Baumann, Dominik; Kowalczyk, Krzysztof; Tiels, Koen; Wachel, Paweł

A computationally lightweight safe learning algorithm

Published in:
2023 62nd IEEE Conference on Decision and Control, CDC 2023

DOI:
10.1109/CDC49753.2023.10384018

Published: 19/01/2024

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Baumann, D., Kowalczyk, K., Tiels, K., & Wachel, P. (2024). A computationally lightweight safe learning algorithm. In *2023 62nd IEEE Conference on Decision and Control, CDC 2023* (pp. 1022-1027). (Proceedings of the IEEE Conference on Decision & Control). IEEE. https://doi.org/10.1109/CDC49753.2023.10384018

A computationally lightweight safe learning algorithm

Dominik Baumann[1,2], Krzysztof Kowalczyk[3], Koen Tiels[4], and Paweł Wachel[3]

***Abstract*— Safety is an essential asset when learning control policies for physical systems, as violating safety constraints during training can lead to expensive hardware damage. In response to this need, the field of safe learning has emerged with algorithms that can provide probabilistic safety guarantees without knowledge of the underlying system dynamics. Those algorithms often rely on Gaussian process inference. Unfortunately, Gaussian process inference scales cubically with the number of data points, limiting applicability to high-dimensional and embedded systems. In this paper, we propose a safe learning algorithm that provides probabilistic safety guarantees but leverages the Nadaraya-Watson estimator instead of Gaussian processes. For the Nadaraya-Watson estimator, we can reach logarithmic scaling with the number of data points. We provide theoretical guarantees for the estimates, embed them into a safe learning algorithm, and show numerical experiments on a simulated seven-degrees-of-freedom robot manipulator.**

I. Introduction

Data-driven or learning-based control approaches have seen tremendous successes over the last years [1], [2]. Nevertheless, many popular machine algorithms, in particular those based on deep reinforcement learning, are challenging to apply to physical systems. One major shortcoming is the lack of theoretical guarantees, especially during exploration. Thus, deep reinforcement learning algorithms may violate safety constraints when applied to physical systems, potentially damaging the system and endangering the surroundings. The field of *safe learning* addresses this challenge. Based on varying degrees of assumed knowledge about the system dynamics, safe learning algorithms seek to learn (sub)optimal policies for dynamical systems while ensuring or encouraging safety [3].

In this paper, we focus on a class of algorithms that ensure safety during exploration with high probability while assuming that the concrete system dynamics are unknown. In exchange for the guarantees, the algorithms assume that a safe initial policy is known – otherwise, the algorithm would have no possibility to start the exploration safely – and a certain degree of regularity of the (unknown) functions that represent the constraints. A popular algorithm of this class is SafeOpt [4], [5]. Starting from the safe initial policy, SafeOpt carefully explores the policy space while only exploring policies that can be classified with high probability as safe given the regularity assumptions. In particular, SafeOpt uses Gaussian processes (GPs) [6] to model the constraint functions. While GPs are a powerful tool for function estimation, they are also a bottleneck of the approach: GP inference scales cubically with the number of data points. This scaling property limits the applicability of such algorithms for many emerging applications. For instance, if we want mobile robots that autonomously learn control policies, learning algorithms must be implemented on embedded devices with limited computing resources. Furthermore, even when extensive computing resources are available, cubic scaling properties limit the potential to scale to high-dimensional systems.

To address this problem, we propose CoLSafe, a computationally lightweight safe algorithm that adopts the main characteristics of SafeOpt but uses the Nadaraya-Watson estimator [7], [8] instead of GPs to approximate the constraints. The Nadaraya-Watson estimator scales logarithmically with the number of data points [9]. We show that we can derive similar safety guarantees as for SafeOpt and compare both algorithms in numerical simulations of a robot arm.

Contributions. In this paper, we

- propose CoLSafe, a novel safe learning algorithm based on SafeOpt with significantly lower computational complexity;
- derive probabilistic safety guarantees;
- evaluate the algorithm in numerical experiments.

Related work. For a general introduction to and overview of the safe learning literature, we refer the reader to [3]. Here, we focus on methods that provide probabilistic safety guarantees without knowledge or explicit learning of the system dynamics. Starting from SafeOpt [4], [5], various algorithms have been proposed that, for instance, try to enable global exploration [10], [11], make the algorithm more adaptive [12], or increase the effectiveness of the exploration strategy [13]. All these variations have in common that they rely on GP inference. The limited scalability has also been addressed [14], [15]. Here, improvements in scalability are mainly connected to better handling the discretization of the parameter space required by the standard SafeOpt algorithm, and

Supported by NWO VIDI 15698 and ECSEL 101007311 (IMOCO4.E).

