H(S_t)) \mid S'_t \neq S_t\right] (1 - (1-\alpha)^t) \\
&\geq -2\gamma^t \max_{s,a} |(\frac{\pi'(a \mid s)}{\pi(a \mid s)} - 1) A(s,a)|\ (1 - (1-\alpha)^t) \\
&= -2\gamma^t \epsilon\ (1 - (1-\alpha)^t)
\end{aligned}$$

Therefore, we can conclude:

$$\begin{aligned}
\mathbb{E}_{\pi'}\left[\sum_{t=0}\gamma^t H(S_t)\right] - \mathbb{E}_{\pi}\left[\sum_{t=0}\gamma^t H(S_t)\right] &= \mathbb{E}_{(\pi',\pi)}\left[\sum_{t=0}\gamma^t (H(S'_t) - H(S_t))\right] \\
&= \sum_{t=0}\mathbb{E}_{(\pi',\pi)}\left[\gamma^t (H(S'_t) - H(S_t))\right] \\
&\geq \sum_{t=0} -2\gamma^t \epsilon\ (1 - (1-\alpha)^t) \\
&\geq -2\gamma\epsilon \frac{1}{1-\gamma}\ \frac{1}{1-\gamma(1-\alpha)} \\
&= -2\gamma\epsilon\ \frac{\alpha\gamma}{(1-\gamma)(1-\gamma(1-\alpha))} \\
&\geq -\frac{2\gamma^2\alpha\ \epsilon}{(1-\gamma)^2}
\end{aligned}$$

Using the same arguments from [24], if $\max_s D_{TV}(\pi'(a \mid .), \pi(a \mid .)) \leq \alpha$, we can define an α-coupled policy pair (π', π).

Equation (5.5) suggests that in addition to constraining $\max_s D_{TV}(\pi'(a \mid .), \pi(a \mid .))$, it is also important to constrain the relative distance, $\frac{\pi'(a \mid s) - \pi(a \mid s)}{\pi'(a \mid s)}$. The proximal policy optimization paper [24] addresses this point by clipping the ratio $\frac{\pi'(a \mid s)}{\pi'(a \mid s)}$ to be close to 1, which constrains $\frac{\pi'(a \mid s)}{\pi'(a \mid s)} - 1$. This serves as a sanity check for (5.5).
Therefore, to maximize $V_{\pi'}(s_0)$, we maximize $\mathbb{E}_\pi \left[\sum_{t=0} \gamma^t (\frac{\pi'(A_t \mid S_t)}{\pi(A_t \mid S_t)} - 1)\ A_\pi(S_t, A_t) \mid S_0 = s \right]$ while ensuring that $\frac{2\alpha\gamma\epsilon}{(1-\gamma)^2}$ is close to zero.

$$\begin{aligned} &\nabla \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t (\frac{\pi'(A_t \mid S_t)}{\pi(A_t \mid S_t)} - 1)\ A_\pi(S_t, A_t) \mid S_0 = s \right] \\ &= \nabla \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t \frac{\pi'(A_t \mid S_t)}{\pi(A_t \mid S_t)}\ A_\pi(S_t, A_t) \mid S_0 = s \right] \end{aligned}$$

We use the update rule from PPO [23] because the gradient is the same, and the constraints are similar to [24].

When dealing with V^c, there is no discount factor. Therefore, a restriction to the episodic case is necessary. Let T be the episode length. In this scenario, equation (5.4) can be written as follows:

$$V^c_{\pi'}(s) \geq V^c_\pi(s) + \mathbb{E}_\pi \left[\sum_{t=0} (\frac{\pi'(A_t \mid S_t)}{\pi(A_t \mid S_t)} - 1)\ A^c_\pi(S_t, A_t) \mid S_0 = s \right] - 2\epsilon(T - \frac{1 - (1-\alpha)^T}{\alpha})$$

5.3 Second estimator

Instead of evaluating $\mathbb{E}_{\pi'} \left[\sum_{t=0} \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) A_\pi(S_t, A_t) \right]$, we can use trajectories generated by π: $\mathbb{E}_\pi \left[\sum_{t=0} \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) A_\pi(S_t, A_t) \right]$.

Theorem 5.3.1 *Let* $\alpha = \max_s D_{TV}(\pi'(a \mid .), \pi(a \mid .))$, $\epsilon' = \max_{s,a} |(1 - \frac{\pi(s \mid a)}{\pi'(s \mid a)}) A_\pi(s, a)|$ *and* $\gamma \in [0, 1)$

$$V_{\pi'}(s) \geq V_\pi(s) + \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) A_\pi(S_t, A_t) \mid S_0 = s \right] - \frac{2\alpha\gamma\epsilon'}{(1-\gamma)^2} \tag{5.6}$$

Proof:
For notational convenience, we define $H'(s, a) = (1 - \frac{\pi(a \mid s)}{\pi'(a \mid s)}) A_\pi(s, a)$.
We use the same principle and notation from the proof (5.5). In addition, let A'_t and A_t be the action chosen by the α-coupled policies (π', π) at timestep t. We denote the event D by $\{S'_t \neq S_t \vee A'_t \neq A_t\}$

$$\begin{cases} P_{(\pi',\pi)}[S'_t = S_t] \leq (1-\alpha)^t \\ P_{(\pi',\pi)}[A'_t = A_t \mid S'_t = S_t] = 1-\alpha \end{cases} \Rightarrow P_{(\pi',\pi)}[S'_t = S_t, A'_t = A_t] \leq (1-\alpha)^{t+1}$$

$$\Rightarrow P_{(\pi',\pi)}[S'_t \neq S_t \vee A'_t \neq A_t] \geq 1-(1-\alpha)^{t+1}$$

$$\Rightarrow P_{(\pi',\pi)}[D] \geq 1-(1-\alpha)^{t+1}$$

If both policies agree on the state and action at time t, the expected value is 0:

$$\mathbb{E}_{(\pi',\pi)}\left[\gamma^t(H'(S'_t,A'_t) - H'(S_t,A_t)) \mid S'_t = S_t, A'_t = A_t\right] = 0$$

Otherwise:

$$\mathbb{E}_{(\pi',\pi)}\left[\gamma^t(H'(S'_t,A'_t) - H'(S_t,A_t)) \mid D\right]\ P_{(\pi',\pi)}[D] \geq -2\gamma^t\epsilon'(1-(1-\alpha)^{t+1})$$

Combining the equations, we deduce the following:

$$\begin{aligned}
\mathbb{E}_{\pi'}\left[\sum_{t=0}\gamma^t H'(S_t,A_t)\right] - \mathbb{E}_{\pi}\left[\sum_{t=0}\gamma^t H'(S_t,A_t)\right] &= \mathbb{E}_{(\pi',\pi)}\left[\sum_{t=0}\gamma^t H'(S'_t,A'_t) - H'(S_t,A_t)\right] \\
&= \sum_{t=0}\mathbb{E}_{(\pi',\pi)}\left[\gamma^t H'(S'_t,A'_t) - H'(S_t,A_t)\right] \\
&\geq \sum_{t=0} -2\gamma^t\epsilon'\ (1-(1-\alpha)^t+1) \\
&\geq -2\gamma\epsilon'\frac{1}{1-\gamma}\ \frac{1}{1-\gamma(1-\alpha)} \\
&= -2\gamma\epsilon'\ \frac{\alpha\gamma}{(1-\gamma)(1-\gamma(1-\alpha))} \\
&\geq -\frac{2\gamma^2\alpha\ \epsilon'}{(1-\gamma)^2}
\end{aligned}$$

5.4 Analyzing the difference

We denote π_θ a parameterized policy with parameter vector θ. We defined $\mathcal{L}$ and $\mathcal{L}'$ as follows:

$$\mathcal{L}_1(\theta,\theta_0,A_{\pi_{\theta_0}}) = \mathbb{E}_{\pi_{\theta_0}}\left[\sum_{t=0}\gamma^t(\frac{\pi_\theta(A_t \mid S_t)}{\pi_{\theta_0}(A_t \mid S_t)} - 1)A_{\pi_{\theta_0}}(S_t,A_t) \mid S_0 = s\right] \quad (5.7)$$

$$\mathcal{L}_2(\theta,\theta_0,A_{\pi_{\theta_0}}) = \mathbb{E}_{\pi_{\theta_0}}\left[\sum_{t=0}\gamma^t(1 - \frac{\pi_{\theta_0}(A_t \mid S_t)}{\pi_\theta(A_t \mid S_t)})\ A_{\pi_{\theta_0}}(S_t,A_t) \mid S_0 = s\right] \quad (5.8)$$

Differentiating with respect to θ:

$$\nabla_\theta \mathcal{L}_1(\theta, \theta_0, A_{\pi_{\theta_0}}) = \mathbb{E}_{\pi_{\theta_0}} \left[\sum_{t=0} \gamma^t \nabla \pi_\theta(A_t \mid S_t) \frac{1}{\pi_{\theta_0}(A_t \mid S_t)} A_{\pi_{\theta_0}}(S_t, A_t) \mid S_0 = s \right] \tag{5.9}$$

$$\nabla_\theta \mathcal{L}_2(\theta, \theta_0, A_{\pi_{\theta_0}}) = \mathbb{E}_{\pi_{\theta_0}} \left[\sum_{t=0} \gamma^t \nabla \pi_\theta(A_t \mid S_t) \frac{\pi_{\theta_0}(A_t \mid S_t)}{\pi_\theta(A_t \mid S_t)^2} A_{\pi_{\theta_0}}(S_t, A_t) \mid S_0 = s \right] \tag{5.10}$$

To compare $\mathcal{L}_1$ and $\mathcal{L}_2$, we plot the ratios $r = \frac{1}{\pi_{\theta_0}}$ and $r' = \frac{\pi_{\theta_0}}{\pi_\theta^2}$ with different values of $\pi_{\theta_0} = 0.2, 0.5, 0.8$

