POLITECNICO DI MILANO
SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING

Balancing Safety and Exploration in Policy Gradient

Supervisor: Prof. Marcello Restelli

Co-supervisor: Dott. Matteo Papini

Andrea Battistello, 873795

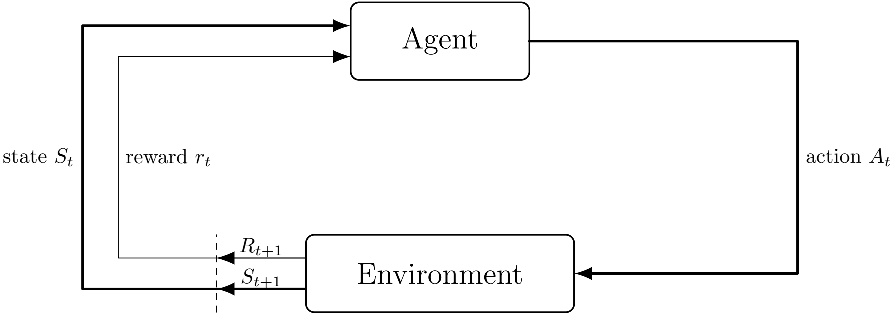
July 25, 2018

Contents

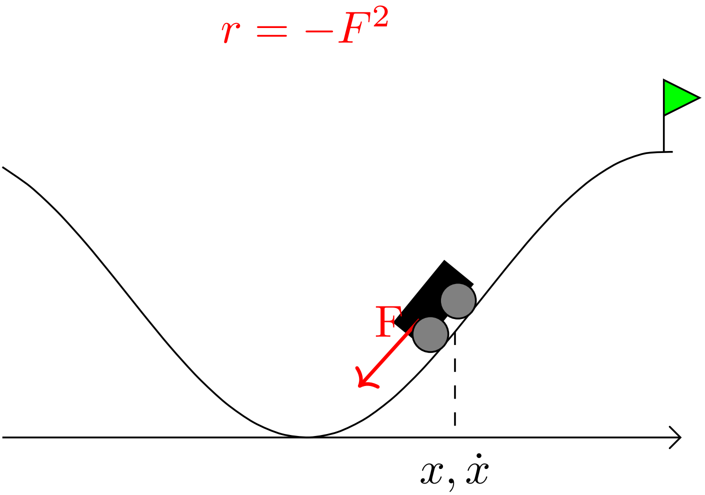
Contents

Reinforcement Learning

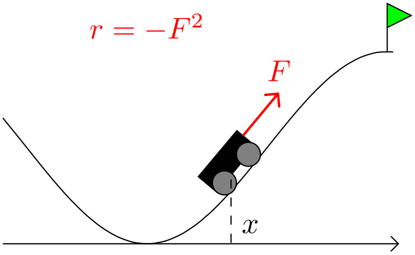
We learn from the **interaction** with the environment
For each step, the agent performs an action and receives an observation

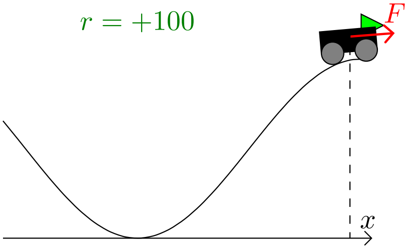


Example: Mountain Car



Example: Mountain Car





A Reinforcement Learning method

Policy

A policy $\pi_{\boldsymbol{\theta}}$ is a function that maps states to actions

Performance

The performance $J(\boldsymbol{\theta})$ of a policy is the discounted sum of rewards obtained by following $\pi_{\boldsymbol{\theta}}$

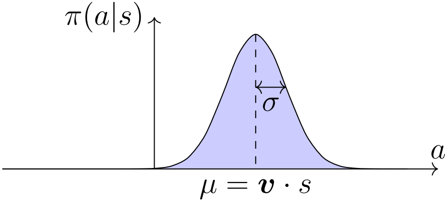
$$J(\boldsymbol{\theta}) = \mathbb{E}_{a_t \sim \pi_{\boldsymbol{\theta}}} \left[\sum_{k=0}^{T} \gamma^k r(s_k, a_k) \right]$$

Policy Gradient method

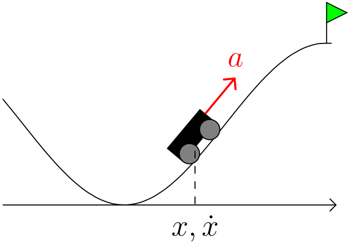
The optimal policy can be found by gradient ascent on policy parameters:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