[1]Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland dominik.baumann@aalto.fi

[2]Department of Information Technology, Uppsala University, Uppsala, Sweden

[3]Department of Control Systems and Mechatronics, Wrocław University of Science and Technology, Wrocław, Poland {krzysztof.kowalczyk, pawel.wachel}@pwr.edu.pl

[4]Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands k.tiels@tue.nl

Accepted final version. To appear in *Proc. of the IEEE Conference on Decision and Control, 2023.*

the algorithms still rely on GP inference. Thus, all of them could be combined with the estimator and bounds we propose in this work, lowering their computational complexity. The Nadaraya-Watson estimator has been leveraged in prior control-related works, for instance, in system identification [16]–[19]. Nevertheless, we are not aware of any prior work leveraging the Nadaraya-Watson estimator for safe learning.

II. Problem setting

We consider a dynamical system

$$\mathrm{d}x(t) = z(x(t), u(t))\,\mathrm{d}t$$

with state space $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ and input $u(t) \in \mathbb{R}^m$. For this system, we want to learn a policy π, parametrized by policy parameters $a \in \mathcal{A} \subseteq \mathbb{R}^d$ that determines the control input $u(t)$ given the current system state $x(t)$. The quality of the policy is evaluated by an unknown reward function $f : \mathcal{A} \to \mathbb{R}$. In general, we can solve these kinds of problems through reinforcement learning. However, this would also mean that during exploration, we would allow for arbitrary actions which might lead to the violation of safety constraints. We define safety in the form of constraint functions $g_i : \mathcal{A} \to \mathbb{R}$ that should always be above zero, with $i \in \mathcal{I}_\mathrm{g} = \{1, \dots, q\}$. For instance, a constraint could be the distance of a robot to an obstacle. We can then write the constrained optimization problem as

$$\max_{a \in \mathcal{A}} f(a) \text{ subject to } g_i(a) \geq 0 \text{ for all } i \in \mathcal{I}_\mathrm{g}. \tag{1}$$

Solving (1) without knowledge about the system dynamics and reward and constraint functions is generally unfeasible. Thus, we need to make some assumptions. First, without any prior knowledge, we have no chance to choose parameters a for a first experiment for which we can be certain that they are safe. Thus, we assume that we have an initial set of safe parameters.

Assumption 1: A set $S_0 \subset \mathcal{A}$ of safe parameters is known. That is, for all parameters a in S_0 we have $g_i(a) \geq 0$ for all $i \in \mathcal{I}_\mathrm{g}$ and $S_0 \neq \varnothing$.

In robotics, for instance, we often have simulation models available. As these models cannot perfectly capture the real world, they are insufficient to solve (1). However, they may provide us with a safe initial parametrization. Generally, the set of initial parameters can have arbitrarily bad performance regarding the reward. Thus, if we imagine a robot manipulator that shall reach a target without colliding with an obstacle, a trivial set of safe parameters would barely move the arm from its initial position. While this would not solve the task, it would be sufficient as a starting point for our exploration.

Ultimately, we want to estimate f and g_i from data. Thus, we require some measurements from both after we do experiments.

Assumption 2: After each experiment, we receive measurements $\hat{f}(a) = f(a) + \omega_0$ and $\hat{g}_i(a) = g_i(a) + \omega_i$ for all $i \in \mathcal{I}_\mathrm{g}$, where ω_i, with $i \in \{0, \dots, q\}$, are independent and identically distributed σ-sub-Gaussian random variables.

Lastly, we require a regularity assumption about the functions f and g_i.

Assumption 3: The functions f and g_i, for all $i \in \mathcal{I}_\mathrm{g}$, are Lipschitz-continuous with known Lipschitz constant $L < \infty$.

Assuming Lipschitz-continuity is relatively common in control and the safe learning literature. Some approaches get around assuming to know the Lipschitz constant. However, they often require knowledge of an upper bound of the norm of f and g_i in a reproducing kernel Hilbert space instead. We argue that having an upper bound on the Lipschitz constant is more intuitive. Generally, the Lipschitz constant can also be approximated from data [20].

Note that we clearly could assume individual Lipschitz constants L_i. However, for simplicity, we assume a common Lipschitz constant L, which would be the maximum of the individual L_i.