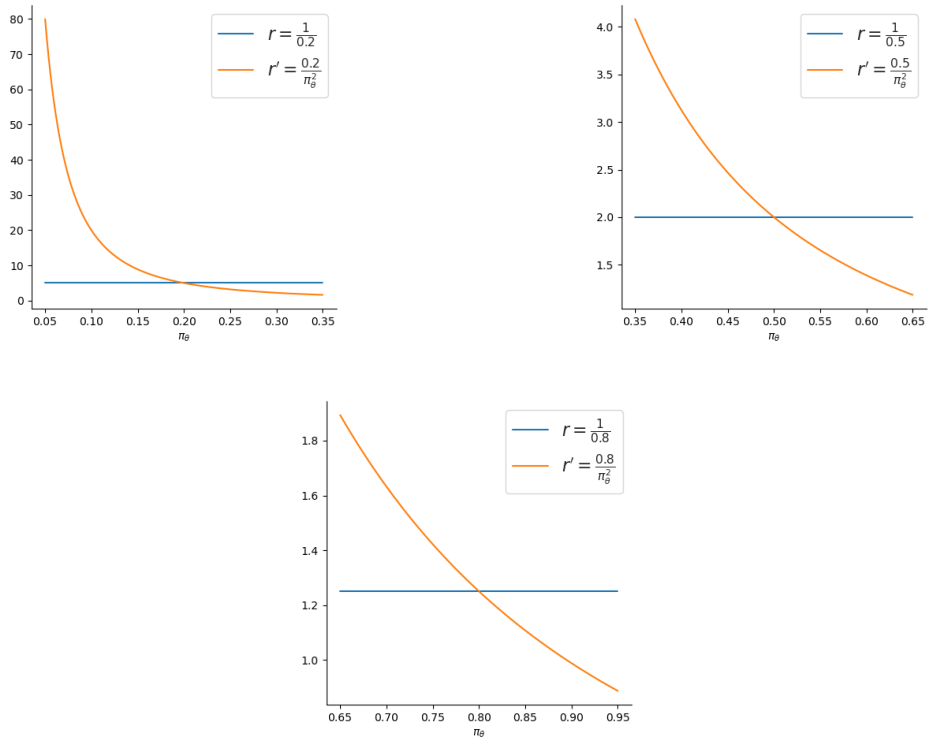


Figure 5.1: Comparing $\mathcal{L}_1$ and $\mathcal{L}_2$ by plotting $r = \frac{1}{\pi_{\theta_0}}$ and $r' = \frac{\pi_{\theta_0}}{\pi_\theta^2}$ with different values of $\pi_{\theta_0} = 0.2, 0.5, 0.8$

6 Safe policy iteration

In this chapter, we use $Q^c(s,a)$ to cancel unsafe rewards. We gradually modify the value function $V_\pi(s_0) = [\sum_{t=0} r(S_t, A_t) \mid S_0 = s_0]$ to cancel unsafe returns and give motivating examples. We assume that the reward function is strictly positive: $\forall s \in \mathcal{X}\ \forall a \in \mathcal{A} \quad r(s,a) > 0$

6.1 Canceling unsafe reward

Let $\tau = (s_0, a_0, s1, a_1.., a_{T-1}, s_{T-1})$ be a trajectory and $R(\tau) = \sum_t \gamma^t r(s_t, a_t)$. We defined $V^r : \mathcal{X}, \mathcal{A} \to \mathbb{R}$ as follows:

$$V^r_\pi(s) = \sum_\tau P_\pi[\tau \mid S_0 = s]\ R(\tau)\ f(\tau)$$

If a trajectory τ is unsafe, then $R(\tau)\ f(\tau) = 0$. Therefore, $\nabla P_\pi[\tau]\ R(\tau)\ f(\tau) = 0$. In other words, we don't increase the probability of generating τ using π when we follow ∇V^r_π. However, if $f(\tau) = 1$, we increase the probability of generating τ proportionally with $R(\tau)$.

$$V^r_\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t r(S_t, A_t)\ Q^c_\pi(S_t, A_t) \mid S_0 = s \right] \tag{6.1}$$

Proof:

$$
\begin{aligned}
V_\pi^r(s) &= \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t r(S_t, A_t) \prod_{t=0} f(S_t) \mid S_0 = s \right] \\
&= \sum_{a \in \mathcal{A}} \pi(a \mid s) E_\pi \left[r(s,a) \prod_{t=0} f(S_t) \mid S_0 = s, A_0 = a \right] \\
&\qquad + \mathbb{E}_\pi \left[\sum_{t=1} \gamma^t r(S_t, A_t) \prod_{t=0} f(S_t) \mid S_0 = s, A_0 = a \right] \\
&= \sum_{a \in \mathcal{A}} \pi(a \mid s)\ r(s,a)\ Q_\pi^c(s,a) + \sum_{s' \in \mathcal{X}} p(s' \mid s,a) f(s)\, \mathbb{E}_\pi \left[\sum_{t=1} \gamma^t r(S_t, A_t) \prod_{t=1} f(S_t) \mid S_1 = s' \right] \\
&= \sum_{a \in \mathcal{A}} \pi(a \mid s)\ r(s,a)\ Q_\pi^c(s,a) + \sum_{s' \in \mathcal{X}} p(s' \mid s,a)\, \mathbb{E}_\pi \left[\sum_{t=1} \gamma^t r(S_t, A_t) \prod_{t=1} f(S_t) \mid S_1 = s' \right] \\
&= \sum_{a \in \mathcal{A}} \pi(a \mid s)\ r(s,a)\ Q_\pi^c(s,a) + \sum_{s' \in \mathcal{X}} p(s' \mid s,a) \gamma\, \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t r(S_t, A_t) \prod_{t=0} f(S_t) \mid S_0 = s' \right] \\
&= \sum_{a \in \mathcal{A}} \pi(a \mid s)\ r(s,a)\ Q_\pi^c(s,a) + \sum_{s' \in \mathcal{X}} p(s' \mid s,a) \gamma V_\pi^r(s') \\
&= \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t r(S_t, A_t)\ Q_\pi^c(S_t, A_t) \mid S_0 = s \right]
\end{aligned}
$$

We follow the standard definition for $Q_\pi^r : \mathcal{X}, \mathcal{A} \to \mathbb{R}$ and $A_\pi^r : \mathcal{X}, \mathcal{A} \to \mathbb{R}$:

$$
\begin{aligned}
Q_\pi^r(s,a) &= \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t r(S_t, A_t) Q_\pi^c(S_t, A_t) \mid S_0 = s, A_0 = a \right] \\
A_\pi^r(s,a) &= Q_\pi^r(s,a) - V_\pi^r(s)
\end{aligned}
$$

V_π^r can be viewed as a value function with reward $r'(s,a) = r(s,a)\ Q_\pi^c(s,a)$. We note that $V_\pi(s_0) \geq V_\pi^r(s_0)$ and the equality holds if $V_\pi^c(s_0) = 1$. Let's consider the 3 state CMDP 6.1, where the maximum cumulative cost $c_0 = 1$.

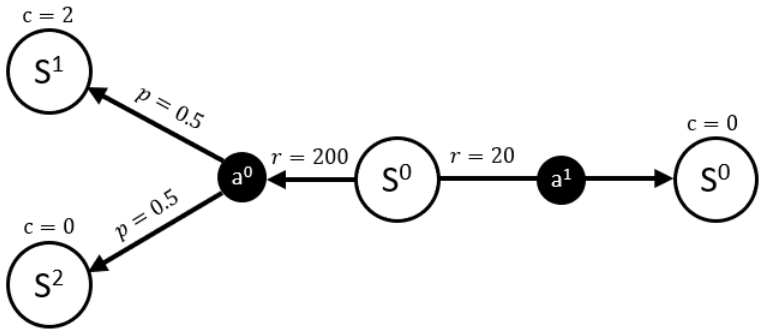


Figure 6.1: CMDP where optimizing w.r.t V^r leads to an unsafe policy

If the agent chooses action a^0 from the initial state s^0, it transitions to s^1 or s^2 with the following transition probability: $p(s^1 \mid a^0, s^0) = 0.5$ and $p(s^2 \mid a^0, s^0) = 0.5$. The agent chooses both a^0 or a^1 with probability 0.5

$$\begin{aligned} Q^c_\pi(s^0, a^0) &= 0.5 \qquad & A^r_\pi(s^0, a^0) &= 40 \\ Q^c_\pi(s^0, a^1) &= 1 \qquad & A^r_\pi(s^0, a^1) &= -40 \end{aligned}$$

Because $Q_\pi(s^0, a^0) \gg Q_\pi(s^0, a^1)$, updating the policy can increase the probability of choosing a^0. However, the reward obtained from choosing a^0 should be canceled because $Q^c_\pi(s^0, a^0) = 0.5 < 1$; picking action a^0 violates the safety constraint with probability 0.5. Therefore, it is reasonable to cancel the reward of actions that violate the safety constraints. We can formalize this property as follows:

$$V^{r,k}_\pi = \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t r(S_t, A_t) \; (Q^c_\pi(S_t, A_t))^k \mid S_0 = s \right]$$

Taking the limit of k as it approaches infinity:

$$\begin{aligned} \lim_{k \to \infty} V^{r,k}_\pi &= \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t r(S_t, A_t) \; ||Q^c_\pi(S_t, A_t))||_\infty \mid S_0 = s \right] \\ &= \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t r(S_t, A_t) \; \mathbb{I}[Q^c_\pi(S_t, A_t) = 1] \mid S_0 = s \right] \end{aligned}$$

Because $Q^c_\pi \in [0, 1]$, following holds:

$$V^{r,0}_\pi = V_\pi \geq V^{r,1}_\pi \geq V^{r,2}_\pi .... \geq V^{r,\infty}_\pi$$

The safe reinforcement learning problem we are solving is:

$$\max_\pi V_\pi(s_0) \qquad \text{s.t } V^c_\pi(s_0) = 1$$

Let π^* a policy that satisfies the safe RL problem. Because $V^c_{\pi^*}(s_0) = 1$, we can deduce the following:

$$\forall s \in \mathcal{X}\ \forall a \in \mathcal{A} \quad P_{\pi^*}[S_t = s \mid S_0 = s_0] > 0 \Rightarrow \pi^*(a \mid s) = 0 \ \vee \ Q^c_{\pi^*}(s,a) = 1$$