Gaussian policies

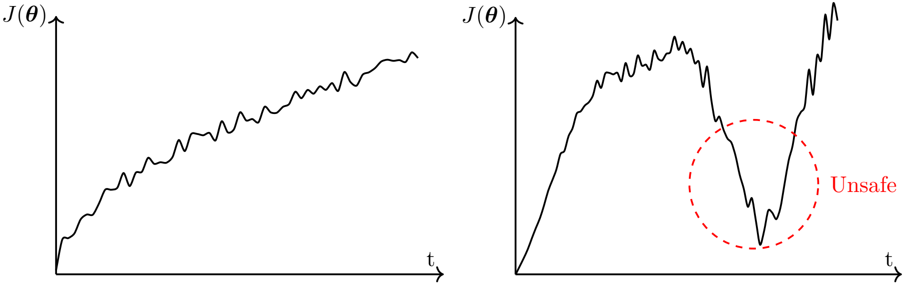


$$\pi(a|s) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2}\left(\frac{a-\boldsymbol{v}\cdot s}{\sigma}\right)^2\right)$$



$$\begin{aligned} s &= [x; \dot{x}]^T \\ \sigma &= e^w \\ \boldsymbol{\theta} &= [\boldsymbol{v}^T; w]^T \end{aligned}$$

Contents



(a) Ok (b) Not ok

Safe Reinforcement Learning

Definition

Given a reference $\underline{J}$ and a confidence level δ, a RL algorithm is **safe** if it yields a policy with performance lower than $\underline{J}$ with probability at most δ.

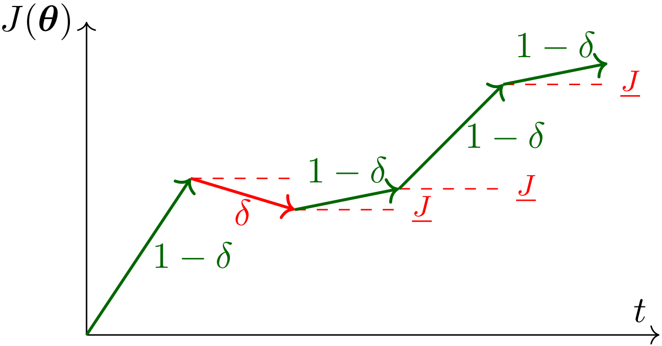


Figure: Example of monotonic improvements

State of the art in Safe Reinforcement Learning

We will mainly refer to the results on Adaptive Step Size from Pirotta et al.

- Assuming a policy gradient method with Gaussian-parameterized policies $\pi_{\boldsymbol{\theta}}$, an update rule of the form $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ and **fixed variance** $\sigma = e^{w} = \mathit{const}$:

$$J(\boldsymbol{\theta}') - J(\boldsymbol{\theta}) \geq \alpha \left\| \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \right\|_2^2 - \alpha^2 \left\| \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \right\|_1^2 \frac{c_3 + c_2 \sigma}{c_1 \sigma^3}$$

- that is maximized by the following step size:

$$\alpha^* = \frac{c_1 \sigma^3 \left\| \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \right\|_2^2}{(c_2 \sigma + c_3) \left\| \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \right\|_1^2},$$

- that guarantees an improvement of:

$$J(\boldsymbol{\theta}') - J(\boldsymbol{\theta}) \geq \frac{1}{2} \alpha^* \left\| \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \right\|_2^2 .$$

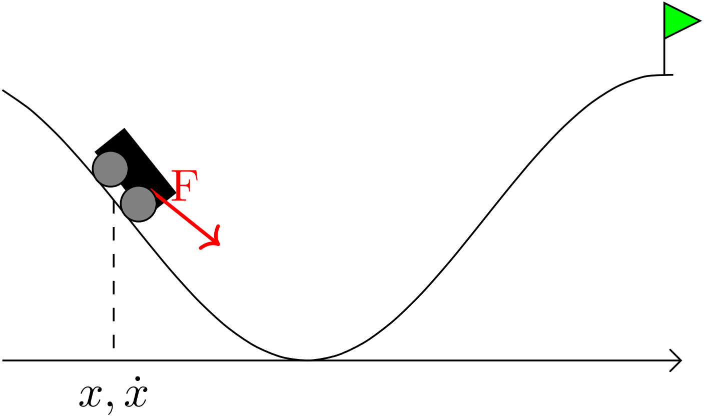
Problems of Safe Reinforcement Learning

The results seen so far suffer from the following problems:

- They target only **monotonic improvements**
  - This covers only specific user needs.
- They are **overly conservative**
  - Resulting in a very slow convergence speed.
- They do not consider an **adaptive variance**
  - The exploration factor remains constant and highly depends on the domain.