III. Safe learning algorithm

CoLSafe mainly follows the structure of the popular SafeOpt algorithm but replaces the GPs used to model reward and constraint functions with the Nadaraya-Watson estimator. In the following section, we show that this estimator can provide similar guarantees as [5] showed for GPs. Then, we embed the estimates in the overall algorithm.

A. Nadaraya-Watson estimator

At every iteration, we receive noisy measurements of the reward and constraint functions. From these measurements, we seek to estimate the reward function f and the constraint functions g_i using the Nadaraya-Watson estimator. For notational convenience, let us introduce a selector function

$$h(a, i) := \begin{cases} f(a) & \text{if } i = 0 \\ g_i(a) & \text{if } i \in \mathcal{I}_\mathrm{g}, \end{cases} \tag{2}$$

and the set $\mathcal{I} = \{0\} \cup \mathcal{I}_\mathrm{g}$. Then, we can get estimates of h at iteration n,

$$\mu_n(a', i) := \sum_{t=1}^{n} \frac{K_\lambda(a', a_t)}{\kappa_n(a')} \hat{h}_t(a, i), \tag{3}$$

where

$$\kappa_n(a') := \sum_{t=1}^{n} K_\lambda(a', a_t),$$

$$K_\lambda(a, a') := \frac{1}{c_K} K\left(\frac{\|a - a'\|}{\lambda}\right),$$

$K(\cdot)$ is the kernel function, λ the bandwidth parameter, c_K a constant, and $\hat{h}_t(a, i)$ the measurements at iteration t. The kernel function needs to meet the following assumption.

Assumption 4: The kernel $K : \mathbb{R}^d \to \mathbb{R}$ is positive definite and bounded such that for any $v \in \mathbb{R}^d$ and some $c_\mathrm{K} < \infty$, $0 \leq K(v) \leq c_\mathrm{K}$, and $K(v) = 0$ for all $\|v\| > 1$.

The computation time of the Nadaraya-Watson estimator scales with $\mathcal{O}(n)$ with the number of data points for a naive implementation and with $\mathcal{O}(\log(n)^d)$ if we allow for some pre-processing [9, Thm. 4]. Thus, it is significantly more efficient than using GPs whose computation time scales with $\mathcal{O}(n^3)$.

Nevertheless, we require theoretical worst-case bounds if we seek to leverage the estimate (3) for safe learning. Errors in (3) stem from trying to estimate an unknown function from finitely many data points and the measurement noise defined in Assumption 2. We start by deriving bounds induced by the noise. The following lemma is a slight variation of [21, Lem. 1].

Lemma 1: Let $\{v_t : t \in \mathbb{N}\}$ be a bounded stochastic process and $\{\omega_t : t \in \mathbb{N}\}$ be an i.i.d. sub-Gaussian stochastic process, i.e., there exists a $\sigma > 0$ such that, for any $\gamma \in \mathbb{R}$, and any $t \in \mathbb{N}$

$$\mathbb{E}[\exp(\gamma \omega_t)] \leq \exp\left(\frac{\gamma^2 \sigma^2}{2}\right). \tag{4}$$

For any $\eta \in \mathbb{R}$, define

$$\omega_n(\eta) \coloneqq \exp\left(\sum_{t=1}^{n} \frac{\eta \omega_t v_t}{\sigma} - \frac{1}{2}\eta^2 v_t^2\right).$$

Then, $\mathbb{E}[\omega_n(\eta)] \leq 1$.

Proof: Let

$$D_t \coloneqq \exp\left(\frac{\eta \omega_t v_t}{\sigma} - \frac{1}{2}\eta^2 v_t^2\right).$$

Clearly, $w_n = D_1 D_2 \dots D_n$. Note, that

$$\begin{aligned}\mathbb{E}[D_t \mid v_t] &= \mathbb{E}\left[\frac{\exp\left(\frac{\eta \omega_t v_t}{\sigma}\right)}{\exp\left(\frac{1}{2} v_t^2 \eta^2\right)} \,\middle|\, v_t\right] \\ &= \frac{\mathbb{E}\left[\exp\left(\frac{\eta \omega_t v_t}{\sigma}\right) \mid v_t\right]}{\exp\left(\frac{1}{2} v_t^2 \eta^2\right)}.\end{aligned}$$

Hence, due to (4),

$$\mathbb{E}[D_t \mid v_t] \leq \frac{\exp\left(\frac{\left(\frac{\eta v_t}{\sigma}\right)^2 \sigma^2}{2}\right)}{\exp\left(\frac{1}{2} v_t^2 \eta^2\right)} = 1.$$