Starting from state s_0, all actions taken by the agent are safe. Based on the previous property, we can deduce the following:

$$V_{\pi^*}(s_0) = V^{r,\infty}_{\pi^*}(s_0)$$

Proof:

$$\begin{aligned}
V_{\pi^*}(s_0) &= \sum_t \sum_{s\in\mathcal{X}} P_{\pi^*}[S_t = s \mid S_0 = s_0] \sum_{a\in\mathcal{A}} \pi^*(a \mid s) r(a,s) \\
&= \sum_t \sum_{s\in\mathcal{X}} P_{\pi^*}[S_t = s \mid S_0 = s_0] \sum_{a\in\mathcal{A}} \pi^*(a \mid s) r(a,s)\ \mathbb{I}[\pi^*(a \mid s) \neq 0] + r(s,a)\ \mathbb{I}[\pi^*(a \mid s) = 0] \\
&= \sum_t \sum_{s\in\mathcal{X}} P_{\pi^*}[S_t = s \mid S_0 = s_0] \sum_{a\in\mathcal{A}} \pi^*(a \mid s) r(a,s)\ \mathbb{I}[\pi^*(a \mid s) \neq 0] \\
&= \sum_t \sum_{s\in\mathcal{X}} P_{\pi^*}[S_t = s \mid S_0 = s_0] \sum_{a\in\mathcal{A}} \pi^*(a \mid s) r(a,s) ||Q^c_{\pi^*}(s,a)||_\infty \\
&= V^{r,\infty}_{\pi^*}(s_0)
\end{aligned}$$

We define $A^{r,k}_\pi(s,a) = Q^{r,k}_\pi(s,a) - V^{r,k}_\pi(s,a)$.
Computing different values of $A^{r,k}_\pi(s,a)$ of the CMDP 6.1:

$$\begin{aligned}
A^{r,0}_\pi(s^0,a^0) &= 90 & A^{r,0}_\pi(s^0,a^1) &= -90 \\
A^{r,1}_\pi(s^0,a^0) &= 40 & A^{r,1}_\pi(s^0,a^1) &= -40 \\
A^{r,4}_\pi(s^0,a^0) &= -3.75 & A^{r,0}_\pi(s^0,a^1) &= 3.75 \\
A^{r,8}_\pi(s^0,a^0) &= -9.6 & A^{r,8}_\pi(s^0,a^1) &= 9.6 \\
A^{r,\infty}_\pi(s^0,a^0) &= -10 & A^{r,\infty}_\pi(s^0,a^1) &= 10
\end{aligned}$$

As demonstrated by this example, it is not always necessary to choose $k \to \infty$. Instead, choosing $k = 4$ solves this toy example.
We use $\tilde{V}^{r,k}_{\pi'} = \mathbb{E}_{\pi'}\left[\sum_{t=0} \gamma^t r(S_t,A_t) Q^c_\pi(S_t,A_t)^k \mid S_0 = s_0\right]$ as a first order approximation of $V^{r,k}_{\pi'}$ when $\pi \approx \pi'$.

Theorem 6.1.1

$$\tilde{V}^{r,k}_{\pi'}(s_0) = V^{r,k}_\pi(s_0) + \mathbb{E}_{\pi'}\left[\sum_{t=0} \gamma^t\ A^{r,k}_\pi(S_t,A_t) \mid S_0 = s_0\right] \tag{6.2}$$

Proof:

$$
\begin{aligned}
(\bar{V}^{r,k}_{\pi'} - V^{r,k}_{\pi})(s) &= \sum_{a\in\mathcal{A}} \pi'(a \mid s)\ \bar{Q}^{r,k}_{\pi'}(s,a) - \sum_{a\in\mathcal{A}} \pi(a \mid s)\ Q^{r,k}_{\pi}(s,a) \\
&= \sum_{a\in\mathcal{A}} \pi'(a \mid s)\ \bar{Q}^{r,k}_{\pi'}(s,a) - \sum_{a\in\mathcal{A}} \pi(a \mid s)\ Q^{r,k}_{\pi}(s,a) \pm \sum_{a\in\mathbf{A}} \pi'(a \mid s)\ \bar{Q}^{r,k}_{\pi}(s,a) \\
&= \sum_{a\in\mathcal{A}} \pi'(a \mid s)\ (\bar{Q}^{r,k}_{\pi'}(s,a) - Q^{r,k}_{\pi}(a,s)) + \sum_{a\in\mathcal{A}} (\pi'(a \mid s) - \pi(a \mid s))\ Q^{r,k}_{\pi}(s,a) \\
&= \sum_{a\in\mathcal{A}} \sum_{s'\in\mathcal{X}} \pi'(a \mid s) p(s' \mid s,a) \gamma (\bar{V}^{r,k}_{\pi'} - V^{r,k}_{\pi})(s') + \sum_{a\in\mathcal{A}} \pi'(s,a)(1 - \frac{\pi(a \mid s)}{\pi'(a \mid s)}) Q^{r,k}_{\pi}(s,a) \\
&= \mathbb{E}_{\pi'} \left[\sum_t \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) Q^{r,k}_{\pi}(S_t, A_t) \mid S_0 = s \right]
\end{aligned}
$$

We omit the proof of the following equation because it is similar to (5.3).

$$
\mathbb{E}_{\pi'} \left[\sum_t \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) Q^{r,k}_{\pi}(S_t, A_t) \right] = \mathbb{E}_{\pi'} \left[\sum_t \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) A^{r,k}_{\pi}(S_t, A_t) \right]
$$

Theorem 6.1.2 *Let* $\alpha = \max_s D_{TV}(\pi'(a \mid .), \pi(a \mid .))$,
$\epsilon = \max_{s,a} |(\frac{\pi'(s \mid a)}{\pi(s \mid a)} - 1) A^{r,k}_{\pi}(s,a)|$, $\epsilon' = \max_{s,a} |(1 - \frac{\pi(s \mid a)}{\pi'(s \mid a)}) A^{r,k}_{\pi}(s,a)|$ *and*
$\gamma \in [0,1)$

$$
\bar{V}^{r,k}_{\pi'}(s) \geq V^{r,k}_{\pi}(s) + \mathbb{E}_{\pi} \left[\sum_{t=0} \gamma^t (\frac{\pi'(A_t \mid S_t)}{\pi(A_t \mid S_t)} - 1)\ A^{r,k}_{\pi}(S_t, A_t) \mid S_0 = s \right] - \frac{2\alpha\gamma\epsilon}{(1-\gamma)^2} \tag{6.3}
$$

$$
\bar{V}^{r,k}_{\pi'}(s) \geq V^{r,k}_{\pi}(s) + \mathbb{E}_{\pi} \left[\sum_{t=0} \gamma^t (1 - \frac{\pi(A_t \mid S_t)}{\pi'(A_t \mid S_t)}) A^{r,k}_{\pi}(S_t, A_t) \mid S_0 = s \right] - \frac{2\alpha\gamma\epsilon'}{(1-\gamma)^2} \tag{6.4}
$$

The proof is omitted due to the similarly with (5.5) and (5.6)

We also note that choosing an appropriate value of k can solve problems where $V^*_{\pi}(s_0) < 1$. To illustrate this, consider the following CMDP:

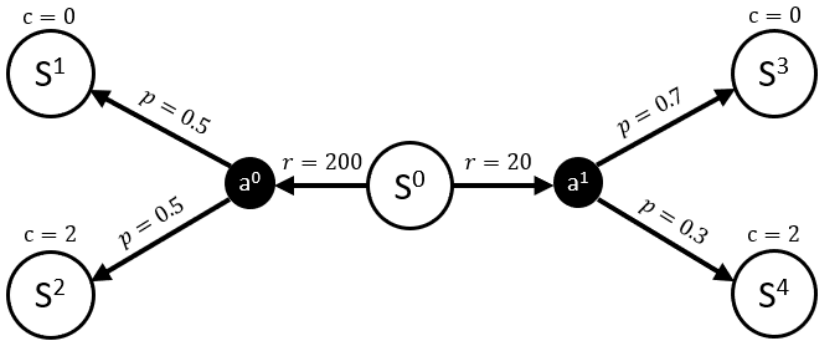


Figure 6.2: CMDP where all policies are not totally safe, e.g $V_\pi^c(s^0) < 1$

The maximum achievable safety is $V_{\pi^*}^c(s^0) = 0.7$. We compute $A_\pi^{r,k}(s,a)$ with different k values:

$$
\begin{aligned}
A_\pi^{r,0}(s^0,a^0) &= 90 & A_\pi^{r,0}(s^0,a^1) &= -90 \\
A_\pi^{r,1}(s^0,a^0) &= 43 & A_\pi^{r,1}(s^0,a^1) &= -43 \\
A_\pi^{r,4}(s^0,a^0) &= 3.84 & A_\pi^{r,0}(s^0,a^1) &= -3.84 \\
A_\pi^{r,8}(s^0,a^0) &= -0.18 & A_\pi^{r,8}(s^0,a^1) &= 0.18 \\
A_\pi^{r,\infty}(s^0,a^0) &= 0 & A_\pi^{r,\infty}(s^0,a^1) &= 0
\end{aligned}
$$

Because $V_{\pi^*}(s^0) < 0$, $A_\pi^{r,\infty}(s,a) = 0$. Choosing $k = 8$ achieves the best trade-off between safety and return maximization.