Contents

$$r = -F^2$$



Exploration in Reinforcement Learning

Exploration is intended as performing actions that are **different** than the ones the agent is used to.

- Obtain more informations about the environment
- Find new solutions (possibly better)
- Escape from local maxima

Contents

Contributions (1)

We extended the performance improvement bounds to include an adaptive exploration factor:

$$J(\boldsymbol{v}, w') - J(\boldsymbol{v}, w) \geq \beta \nabla_w J(\boldsymbol{v}', w)^2 - d\beta^2 \nabla_w J(\boldsymbol{v}', w)^2$$

that is maximized by:

$$\beta^* = \frac{1}{2d}$$

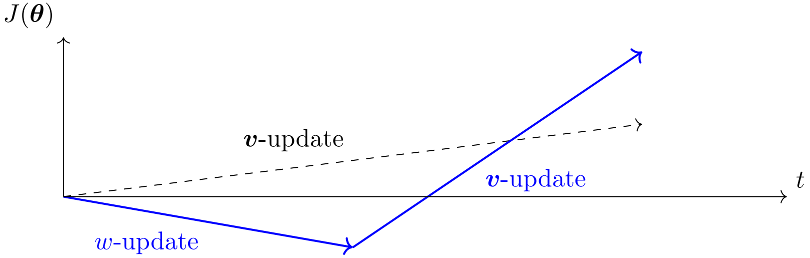
which guarantees:

$$J(\boldsymbol{v}, w') - J(\boldsymbol{v}, w) \geq \frac{\nabla_w J(\boldsymbol{v}', w)^2}{4d}$$

Recall:

$$\boldsymbol{\theta} = [\boldsymbol{v}; w]$$

Contributions (1)



We adapted the results by employing a new gradient direction:

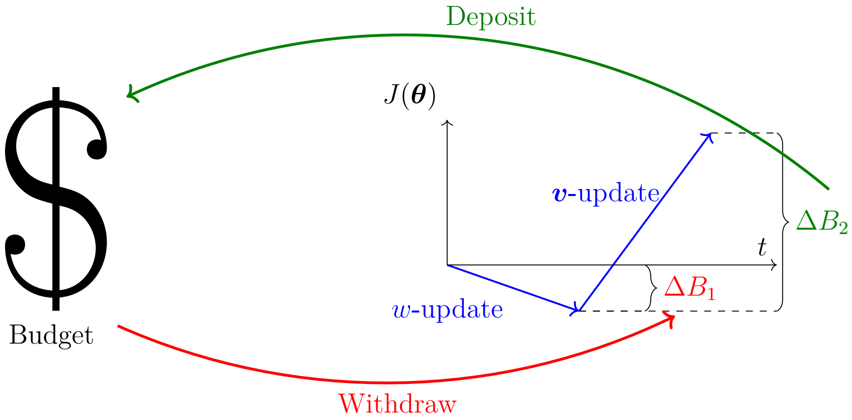
$$w \leftarrow w + \beta \nabla_w \mathcal{L}(\boldsymbol{v}, w),$$

where:

$$\nabla_w \mathcal{L}(\boldsymbol{v}, w) := \nabla_w \left(J(\boldsymbol{v}', w) - J(\boldsymbol{v}, w) \right)$$

Contributions (1)

We employed the budget trick to allow for a customizable implementation of this type of update (and more).



Contributions (2)

The second contribution was to introduce a new framework that goes beyond the single monotonic improvement case.

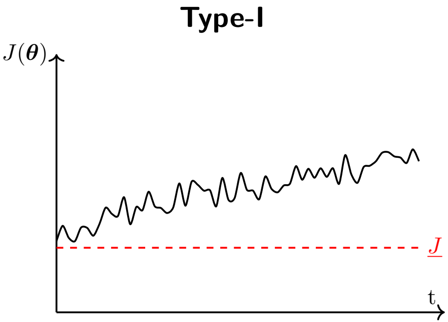
Type-I

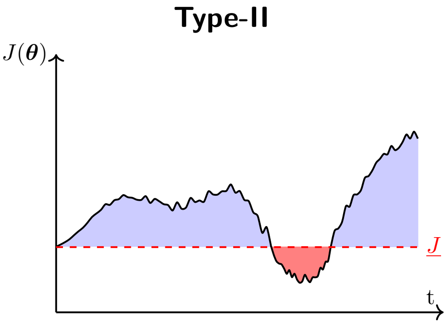
- We guarantee $J(\boldsymbol{\theta}^t) \geq J_B^t$ for each **policy update**.
- Suited for systems that requires high reliability.

Type-II

- We guarantee $J(\boldsymbol{\theta}^t) \geq J_B^t$ **on average** over a learning iteration
  (e.g., a production day).
- Suited for systems with lower safety needs.