Next, for every $n \in \mathbb{N}$,

$$\begin{aligned}\mathbb{E}[\omega_n(\eta) \mid v_n] &= \mathbb{E}[D_1 \dots D_{n-1} D_n \mid v_n] \\ &= D_1 \dots D_{n-1}\, \mathbb{E}[D_n \mid v_n] \leq \omega_{n-1}(\eta).\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{E}[\omega_n(\eta)] &= \mathbb{E}[\mathbb{E}[\omega_n(\eta) \mid v_n]] \\ &\leq \mathbb{E}[\omega_{n-1}] \leq \dots \leq \mathbb{E}[\omega_1(\eta)] \\ &= \mathbb{E}[\mathbb{E}[D_1 \mid v_1]] \leq 1,\end{aligned}$$

which completes the proof. ∎

With this result, we can now derive bounds for the summation of noise terms as in (3). The following result is a variation of [21, Thm. 3].

Lemma 2: Consider v_t and ω_t as in Lemma 1. Further, let $S_n \coloneqq \sum_{t=1}^{n} v_t \omega_t$ and $V_n \coloneqq \sum_{t=1}^{n} v_t^2$. Then, for any $n \in \mathbb{N}$ and $0 < \delta < 1$, with probability at least $1 - \delta$,

$$|S_n| \leq \sqrt{2\sigma^2 \log(\delta^{-1}\sqrt{1 + V_n})(1 + V_n)}.$$

Proof: Without loss of generality, let $\sigma = 1$. For any $\eta \in \mathbb{R}$, let

$$\omega_n(\eta) \coloneqq \exp\left(\eta S_n - \frac{1}{2}\eta^2 V_n\right).$$

From Lemma 1, we note that for any $\eta \in \mathbb{R}$, $\mathbb{E}[\omega_n(\eta)] \leq 1$. Let now H be a $\mathcal{N}(0, 1)$ random variable, independent of all other variables. Clearly, $\mathbb{E}[\omega_n(H) \mid H] \leq 1$. Define

$$\omega_n \coloneqq \mathbb{E}[\omega_n(H) \mid v_t, \omega_t : t \in \mathbb{N}].$$

Then, $\mathbb{E}[\omega_n] \leq 1$ since $\mathbb{E}[\omega_n] = \mathbb{E}[\mathbb{E}[\omega_n(H) \mid v_t, \omega_t : t \in \mathbb{N}]] = \mathbb{E}[\omega_n(H)] = \mathbb{E}[\mathbb{E}[\omega_n(H) \mid H]] \leq 1$. We can also express ω_n directly as

$$\begin{aligned}\omega_n &= \frac{1}{\sqrt{2\pi}} \int \exp\left(\eta S_n - \frac{1}{2}\eta^2 V_n\right) \exp\left(\frac{-\eta^2}{2}\right) \mathrm{d}\eta \\ &= \frac{1}{\sqrt{2\pi}} \int \exp\left(-\frac{1}{2}(V_n + 1)\eta^2 + S_n \eta\right) \mathrm{d}\eta,\end{aligned}$$

which further gives

$$\begin{aligned}\omega_n &= \frac{1}{\sqrt{2\pi}} \int \exp\left(-\frac{1}{2}\frac{\left(\eta - \frac{1}{1+V_n} S_n\right)^2}{(1 + V_n)^{-1}}\right) \exp\left(\frac{1}{2}\frac{S_n^2}{1 + V_n}\right) \\ &= \frac{1}{\sqrt{1 + V_n}} \exp\left(\frac{1}{2}\frac{S_n^2}{1 + V_n}\right) \int \frac{\sqrt{1 + V_n}}{\sqrt{2\pi}} \\ &\quad \exp\left(-\frac{1}{2}\left(\frac{\eta - \frac{1}{1+V_n} S_n}{(1 + V_n)^{-\frac{1}{2}}}\right)^2\right) \mathrm{d}\eta \\ &= \frac{1}{\sqrt{1 + V_n}} \exp\left(\frac{1}{2}\frac{S_n^2}{1 + V_n}\right).\end{aligned}$$