6.2 Reducing cost

The safety metric $V^c(s)$ reflects the probability of visiting safe states starting from s. Let a^1 and a^2 be two actions such that $Q^c(s^0,a^1) = Q^c(s^0,a^2) = 0.9$. Even though both actions are equally safe, it is possible that $E_\pi[\sum_t c(S_t) \mid S_0 = s, A_0 = a_1] \gg E[\sum_t c(S_t) \mid S_0 = s, A_0 = a_2]$. Therefore, it is desirable to favour a^2 over a^1.
The following CMDP illustrates this problem:

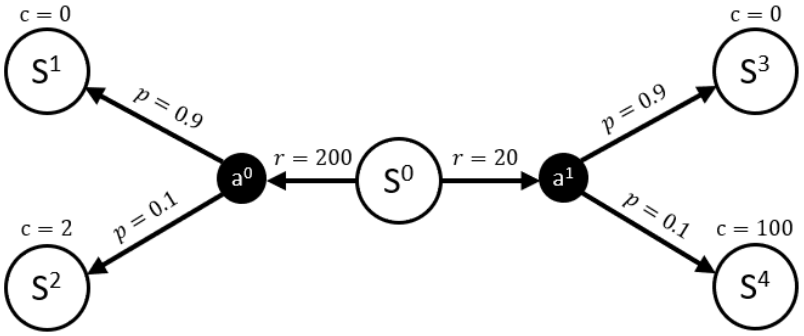


Figure 6.3: Using the objective function $V^{r,k}$ leads to multiple solutions; Actions a^0 and a^1 are equally favoured. The optimal solution should also minimize the cumulative cost. This CMDP motivates the introduction of a cost term to $V^{r,k}$.

Because $\forall k \geq 0 \; A_\pi^{r,k}(s^0, a^1) = A_\pi^{r,k}(s^0, a^2)$, our objective function $V_\pi^{r,k}(s^0)$ equally favors both actions. This motivates the introduction of a new term to $V^{r,k}$:

Definition 6.2.1 *Let* $\beta > 0$

$$V_\pi^{rc,k}(s) = \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t (r(S_t, A_t)\; Q_\pi^c(S_t, A_t)^k - \beta\; (1 - Q_\pi^c(S_t, A_t)^k)\; c(S_t)) \mid S_0 = s \right] \tag{6.5}$$

If $V_\pi^c(s_0) = 1$, then $V_\pi(s_0) = V^{r,k}(s_0) = V^{rc,k}(s_0)$. Therefore, the maximum achievable safe return does not change.
Moreover, $\mathbb{E}_\pi \left[\sum_{t=0} \gamma^t (1 - Q_\pi^c(S_t, A_t)^k)\; c(S_t) \mid S_0 = s, A_t = a \right]$ allows us to distinguish between unsafe actions based on how much cumulative cost they incur. This facilitates avoiding extremely hazardous behavior.
Using $V^{rc,k}$ is also useful when the randomly initialized policy is entirely unsafe, e.g. $Q_\pi^c(s, a) = 0$ for all state-action pairs. In such scenario, using $V^{rc,k}$ is equivalent to minimizing the cumulative cost:

$$\forall a \in \mathcal{A}\; \forall s \in \mathcal{X} \quad Q_\pi^c(s, a) = 0 \Rightarrow V_\pi^{rc,k}(s_0) = -\,\mathbb{E}_\pi \left[\sum_{t=0} \gamma^t c(S_t) \mid S_0 = s_0 \right]$$

When all generated trajectories are unsafe, minimizing the cumulative cost is necessary to find a safe policy eventually. We define the Q function and

advantage with respect to $V^{rc,k}$ as follows:

$$Q_\pi^{rc,k}(s,a) = \mathbb{E}_\pi \left[\sum_{t=0} \gamma^t (r(S_t, A_t)\ Q_\pi^c(S_t, A_t)^k - \beta\ (1 - Q_\pi^c(S_t, A_t)^k)\ c(S_t)) \mid S_0 = sA_0 = a \right]$$
$$A_\pi^{rc,k}(s,a) = Q_\pi^{rc,k}(s,a) - V_\pi^{rc,k}(s)$$

6.3 Generalized advantage estimation

We can leverage the findings of the GAE paper to estimate $A_\pi^{rc,k}$ by proving shared properties between V and $V^{rc,k}$. We use the same notation and definitions in [22]. Let $r'_t = r_t Q_t^c - \beta c_t$ and $\delta_t^V = r'_t + \gamma V_{t+1}^{rc,k} - V_t^{rc,k}$.

$$\begin{aligned}
\hat{A}_t^{(1)} &= \delta_t^V && = -V_t^{rc,k} + r'_t + \gamma V_{t+1}^{rc,k} \\
\hat{A}_t^{(2)} &= \delta_t^V + \gamma\delta_{t+1}^V && = -V_t^{rc,k} + r'_t + \gamma r'_{t+1} + \gamma^2 V_{t+2}^{rc,k} \\
\hat{A}_t^{(3)} &= \delta_t^V + \gamma\delta_{t+1}^V + \gamma^2\delta_{t+2}^V && = -V_t^{rc,k} + r'_t + \gamma r'_{t+1} + \gamma^2 r'_{t+2} + \gamma^{t+3} V_{t+3}^{rc,k}
\end{aligned}$$

$$\hat{A}_t^{(q)} \sum_{l=0}^{q-1} \gamma^l \delta_{t+l}^V = -V_t^{rc,k} + r'_t + \gamma r'_{t+1} + \gamma^2 r'_{t+1} + ... + \gamma^{q-1} r'_{t+q-1} + \gamma^q V_{t+q}^{rc,k}$$

Therefore, the following equation can be used to estimate $A_\pi^{rc,k}$:

$$\hat{A}^{\mathrm{GAE}(\gamma,\lambda)} = \sum_{l=0} (\gamma\lambda)^l \delta_{t+l}^V \tag{6.6}$$

7 Practical implementation

7.1 Safety critic policy optimization

After generating trajectories using π, we can compute $\hat{A}^{r,r}_{\pi}(s,a)$ and $\hat{A}^{c}_{\pi}(s,a)$ for every state-action pair.
We denote a parameterized policy by π_θ. We refer to the The initial policy by π_{θ_0}.
For notational convenience, we define $r_t(\theta) = \frac{\pi_\theta(A_t \mid S_t)}{\pi_{\theta_0}(A_t \mid S_t)} - 1$, $r'_t(\theta) = 1 - \frac{\pi_{\theta_0}(A_t \mid S_t)}{\pi_\theta(A_t \mid S_t)}$ and $A_t = A(S_t, A_t)$.
We follow the same clipping strategy in [23]. We define clipped version of $\mathcal{L}_1$ and $\mathcal{L}_2$:

$$\mathcal{L}_1^{\text{CLIP}}(\theta, \theta_0, A_{\pi_{\theta_0}}) = \mathbb{E}_{\pi_{\theta_0}}\left[\sum_{t=0} \min(r_t(\theta)A_t, \text{clip}(r_t(\theta), -\epsilon, +\epsilon)A_t) \mid S_0 = s\right]$$

$$\mathcal{L}_2^{\text{CLIP}}(\theta, \theta_0, A_{\pi_{\theta_0}}) = \mathbb{E}_{\pi_{\theta_0}}\left[\sum_{t=0} \min(r'_t(\theta)A_t, \text{clip}(r'_t(\theta), -\epsilon, +\epsilon)A_t) \mid S_0 = s\right]$$

Where ϵ is hyperparameter, usually chosen $\epsilon = 0.2$
To approximate π, V^c, and $V^{rc,k}$, we use three fully-connected MLPs with two hidden layers and tanh nonlinearities. The policy network outputs the mean of a Gaussian distribution with variable standard deviations for continuous tasks, as described in [24, 7]. When the reward is negative, a reward bias is added such that for all state s and action a, $r'(s,a) = r(s,a) + b > 0$.
The output of the safety critic MLP is tanh clipped to the range $[0, 1]$. This ensures the safety critic is pessimistic when randomly initialized: $V^c_{\pi_0}(s) \approx 0$.
In the following algorithm, we refer to $\mathcal{L} = \mathcal{L}_1$ or $\mathcal{L} = \mathcal{L}_2$. We start by a randomly initialized policy π_θ

Algorithm 1 Safety critic policy optimization (SCPO)

for iteration= $0, 1, 2, 3...$ **do**

Sample $(a_t, s_t) \sim \pi_{\theta_0}$ for T timesteps

Evaluate $\hat{Q}^c(s_t, a_t)$

$r(s,a) \leftarrow r(s,a) + b$ (positive reward, $b \geq 0$)

$r(s,a) \leftarrow r(s,a)\ \hat{Q}^c(s,a)^k - \beta(1 - Q^c(s,a)^k)c(s,a)$

Use GAE to estimate $\hat{A}^{rc,k}_{\pi_{\theta_0}}(s_t, a_t)$

for epoch= $0, 1, 2, 3...$ **do**

Optimize $\mathcal{L}^{\text{CLIP}}(\theta, \theta_0, A^{rc,k}_{\pi_{\theta_0}})$ wrt θ

Critic update: $V^{rc,k}$ and V^c

end for

$\theta_0 \leftarrow \theta$

end for

When $k = 0$ and $\beta > 0$, then β is a Lagrange multiplier for the constrained safe RL problem:

$$\max_{\pi} V_\pi(s_0) \qquad \text{s.t} \quad \mathbb{E}_\pi \left[\sum_{t=0} c(S_t) \mid S_0 = s_0 \right] \leq c_0$$

8 Experiments

In this chapter, we evaluate the performance of our algorithm (SCPO) by comparing it to TRPO-L [21], CPO [2], PDO, and PCPO [26]. We use the ball agent from safety bullet gym [8] on four tasks: circle, reach, gather, run. The experiment results demonstrate the effectiveness of our approach. We also highlight the importance of augmenting the state representation using the cumulative cost as described in 4.1.

8.1 Environments

We use a subset of environments from a free and open-source framework called Bullet-Safety-Gym [8].
The agent is a ball which can move freely on the xy-plane. The shape of the observations space is $\mathbb{R}^9$, which contains the position $x \in \mathbb{R}^3$ and the velocity $\dot{x} \in \mathbb{R}^3$.Actions are applied as forced $a \in [-1, 1]^2$.