Contributions (2)





Contributions (2)

Among these classes we have identified several variants:

MI Monotonic Improvement with type-I safety

LB-I Lower bounds to the initial policy with type-I safety

LB-II Lower bounds to the initial policy with type-II safety

— Variants with an additional target policy evaluation —

LV A low variance (e.g., prototype) policy is tested

ZV The deterministic policy is tested

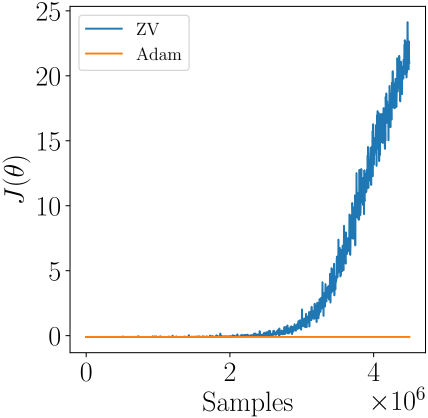
We devised a new general algorithm that can be customized to specific user needs.

Algorithm 1 Safely-Exploring Policy Gradient

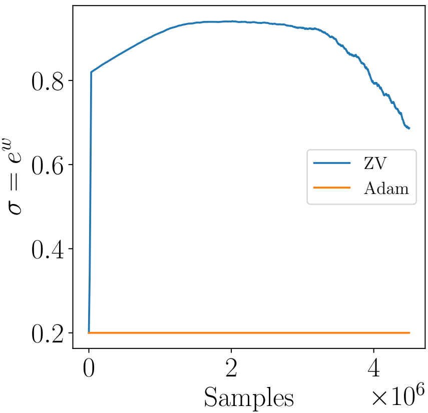
input: $\theta^0 = [\boldsymbol{v}^0, w^0],\ B^0$
for $t = 1, 2 \dots$ **do**
  $\boldsymbol{v}^{t+1} \leftarrow \boldsymbol{v}^t + \overline{\alpha}\nabla_{\boldsymbol{v}} J(\boldsymbol{v}^t, w^t)$ ▷ $\boldsymbol{v}$-update
  $B \leftarrow B + J(\boldsymbol{v}^{t+1}, w^t) - J(\boldsymbol{v}^t, w^t).$ ▷ $\boldsymbol{v}$-budget update
  $w^{t+1} \leftarrow w^t + \overline{\beta}\nabla_w \mathcal{L}(\boldsymbol{v}^{t+1}, w^t)$ ▷ w-update
  $B \leftarrow B + J(\boldsymbol{v}^{t+1}, w^{t+1}) - J(\boldsymbol{v}^{t+1}, w^t).$ ▷ w-budget update
end for

Contents

Results - Mountain Car task

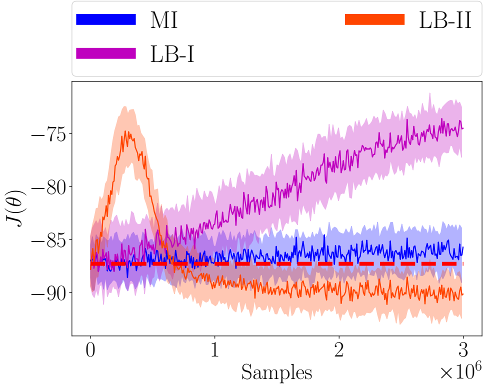


(a) Evolution of the performance

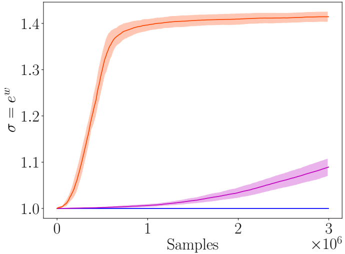


(b) Evolution of the exploration parameter

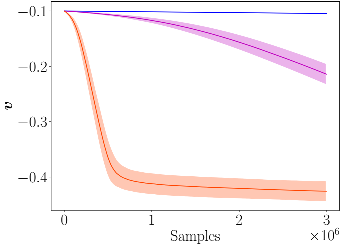
Results - Linear Quadratic Gaussian controller



(a) Performance of MI, LB-I and LB-II variants



(b) Evolution of variance σ



(c) Evolution of the mean $\boldsymbol{v}$

Contents

Conclusions

In this work we have introduced:

- An **adaptive way to explore** the environment
- A **new general framework** for safe reinforcement learning
- A **general algorithm** that can be customized to the user needs

Further works can focus on:

- New methods to adapt exploration using previous knowledge about the environment
- New ways to invest the budget
- Extend the result beyond the policy gradient method

Thank you for your attention

Questions?