Therefore, $\mathbb{P}[\delta \omega_n \geq 1]$ equals

$$\begin{aligned}&\mathbb{P}\left[\frac{\delta}{\sqrt{1 + V_n}} \exp\left(\frac{1}{2}\frac{S_n^2}{1 + V_n}\right) \geq 1\right] \\ =&\mathbb{P}\left[\exp\left(\frac{1}{2}\frac{S_n^2}{1 + V_n}\right) \geq \frac{\sqrt{1 + V_n}}{\delta}\right] \\ =&\mathbb{P}\left[\frac{S_n^2}{1 + V_n} \geq 2\log\left(\frac{\sqrt{1 + V_n}}{\delta}\right)\right] \\ =&\mathbb{P}\left[S_n^2 \geq 2\log\left(\frac{\sqrt{1 + V_n}}{\delta}\right)(1 + V_n)\right].\end{aligned}$$

Recall now that $\mathbb{E}[\omega_n] \leq 1$. Hence, due to Markov's inequality, we have

$$\mathbb{P}[\delta \omega_n \geq 1] \leq \delta\, \mathbb{E}[\omega_n] \leq \delta,$$

which completes the proof. ∎

With this, we can provide the required bounds (cf. [22]).

Lemma 3: Under Assumptions 2–4, we have, for all $n \geq 0$, $a' \in \mathcal{A}$, and all $i \in \mathcal{I}$, with probability at least $1-\delta$,

$$|h(a',i) - \mu_n(a',i)| < \beta_n(a',i),$$

where

$$\beta_n(a',i) \coloneqq L\lambda + 2\sigma \frac{\alpha_n(a',\delta)}{\kappa_n(a')},$$

and

$$\alpha_n(a',\delta) \coloneqq \begin{cases} \sqrt{\log\left(\frac{\sqrt{2}}{\delta}\right)}, & \text{if } \kappa_n(a') \leq 1 \\ \sqrt{\kappa_n(a') \log\left(\frac{\sqrt{1+\kappa_n(a')}}{\delta}\right)}, & \text{if } \kappa_n(a') > 1. \end{cases}$$

Proof: Due to Assumption 4, we have $\frac{1}{c_K} K(\cdot) \leq 1$. Hence, without loss of generality, we assume in the following that K_λ is bounded by 1. Following Assumption 2, we further have (cf. [23])

$$\begin{aligned} &\left| \sum_{t=1}^{n} \frac{K_\lambda(a',a_t)\hat{h}_t(a,i)}{\kappa_n(a')} \hat{h}_t(a,i) - h(a',i) \right| \\ \leq &\sum_{t=1}^{n} \theta_t |h(a_t,i) - h(a',i)| + \left| \sum_{t=1}^{n} \theta_t \omega_t \right|, \end{aligned} \tag{5}$$

where $\theta_t \coloneqq \frac{K_\lambda(a',a_t)}{\kappa_n(a')}$ and ω_t the measurement noise at iteration t. Note, that $\sum_{t=1}^{n} \theta_t = 1$. Due to Assumption 4, if $K_\lambda(a',a_t) > 0$, then $\frac{\|a'-a_t\|}{\lambda} \leq 1$. Therefore, if $K_\lambda(a',a_t) > 0$, cf. Assumption 3,

$$|h(a_t,i) - h(a',i)| \leq L\|a' - a_t\| \leq L\lambda,$$

and since the weights θ_t sum to 1,

$$\sum_{t=1}^{n} \theta_t |h(a_t,i) - h(a',i)| \leq L\lambda.$$

Finally, for the noise term in (5), observe that

$$\left| \sum_{t=1}^{n} \theta_t \omega_t \right| = \frac{1}{\kappa_n(a')} \left| \sum_{t=1}^{n} K_\lambda(a',a_t)\omega_t \right|.$$

According to Lemma 2, this term is upper bounded, with probability $1-\delta$, by

$$\frac{1}{\kappa_n(a')} \sigma \times \sqrt{2 \log\left(\delta^{-1} \sqrt{1 + \sum_{t=1}^{n} K_\lambda^2(a',a_t)} \right) \left(1 + \sum_{t=1}^{n} K_\lambda^2(a',a_t) \right)}.$$

Furthermore, since $K_\lambda(a',a_t) \leq 1$ (cf. Assumption 4), we obtain

$$\begin{aligned} &\frac{1}{\kappa_n(a')} \left| \sum_{t=1}^{n} K_\lambda(a',a_t)\omega_t \right| \\ \leq &\sigma \sqrt{2 \log(\delta^{-1}\sqrt{1+\kappa_n(a')})} \frac{\sqrt{1+\kappa_n(a')}}{\kappa_n(a')}. \end{aligned}$$