SafetyBallCircle: The goal is to move clockwise without leaving the safe area.
Reward: The reward is maximized when the agent moves clockwise as fast as possible. $r(s) = \frac{v^T[-y,x]}{1+3|r_{\text{agent}}-r_{\text{circle}}|}$.
Cost: A cost of 1 is incurred when the agent is outside the bounds denoted by two vertical lines, e.g. $c(s) = \mathbb{1}[|x| \geq x_{\text{lim}}]$

SafetyBallGather: The Agent is spawned randomly and incentivized to collect blue balls and avoid collecting red ones.
Reward: The agent receives a reward of 10 when it comes in contact with a blue ball and collects it.
Cost: The agent receives immediate cost 1 when it comes in contact with a red ball, collecting it.

SafetyBallRun: The agent is incentivized to run as fast as possible in the x direction.
Reward: Increases proportionally to the velocity in the x direction.
Cost: Received when the agent exceeds a velocity threshold of 2.5 or when leaving the non-physical boundary.

SafetyBallReach: The agent is incentivized to chase the area marked in

green while avoiding the rectangular obstacle and areas marked in blue.
Reward: Shaped reward based on the Euclidean distance between the agent and the goal. Sparse when the agent comes in contact with the green area.
Cost: sparse cost received when the agent comes in contact with the rectangular obstacle or walks over the blue zone.

CartSafe: The agent is a cart. Actions are discrete $a \in \{0, 1\}$. The agent is incentivized to balance the pole upright while staying within bounds.
Reward: Shaped reward that scales with the upright pole position: $r(s) = 1 + cos(Poleangle)$.
Cost: $c(s) = \mathbb{1}[-1 < \textbf{cart position} < 1]$
The episode terminates if the distance between the cart and the centre exceeds 2.4.

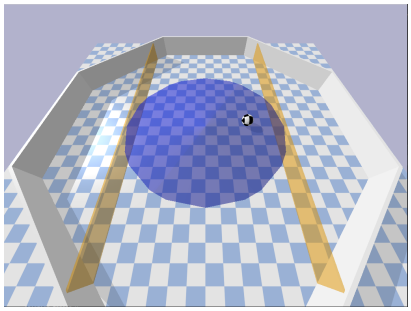


Figure 8.1: SafetyBallCircle

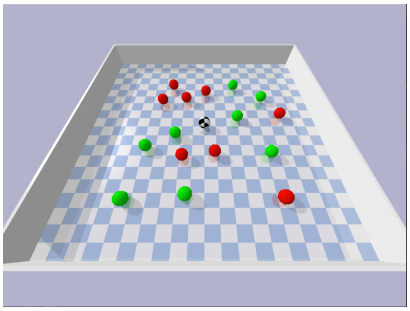


Figure 8.2: SafetyBallGather

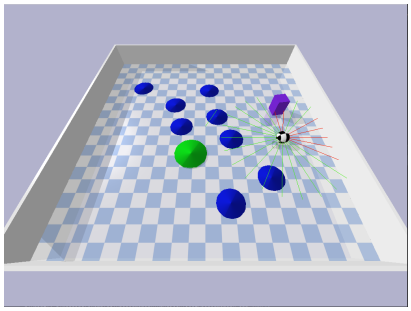


Figure 8.3: SafetyBallReach

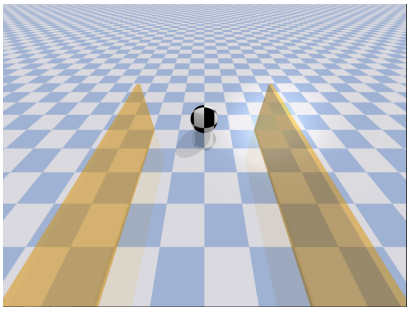


Figure 8.4: SafetyBallRun

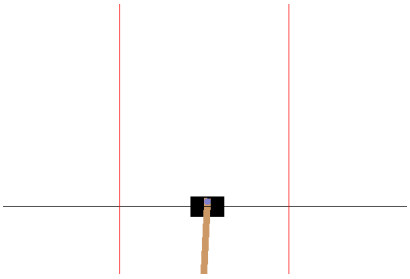


Figure 8.5: CartSafe

8.2 Bullet-Safety-Gym benchmark

8.2.1 Other safe reinforcement learning algorithms

We compare our algorithm SCPO [1] to the following on-policy algorithms:

- Trust-Region Policy Optimization (TRPO) [24]: unconstrained policy optimization algorithm. A line search is used to determine policy update step size.
- TRPO-L: Uses the TRPO objective and transforms the constrained optimization problem to an unconstrained one using Lagrange multiplier. The Lagrange multiplier is learnable and changes during policy iteration.
- Constrained Policy Optimization (CPO) [2]: Computes Lagrange multiplier for each policy iteration step. Uses the trust region objective.
- Primal-dual Optimization (PDO): Uses a Lagrange multiplier that can be learned and retains its state.
- Projection-based Constrained Policy Optimization (PCPO) [26]: is a two-stage optimization technique based on CPO. The first step updates the parameters without constraints. The second addresses constrained violation by projecting the policy parameters on the constraint set.

We use TRPO to estimate an upper bound of the return and cost when safety is not considered. The above algorithms and their benchmarks are provided by Sven Gronauer and described in more detail in his technical report [8].
A discount factor $\gamma = 0.99$ was used, and $B = 32000$ environment steps were collected. To train safe SCPO, we use $B = 32768$ environment steps. Therefore, we scale our plots to match the other algorithms and omit the x-axis label.
The neural network architecture of all algorithms consists of a multi-layer perceptron (MLP) with two hidden layers of size 64, followed by a tanh non-linearity. The Adam optimizer [13] is used in our implementation.
Each algorithm was evaluated on four different random seeds. If an algorithm violates the safety constraint over all hyperparameters, we pick the hyperparameter with the least average cumulative cost over the last 50 iterations. We use Stable Baselines 3 [20] as the foundation for our implementation.

8.2.2 Experiment results

Figure [8.7] demonstrates that SCPO outperforms all other algorithms. Although the final return is similar for most tasks, SCPO significantly outperforms in the SafetyBallGather task. Furthermore, the cost and cost standard deviation of SCPO are consistently lower than the other algorithms during training.
In Figure [8.6], the safe episode ratio generated by the policy during training is depicted. Although the safe episode ratio is not available for the other algorithms, we can infer that they violate the safety constraint more than SCPO throughout training; the mean cost is almost equal to the max cost, and the cost standard deviation is high.
Additionally, SCPO is capable of finding a safe policy at a faster rate compared to the other algorithms. The cost promptly drops below the threshold without compromising the improvement of the return. This highlights SCPO's ability to better balance the two primary objectives: satisfying safety constraints and maximizing return.

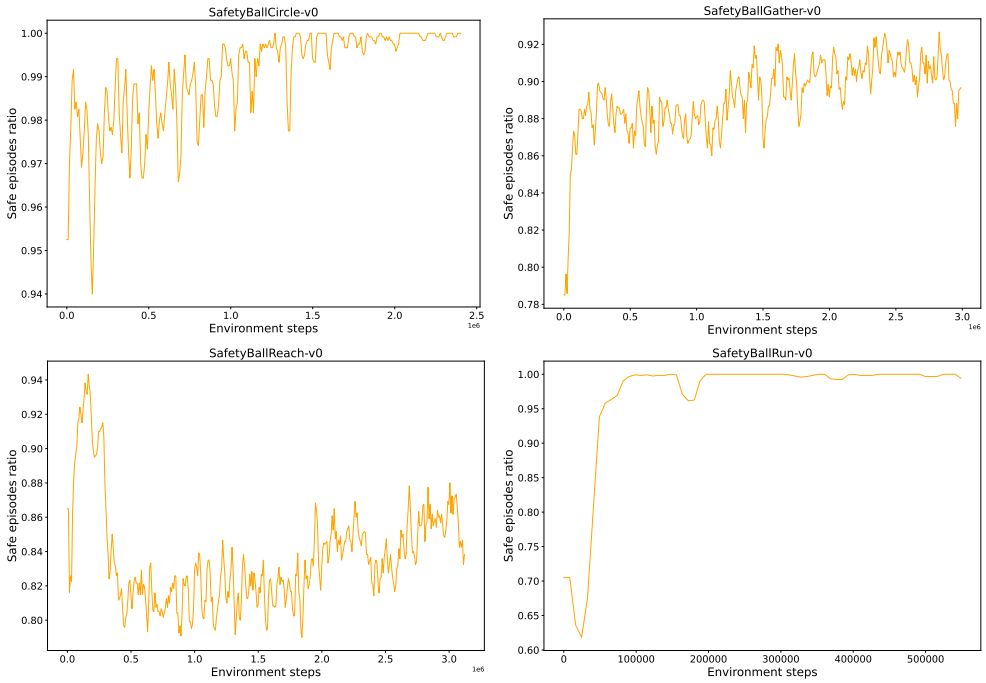


Figure 8.6: The ratio of safe episodes generated when training using SCPO.