Observe next that, if $\kappa_n(a') > 1$,

$$\frac{\sqrt{1+\kappa_n(a')}}{\kappa_n(a')} < \frac{\sqrt{2\kappa_n(a')}}{\kappa_n(a')} = \frac{\sqrt{2}}{\sqrt{\kappa_n(a')}}.$$

Therefore, with probability at least $1-\delta$, for $\kappa_n(a') > 1$,

$$\begin{aligned} &\frac{1}{\kappa_n(a')} \left| \sum_{t=1}^{n} K_\lambda(a',a_t)\omega_t \right| \\ \leq &\frac{2\sigma}{\kappa_n(a')} \sqrt{\kappa_n(a') \log(\delta^{-1}\sqrt{1+\kappa_n(a')})}, \end{aligned}$$

whereas for $0 < \kappa_n(a') \leq 1$,

$$\begin{aligned} &\frac{1}{\kappa_n(a')} \left| \sum_{t=1}^{n} K_\lambda(a',a_t)\omega_t \right| \\ \leq &\frac{\sigma}{\kappa_n(a')} \sqrt{2\log(\delta^{-1}\sqrt{1+\kappa_n(a')})} \sqrt{1+\kappa_n(a')} \\ \leq &\frac{2\sigma}{\kappa_n(a')} \sqrt{\log\left(\frac{\sqrt{2}}{\delta}\right)}, \end{aligned}$$

which completes the proof. ∎

With these bounds, we can next present the safe learning algorithm.

B. The algorithm

Starting from the initial safe seed given by Assumption 1, we can start a first experiment and receive a measurement of reward and constraint functions. Following [5], we then leverage the functions' Lipschitz-continuity to generate a set of parameters a that is safe with high probability. For this, we first construct confidence intervals as

$$Q_n(a,i) = [\mu_{n-1}(a,i) \pm \beta_{n-1}(a,i)]. \tag{6}$$

As the SafeOpt algorithm requires that the safe set does not shrink, we further define the contained set as

$$C_n(a,i) = C_{n-1} \cap Q_n(a,i), \tag{7}$$

where $C_0(a,i)$ takes values in $[0,\infty)$ for all $a \in S_0$ and takes values in $\mathbb{R}$ for all $a \in \mathcal{A} \setminus S_0$. This enables us to define lower and upper bounds as $l_n(a,i) \coloneqq \min C_n(a,i)$ and $u_n(a,i) \coloneqq \max C_n(a,i)$ with which we can update the safe set:

$$S_n = \bigcap_{i \in \mathcal{I}_g} \bigcup_{a \in S_{n-1}} \{a' \in \mathcal{A} \mid l_n(a,i) - L\|a - a'\| \geq 0\}. \tag{8}$$

By sampling only from this set, we can guarantee that each experiment will be safe with high probability. However, we also seek to optimize the policy. Thus, for a meaningful exploration strategy, we define two additional sets [5]: parameters that are likely to yield a higher reward than our current optimum (potential maximizers, M_n), and parameters that are likely to enlarge S_n (potential expanders, G_n). Formally, we define them as

$$M_n \coloneqq \{a \in S_n \mid u_n(a,0) \geq \max_{a' \in S_n} l_n(a',0)\} \tag{9}$$

$$G_n \coloneqq \{a \in S_n \mid e_n(a) > 0\}, \tag{10}$$

with

$$e_n(a) \coloneqq |\{a' \in \mathcal{A} \setminus S_n \,|\exists i \in \mathcal{I}_\mathrm{g} : u_n(a,i) - L\|a - a'\| \geq 0\}|.$$

Given those sets, we select our next sample location as

$$a_n = \operatorname*{arg\,max}_{a \in M_n \cup G_n} \max_{i \in \mathcal{I}} w_n(a,i), \tag{11}$$

with $w_n(a,i) = u_n(a,i) - l_n(a,i)$, which is basically a variant of the upper confidence bound algorithm [24]. Further, we can, at any iteration, obtain an estimate of the optimum through

$$\hat{a}_n = \operatorname*{arg\,max}_{a \in S_n} l_n(a,0). \tag{12}$$

The entire algorithm is summarized in Algorithm 1.

Lastly, we provide the overall safety guarantees of CoLSafe. For these guarantees, we assume a finite parameter set $\mathcal{A}$. However, heuristic extensions to continuous domains exist [14].