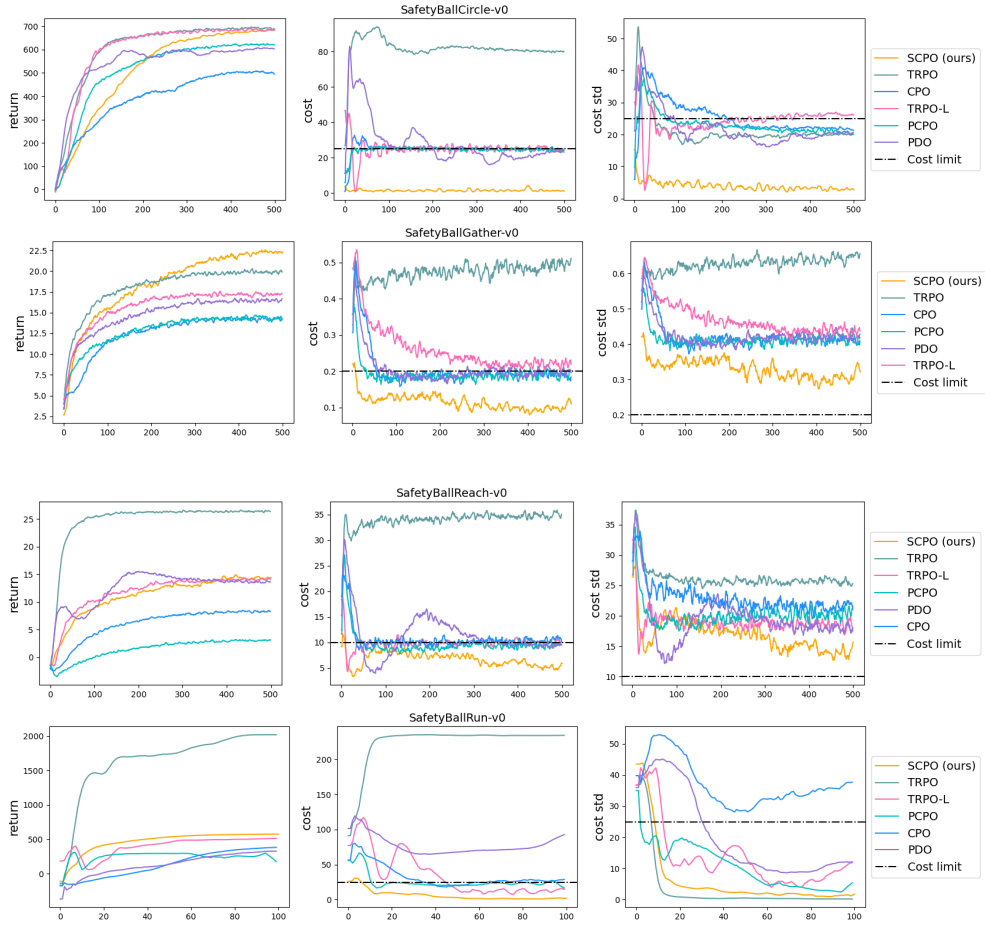


Figure 8.7: The learning curve of each algorithm average over four different seeds.

8.2.3 CartSafe

The action space of CartSafe is discrete. Unlike the other environments, a randomly initialized policy violates the safety constraint with high probability, e.g. $Q^c_{\pi_0}(s,a) \approx 0$ for all state s and action a. Under this condition, using the objective function $V^{r,k}$ might lead to difficulty in learning a safe behavior. However, this problem is solved by introducing the objective function $V^{rc,k}$ as described in subsection 6.2. When all state-action pairs are unsafe, using $V^{rc,k}$ is equivalent to minimizing the cumulative cost, eventually leading to a safe policy.
Figure 8.8 demonstrates the ability of SCPO to quickly find a safe policy and simultaneously improve the return. Safety critic plays a crucial role, enabling a seamless transition from cost reduction to return maximization.

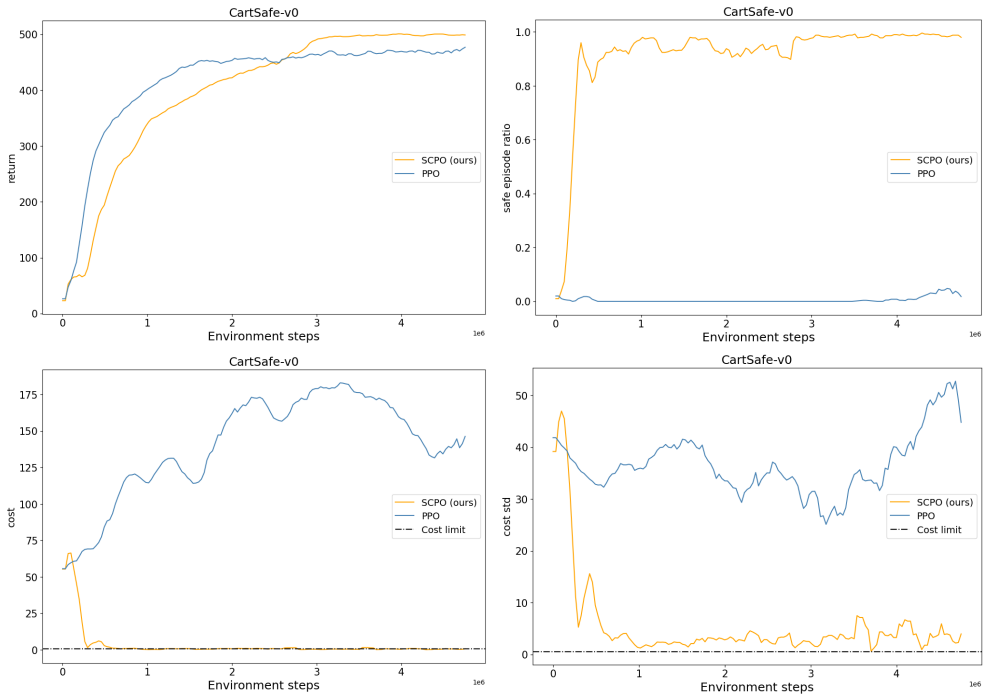


Figure 8.8: The learning curve of SCPO and PPO using the environment CartSafe-v0.

8.3 Augmenting state effect

We investigate the effect of state augmentation by training a policy using safe ppo on the environment **SafetyBallRun**. The agent receives a reward proportional to its velocity. A cost of 1 is incurred when the agent's velocity exceeds 2.5. We train two policies: one using augmented states (as explained in Section 4.1) and another using normal states. Both policies share the same hyperparameters.

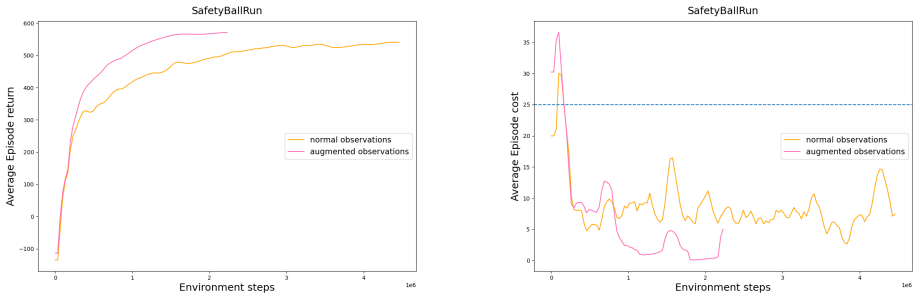


Figure 8.9: SafetyBallCircle return and cumulative cost.

The policy trained with augmented states converges faster to the maximum return and requires training samples, as shown in figure 8.10. While both policies achieve a similar return, they exhibit different behavior.

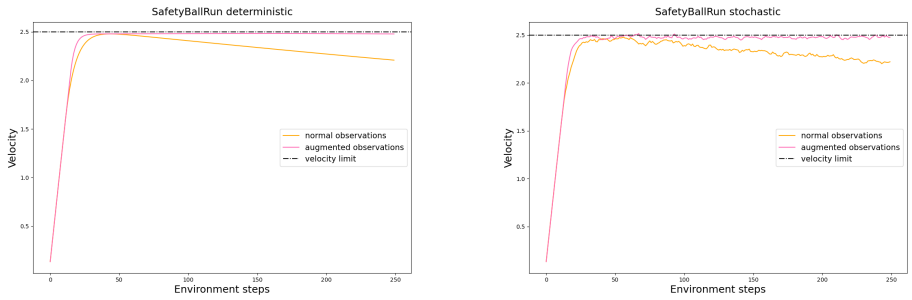


Figure 8.10: SafetyBallCircle velocity.

The policy trained with augmented states maintains constant velocity after accelerating. It knows the current cumulative cost. Exceeding the velocity limit does not immediately violate the safety constraint. The same unaugmented state can cause constraint violation or be safe depending on the current cumulative cost value. Therefore, using augmented states removes this ambiguity. The policy trained with normal states accelerates then decelerates. The policy likely anticipates the constraint violation but does not have the means to correctly identify risky states.

9 Conclusion

In this thesis, we proposed a novel approach to solve safe reinforcement learning problems. Our work introduces the safety critic, which is used to nullify rewards obtained by violating safety constraints. Moreover, safety critic helps manage the trade-off between cost reduction and return maximization. Our approach is straightforward to implement and can be easily integrated with current RL methods. Empirically, we validate our theoretical findings and compare our algorithm (SCPO) to TRPO-L [21], CPO [2], PDO and PCPO [26]. Our approach violates the safety constraint significantly less than the other algorithms throughout training without sacrificing improving the return. It also converges to a safe policy faster than other approaches. Our work is a step forward in deploying RL to real-world problems where safety guarantees are critical. Future research will focus on convergence analysis and evaluating the efficacy of SCPO in complex and challenging environments. Additionally, the neural network architecture of the safety critic can be improved to be statistically more sound.

10 Appendix

Environment Name	Cost limit	Maximum Episode Length
SafetyBallCircle-v0	25	250
SafetyBallRun-v0	25	250
SafetyBallGather-v0	0.2	250
SafetyBallReach-v0	10	250
CartSafe-v0	1	$[0, 300]$

Table 10.1: Environments overview.

Hyper-parameter	BallCircle	BallGather	BallRun	BallReach	CartSafe
Batch-size	64	64	64	64	64
Epochs	5	5	5	5	5
Learning rate	2e-4	2e-4	2e-4	2e-4	2e-4
Optimizer	Adam	Adam	Adam	Adam	Adam
Timesteps T	32768	32768	32768	32768	32768
Entropy co-efficient	0.01	0.01	0.005	0.01	0.001
Clip range	0.2	0.2	0.2	0.2	0.2
GAE factor rewards λ	0.95	0.95	0.95	0.95	0.95
Discount γ	0.99	0.99	0.99	0.99	0.99
Safety discount γ (4.3)	0.995	0.995	0.995	0.995	0.995
Reward bias b	1.5	0.05	1	0.1	0
k ($V^{rc,k}$)	2	4	4	4	5
Cost factor β	0	15	0.5	0	3

Table 10.2: Hyper-parameters used to train SCPO.

10.1 Lyapunov Approach

In the early stages of this thesis, we considered basing our work on the Lyapunov approach [18, 5, 6]. After further investigation, we decided to explore a novel algorithm instead. In the following, we provide some theoretical properties and explain how the Lyapunov approach can be simplified in the deterministic case.