Theorem 1: Under Assumptions 1–4, when following Algorithm 1, we have for all $n \geq 0$, with probability at least $1 - \delta$, that $g_i(a_n) \geq 0$ for all $i \in \mathcal{I}_g$, i.e., the algorithm is safe.

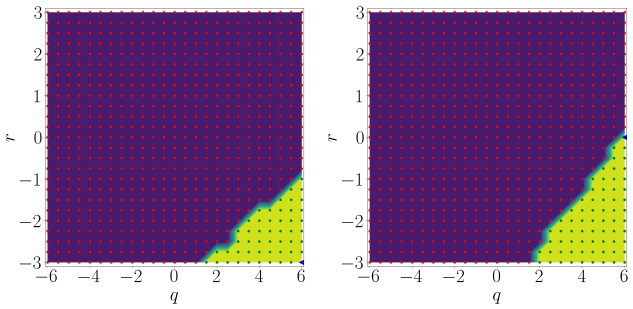
Proof: Lemma 3 provides equivalent bounds to the ones derived in [5, Lem. 1]. Thus, the proof follows from [5, Lem. 11]. ∎

Algorithm 1 Pseudocode of CoLSafe.

1: **Input:** Domain $\mathcal{A}$, Safe seed S_0, Lipschitz constant L
2: **for** $n = 1, 2, \ldots$ **do**
3: Update safe set with (8)
4: Update set of potential maximizers with (9)
5: Update set of potential expanders with (10)
6: Select a_n with (11)
7: Receive measurements $\hat{f}(a_n)$, $\hat{g}_i(a_n)$, for all $i \in \mathcal{I}_\mathrm{g}$
8: Update Nadaraya-Watson estimator (3) with new data
return Best guess (12)

IV. Numerical experiments

In the evaluation, we seek to demonstrate that CoLSafe is generally capable of learning control policies for dynamical systems and that its computations are more lightweight than those of SafeOpt. Thus, we take a simulation model that has been used in earlier work on SafeOpt [11], run both CoLSafe and SafeOpt, and compare the results. In particular, both algorithms are supposed to learn a control policy for a simulation model of a seven-degrees-of-freedom Franka robot arm. The goal is to let the arm of the robot reach a desired set point without colliding with an obstacle. We consider a feedback linearization approach based on an approximate system model and design a linear quadratic regulator (LQR) for the then approximately linear system. Since the system model is not perfect, the LQR does not achieve optimal performance. However, it can provide us with a safe seed. Starting from there, we seek to optimize the cost matrices of the LQR to compensate for the model mismatch. The parameter space for this experiment is $d = 2$. For further details on the setup, we refer the reader to [11].



(a) CoLSafe after 410 episodes. (b) SafeOpt after 410 episodes.

Fig. 1: Performance of CoLSafe and SafeOpt on the simulated Franka robot. *Both algorithms can similarly explore the safe region while the computational footprint of CoLSafe is significantly lower. The yellow regions mark the safe set, the blue triangle the current optimum, and q and r are the tuning parameters of the controller.*

For SafeOpt, we adopt the parameter settings from [11], i.e., we use a Matérn kernel with $\nu = 1.5$. For CoLSafe, we use the same kernel with a length scale of 0.1, set the bandwidth parameter $\lambda = 0.5$, and the Lipschitz constant $L = 1.75$. To meet Assumption 4, we only use the output of the kernel for the Nadaraya-Watson estimator if $\|a - a'\| < 1$; else, we set it to 0.

We run SafeOpt and CoLSafe and compare the results after 410 episodes in Fig. 1. Both algorithms expand the safe set beyond the initial region and find better controller parameters. SafeOpt can explore a larger part of the state space within the 410 episodes, i.e., CoLSafe is more conservative. However, SafeOpt requires more time for each individual optimization step. We show this in Fig. 2. We measure the time for updating the safe set and suggesting the next policy parameters for both algorithms. Especially during later iterations, when the data set is larger, SafeOpt requires significantly more time for each update step. In particular, in the last iteration, SafeOpt needs more than 1 h to suggest the next sample point, while CoLSafe needs only 12.5 s. The times were measured on a standard laptop.