10.1.1 Preliminaries

We defined the generic bellman operator as:

$$T_{\pi,h}[V](x) = \sum_a \pi(a \mid x) \left[h(x,a) + \sum_{x' \in \mathcal{X}'} P\left(x' \mid x, a\right) V\left(x'\right) \right]$$

We model the constraint reinforcement problem with a constraint Markov decision process (CMDP), which is defined by $(\mathcal{X}, \mathcal{A}, c, d, P, x_0, d_0)$. $\mathcal{X}$ and $\mathcal{A}$ are the state and action space, $d : \mathcal{X} \to \mathbb{R}_{\geq 0}$ is the immediate constraint cost and $d(x) \in [0, D_{\max}]$. $c : \mathcal{X}, \mathcal{A} \to \mathbb{R}$ is the cost function and $P(\cdot \mid x, a)$ is the transition probability.
Let $\Delta(x) = \{\pi(\cdot \mid x) : \mathcal{X} \to \mathbb{R}_{\geq 0s} : \sum_a \pi(a \mid x) = 1\}$ be the set of Markov stationary policies for any state $x \in \mathcal{X}$.
The paper [5] also defines T^* as a random variable corresponding to the first-hitting time of the terminal state x_{Term} induced by policy π.
We denote $\mathcal{C}_\pi(x) = \mathbb{E}_\pi \left[\sum_{t=0}^{T^*-1} c(x_i, a_i) \mid x_0 = x \right]$ and
$\mathcal{D}_\pi(x) = \mathbb{E}_\pi \left[\sum_{t=0}^{T^*-1} d(x_i) \mid x_0 = x \right]$
Given an initial state x_0 and a threshold d_0, We wish to solve the following problem denoted as $\mathcal{OPT}$:

$$\min_{\pi \in \Delta} \mathcal{C}_\pi\left(x_0\right) \quad \text{s.t } \mathcal{D}_\pi\left(x_0\right) \leq d_0$$

Let π_B be a feasible policy of the $\mathcal{OPT}$ problem. The paper defines a non empty set of Lyapunov functions w.r.t state $x_0 \in \mathcal{X}$ and constraint threshold d_0 as:

$$\mathcal{L}_{\pi_B}\left(x_0, d_0\right) = \left\{ L : \mathcal{X} \to \mathbb{R}_{\geq 0} : T_{\pi_B,d}[L](x) \leq L(x), \forall x \in \mathcal{X}'; L(x) = 0, \forall x \in \mathcal{X} \backslash \mathcal{X}'; L\left(x_0\right) \leq d_0 \right\}$$

For any arbitrary Lyapunov function $L \in \mathcal{L}_{\pi_B}\left(x_0, d_0\right)$, denote by $\mathcal{F}_L(x) = \{\pi(\cdot \mid x) \in \Delta : T_{\pi,d}[L](x) \leq L(x)\}$ the set of L-induced Markov stationary policies.

Lemma 10.1.1 *There exists an auxiliary constraint cost $\epsilon : \mathcal{X}' \to \mathbb{R}$ such that the Lyapunov function is given by $L_\epsilon(x) = \mathbb{E}\left[\sum_{t=0}^{T^*-1} d(x_t) + \epsilon(x_t) \mid \pi_B, x\right], \forall x \in \mathcal{X}'$, and $L_\epsilon(x) = 0$ for $x \in \mathcal{X} \backslash \mathcal{X}'$. Moreover, L_ϵ is equal to the constraint value function w.r.t. π^*, i.e., $L_\epsilon(x) = \mathcal{D}_{\pi^*}(x)$.*

Estimating ϵ requires knowledge of the optimal policy π^*. Therefore, the paper proposes to estimate $\epsilon^*(x) = \max_{x \in \mathcal{X}}(x) \geq 0$.
For any arbitrary Lyapunov function $L \in \mathcal{L}_{\pi_B}(x_0, d_0)$, denote by
$\mathcal{F}_L = \{\pi(\cdot \mid x) \in \Delta : T_{\pi,d}[L](x) \leq L(x)\}$ the set of L-induced Markov stationary policies.
The paper proposes an assumption that constrains the maximal distance between π^* and π_B. Under it, they can guarantee that $\pi^* \in \mathcal{F}_{L_{\epsilon^*}}$ where

$$L_{\epsilon^*}(x) = \mathbb{E}\left[\sum_{t=0}^{T^*-1} d(x_t) + \epsilon^*(x_t) \mid x_0 = x\right] \qquad \text{and } \epsilon^*(x) \geq 0$$

10.1.2 Lyapunov function properties

Lemma 10.1.2 *For an arbitrary policy π*

$$\forall L \in \mathcal{L}_\pi \; \forall k \in \mathbb{N}_{\geq 1} : D_\pi(x) = \lim_{q \to \infty} T^q_{\pi,d}[L](x) \leq T^k_{\pi,d}[L](x) \leq L(s)$$

Proof:
For k=1, $T_{\pi,d}[L](x) \leq L(s)$ follows by definition.
Assuming the property holds for k:

$$\begin{aligned} T^{k+1}_{\pi,d}[L](x) &= d(x) + \sum_{a \in \mathcal{A}} \sum_{x' \in \mathcal{X}} \pi(a \mid x) P(x' \mid x, a) T^k_{\pi,d}[L](x) \\ &\leq d(x) + \sum_{a \in \mathcal{A}} \sum_{x' \in \mathcal{X}} \pi(a \mid x) P(x' \mid x, a) L(x') \\ &= L(x) \end{aligned}$$

$D_{\pi,d}(x) = \lim_{q \to \infty} T^q_{\pi,d}[L](x)$ Because $T_{\pi,d}$ is a contraction mapping and $\mathcal{D}_\pi$ is a fix point.

A Lyapunov function is an upper bound on the expected cumulative cost.

We use the following notion:
$P_\pi[X_{t+1} = x_{t+1} \mid X_t = x_t] = \sum_{a \in \mathcal{A}} \pi(a \mid x_t)\; P[X_{t+1} = x_{t+1} \mid X_t = x_t, A_t = a]$

Lemma 10.1.3 $\forall x \in \mathcal{X}\ \forall T \geq 0$:

$$\sum_{t=0}^{T} \sum_{x \in \mathcal{X}} P_\pi[X_t = x \mid X_0 = x_0] d(x) + \sum_{x \in \mathcal{X}} P_\pi[X_{T+1} = x \mid X_0 = x_0]\ L(x) \leq L(x_0) \tag{10.1}$$

Proof:
For $T = 0$, the inequality holds by definition.
Suppose the inequality holds for arbitrary $T \geq 0$:

$$\sum_{t=0}^{T} \sum_{x \in \mathcal{X}} P_\pi[X_t = x \mid X_0 = x_0] d(x) + \sum_{x \in \mathcal{X}} P_\pi[X_{T+1} = x \mid X_0 = x_0]\ L(x) \leq L(x_0)$$

$$\Rightarrow \begin{cases} \sum_{t=0}^{T} \sum_{x \in \mathcal{X}} P_\pi[X_t = x \mid X_0 = x_0] d(x) + \sum_{x \in \mathcal{X}} P_\pi[X_{T+1} = x \mid X_0 = x_0]\ L(x) \leq L(x_0) \\ L(x) = d(x) + \sum_{x' \in \mathcal{X}} P[X_{T+2} = x' \mid X_{T+1} = x] L(x') \end{cases}$$

$$\Rightarrow \sum_{t=0}^{T+1} \sum_{x \in \mathcal{X}} P_\pi[X_t = x \mid X_0 = x_0] d(x) + \sum_{x,x' \in \mathcal{X}} P_\pi[X_{T+2} = x' \mid X_{T+1} = x] P_\pi[X_{T+1} = x \mid X_0 = x_0] L(x')$$

$$\Rightarrow \sum_{t=0}^{T+1} \sum_{x \in \mathcal{X}} P_\pi[X_t = x \mid X_0 = x_0] d(x) + \sum_{x,x' \in \mathcal{X}} P_\pi[X_{T+2} = x', X_{T+1} = x \mid X_0 = x_0] L(x')$$

$$\Rightarrow \sum_{t=0}^{T+1} \sum_{x \in \mathcal{X}} P_\pi[X_t = x \mid X_0 = x_0] d(x) + \sum_{x' \in \mathcal{X}} P_\pi[X_{T+2} = x' \mid X_0 = x_0] L(x')$$

Therefore the inequality holds for all $T + 1$

Corollary 10.1.3.1 *For all π in $\mathcal{F}_L$ The following inequality holds:*

$$\forall x \in \mathcal{X} : \qquad \max_{t \geq 0} P_\pi[X_t = x \mid X_0 = x_0] L(x) \leq d_0 \tag{10.2}$$

Proof:
Let $x' \in \mathcal{X}$ and $T \geq 0$

$$\begin{cases} \sum_{t=0}^{T} \sum_{x \in \mathcal{X}} P_\pi[X_t = x \mid X_0 = x_0] d(x) + \sum_{x \in \mathcal{X}} P_\pi[X_{T+1} = x \mid X_0 = x_0]\ L(x) \leq L(x_0) \\ -\sum_{t=0}^{T} \sum_{x \in \mathcal{X}} P_\pi[X_t = x \mid X_0 = x_0] d(x) \leq 0 \end{cases}$$

$$\Rightarrow \sum_{x \in \mathcal{X}} P_\pi[X_{T+1} = x \mid X_0 = x_0]\ L(x) \leq L(x_0)$$

$$\Rightarrow P_\pi[X_{T+1} = x' \mid X_0 = x_0]\ L(x') \leq L(x_0)$$

Therefore:

$$
\begin{aligned}
&\begin{cases} \forall t \geq 1 : \quad P_\pi[X_t = x' \mid X_0 = x_0]\ L(x') \leq L(x_0) \\ P_\pi[X_0 = x' \mid X_0 = x_0]\ L(x') = \mathbb{1}[x_0 = x']L(x_0) \end{cases} \\
&\Rightarrow \begin{cases} \forall t \geq 1 : \quad P_\pi[X_t = x' \mid X_0 = x_0]\ L(x') \leq L(x_0) \\ P_\pi[X_0 = x' \mid X_0 = x_0]\ L(x') \leq L(x_0) \end{cases} \\
&\Rightarrow \forall t \geq 0\ P_\pi[X_t = x' \mid X_0 = x_0]\ L(x') \leq L(x_0) \\
&\Rightarrow \max_{t \geq 0}\ P_\pi[X_t = x' \mid X_0 = x_0]\ L(x') \leq L(x_0) \\
&\Rightarrow \max_{t \geq 0}\ P_\pi[X_t = x' \mid X_0 = x_0]\ L(x') \leq d_0
\end{aligned}
$$

10.1.3 Deterministic case

In this section, we assume a deterministic environment. Therefore, an optimal deterministic policy exists π^* for the $\mathcal{OPT}$ problem.
Let τ^* be the optimal trajectory induced by π^* of length T^*. For all $t \in [0, T^* - 1]$, we defined $x_t^* \in \mathcal{X}$ such that $P_\pi^*[X_t = x_t^* \mid X_0 = x_0] = 1$. Hence, $\tau^* = x_0 x_1^*, x_2^* ... x_{T-1}^*$
Let $L(x) = D_{\pi_B}(x)$ for all $x \in \mathcal{X}$

Lemma 10.1.4

$$\forall i \in [0, T^* - 1] \quad L(x_i^*) \leq d_0 \quad \textit{and} \quad T_{\pi^*,d}[L](x_i^*) \leq d_0 \tag{10.3}$$

Proof:
Let $t \in [0, T^* - 1]$ arbitrary. L_{ϵ^*} is a valid Lyapunov function for π^* Because $\pi^* \in \mathcal{F}_{L_{\epsilon^*}}$

$$
\begin{aligned}
\forall x \in \mathcal{X}\ \max_{t' \geq 0} P_{\pi^*}[X_{t'} = x \mid X_0 = x_0] L_{\epsilon^*}(x) \leq d_0 \Rightarrow \max_{t' \geq 0} P_{\pi^*}[X_{t'} = x_t^* \mid X_0 = x_0] L_{\epsilon^*}(x_t^*) \leq d_0 \\
\Rightarrow P_{\pi^*}[X_t = x_t^* \mid X_0 = x_0] L_{\epsilon^*}(x_t^*) \leq d_0 \\
\Rightarrow L_{\epsilon^*}(x_t^*) \leq d_0 \\
\Rightarrow \begin{cases} L(x_t^*) \leq d_0 \\ T_{\pi^*,d}[L_{\epsilon^*}](x_i^*) \leq d_0 \end{cases} \\
\Rightarrow \begin{cases} L(x_t^*) \leq d_0 \\ T_{\pi^*,d}[L](x_i^*) \leq d_0 \end{cases}
\end{aligned}
$$

$\forall a \in \mathcal{A}$, we define $\pi_1^* \in \Delta$ as follows:

$$\pi_1^*(a \mid x) = \begin{cases} \pi^*(a \mid x) & \text{if } \exists t \geq 0 \; P_{\pi^*}[X_t = x \mid X_0 = x_0] > 0 \\ \pi_B(a \mid x) & \text{otherwise} \end{cases}$$

π_1^* is also optimal because it agrees with π^* for all states on the optimal trajectory τ^*

Corollary 10.1.4.1 *Let $L(x) = D_{\pi_B}(x)$ and assume $\pi^* \in \mathcal{F}_{L_{\epsilon^*}}$*

$$\pi_1^* \in \{\pi(. \mid x) \mid \forall x \in \mathcal{X} \quad T_{\pi,d}[L](x) \leq \max(d_0, L(x))\} \tag{10.4}$$

Proof:
For all states x such that $P_{\pi^*}[X_t = x \mid X_0 = x_0] \geq 0$, $\pi^* = \pi_1^*$ and the inequality holds directly from (10.3).
For other states π_B and π_1^* are equal, the inequality holds because L is a valid Lyapunov function with respect to π_B

Hence, instead of estimating $\epsilon^* : \mathcal{X} \to \mathbb{R}_{\geq 0s}$ and searching for the optimal policy in the set $\mathcal{F}_{L_{\epsilon^*}}$ we can instead search for it in the above-mentioned set, which does not require extra computations.

List of Figures

Bibliography

[1] Naoki Abe et al. "Optimizing debt collections using constrained reinforcement learning." In: July 2010, pp. 75–84. DOI: 10.1145/1835804.1835817.

[2] Joshua Achiam et al. *Constrained Policy Optimization*. 2017. DOI: 10.48550/ARXIV.1705.10528. URL: https://arxiv.org/abs/1705.10528.

[3] Rafael Basso et al. "Dynamic stochastic electric vehicle routing with safe reinforcement learning." In: *Transportation Research Part E: Logistics and Transportation Review* 157 (2022), p. 102496. ISSN: 1366-5545. DOI: https://doi.org/10.1016/j.tre.2021.102496. URL: https://www.sciencedirect.com/science/article/pii/S1366554521002581.

[4] Miguel Calvo-Fullana et al. *State Augmented Constrained Reinforcement Learning: Overcoming the Limitations of Learning with Rewards*. 2021. DOI: 10.48550/ARXIV.2102.11941. URL: https://arxiv.org/abs/2102.11941.

[5] Yinlam Chow et al. *A Lyapunov-based Approach to Safe Reinforcement Learning*. 2018. DOI: 10.48550/ARXIV.1805.07708. URL: https://arxiv.org/abs/1805.07708.

[6] Yinlam Chow et al. *Lyapunov-based Safe Policy Optimization for Continuous Control*. 2019. DOI: 10.48550/ARXIV.1901.10031. URL: https://arxiv.org/abs/1901.10031.

[7] Yan Duan et al. *Benchmarking Deep Reinforcement Learning for Continuous Control*. 2016. DOI: 10.48550/ARXIV.1604.06778. URL: https://arxiv.org/abs/1604.06778.

[8] Sven Gronauer. *Bullet-Safety-Gym: A Framework for Constrained Reinforcement Learning*. Tech. rep. mediaTUM, 2022.

[9] Shangding Gu et al. *A Review of Safe Reinforcement Learning: Methods, Theory and Applications*. 2022. DOI: 10.48550/ARXIV.2205.10330. URL: https://arxiv.org/abs/2205.10330.

[10] Shangding Gu et al. "Constrained Reinforcement Learning for Vehicle Motion Planning with Topological Reachability Analysis." In: *Robotics* 11.4 (2022). ISSN: 2218-6581. DOI: 10.3390/robotics11040081. URL: https://www.mdpi.com/2218-6581/11/4/81.

[11] Mohammadhosein Hasanbeig, Alessandro Abate, and Daniel Kroening. *Cautious Reinforcement Learning with Logical Constraints*. 2020. DOI: 10.48550/ARXIV.2002.12156. URL: https://arxiv.org/abs/2002.12156.

[12] Ashish Kumar Jayant and Shalabh Bhatnagar. *Model-based Safe Deep Reinforcement Learning via a Constrained Proximal Policy Optimization Algorithm*. 2022. DOI: 10.48550/ARXIV.2210.07573. URL: https://arxiv.org/abs/2210.07573.

[13] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[14] Pavlo Krokhmal, Jonas Palmquist, and Stan Uryasev. "Portfolio Optimization With Conditional Value-at-Risk Objective and Constraints." In: *Journal of Risk* 4 (May 2003). DOI: 10.21314/JOR.2002.057.

[15] Hepeng Li, Zhiqiang Wan, and Haibo He. "Constrained EV Charging Scheduling Based on Safe Deep Reinforcement Learning." In: *IEEE Transactions on Smart Grid* 11.3 (2020), pp. 2427–2439. DOI: 10.1109/TSG.2019.2955437.

[16] Xiaobai Ma et al. *Reinforcement Learning for Autonomous Driving with Latent State Inference and Spatial-Temporal Relationships*. 2020. DOI: 10.48550/ARXIV.2011.04251. URL: https://arxiv.org/abs/2011.04251.

[17] Branka Mirchevska et al. "High-level Decision Making for Safe and Reasonable Autonomous Lane Changing using Reinforcement Learning." In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 2156–2162. DOI: 10.1109/ITSC.2018.8569448.

[18] Theodore J Perkins and Andrew G Barto. "Lyapunov design for safe reinforcement learning." In: *Journal of Machine Learning Research* 3.Dec (2002), pp. 803–832.

[19] Tu-Hoa Pham, Giovanni De Magistris, and Ryuki Tachibana. *OptLayer - Practical Constrained Optimization for Deep Reinforcement Learning in the Real World*. 2017. DOI: 10.48550/ARXIV.1709.07643. URL: https://arxiv.org/abs/1709.07643.

[20] Antonin Raffin et al. "Stable-Baselines3: Reliable Reinforcement Learning Implementations." In: *Journal of Machine Learning Research* 22.268 (2021), pp. 1–8. URL: http://jmlr.org/papers/v22/20-1364.html.

[21] Alex Ray, Joshua Achiam, and Dario Amodei. "Benchmarking Safe Exploration in Deep Reinforcement Learning." In: (2019).

[22] John Schulman et al. *High-Dimensional Continuous Control Using Generalized Advantage Estimation*. 2015. DOI: 10.48550/ARXIV.1506.02438. URL: https://arxiv.org/abs/1506.02438.

[23] John Schulman et al. *Proximal Policy Optimization Algorithms*. 2017. DOI: 10.48550/ARXIV.1707.06347. URL: https://arxiv.org/abs/1707.06347.

[24] John Schulman et al. *Trust Region Policy Optimization*. 2015. DOI: 10.48550/ARXIV.1502.05477. URL: https://arxiv.org/abs/1502.05477.

[25] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*.

[26] Tsung-Yen Yang et al. *Projection-Based Constrained Policy Optimization*. 2020. DOI: 10.48550/ARXIV.2010.03152. URL: https://arxiv.org/abs/2010.03152.