This evaluation suggests an interesting trade-off. While the computations for CoLSafe are computationally cheaper, the bounds are more conservative, and thus, only a smaller safe set can be explored. That is, when computing resources are not a concern, and the dimensionality and expected number of data samples are relatively low, but we need to explore the largest possible safe set to increase the chance of finding the global optimum, SafeOpt is the better choice. However, for higher dimensional systems and restricted computing resources, e.g., when learning policies on embedded devices, CoLSafe

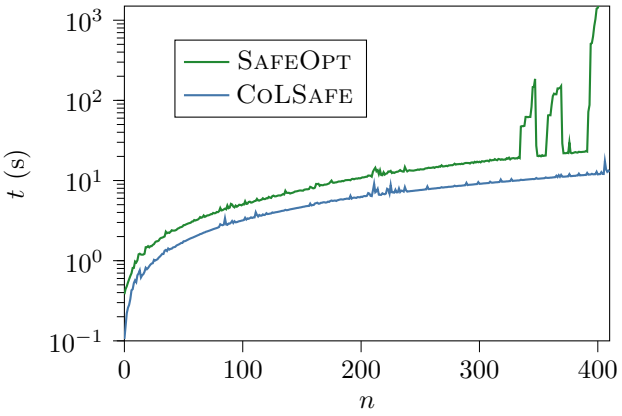


Fig. 2: Time complexity of SafeOpt and CoLSafe. *We measure the time t for updating the safe set and suggesting the next candidate point in each iteration n. For higher iterations, i.e., more data,* SafeOpt *requires significantly more time. Note the logarithmic scaling of the y-axis.*

is superior.

V. Conclusions

We propose a novel algorithm for safe exploration in reinforcement learning. Under assumptions that are comparable to other safe learning algorithms, such as the popular SafeOpt algorithm, we can provide similar high-probability safety guarantees. However, our algorithm has significantly lower computational complexity, simplifying its use on embedded devices and for high-dimensional systems. We also see that we trade this lower computational complexity for a more conservative exploration strategy. Thus, in practice, it depends on the application and the available computational resources which algorithm is preferable.

Herein, we basically adopted the SafeOpt algorithm and replaced the Gaussian process estimates with the Nadaraya-Watson estimator. For future work, it might be interesting to combine those estimates also with more recent advances of SafeOpt such as [11], [12], [14], [15].

Further, we here considered a constant bandwidth parameter λ. This parameter could also be tuned to make the algorithm less conservative, and that way, potentially come closer to the performance of SafeOpt under the same amount of samples.

Acknowledgements

The authors would like to thank Bhavya Sukhija for support with the robot simulations.

References

[1] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[2] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking deep reinforcement learning for continuous control," in *International Conference on Machine Learning*, 2016, pp. 1329–1338.

[3] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[4] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe exploration for optimization with Gaussian processes," in *International Conference on Machine Learning*, 2015, pp. 997–1005.

[5] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics," *Machine Learning*, pp. 1–35, 2021.

[6] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*. MIT press Cambridge, MA, 2006.

[7] E. A. Nadaraya, "On estimating regression," *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[8] G. S. Watson, "Smooth regression analysis," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 359–372, 1964.

[9] N. S. Rao and V. A. Protopopescu, "On PAC learning of functions with smoothness properties using feedforward sigmoidal networks," *Proc. IEEE*, vol. 84, no. 10, pp. 1562–1569, 1996.

[10] D. Baumann, A. Marco, M. Turchetta, and S. Trimpe, "GoSafe: Globally optimal safe robot learning," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 4452–4458.

[11] B. Sukhija, M. Turchetta, D. Lindner, A. Krause, S. Trimpe, and D. Baumann, "GoSafeOpt: Scalable safe exploration for global optimization of dynamical systems," *Artificial Intelligence*, vol. 320, p. 103922, 2023.

[12] C. König, M. Turchetta, J. Lygeros, A. Rupenyan, and A. Krause, "Safe and efficient model-free adaptive control via Bayesian optimization," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 9782–9788.

[13] A. Wachi, Y. Sui, Y. Yue, and M. Ono, "Safe exploration and optimization of constrained MDPs using Gaussian processes," in *AAAI Conference on Artificial Intelligence*, 2018, pp. 6548–6555.

[14] R. R. Duivenvoorden, F. Berkenkamp, N. Carion, A. Krause, and A. P. Schoellig, "Constrained Bayesian optimization with particle swarms for safe adaptive controller tuning," in *IFAC World Congress*, 2017, pp. 11 800–11 807.

[15] Y. Sui, V. Zhuang, J. Burdick, and Y. Yue, "Stagewise safe Bayesian optimization with Gaussian processes," in *International Conference on Machine Learning*, 2018, pp. 4781–4789.

[16] E. Schuster and S. Yakowitz, "Contributions to the theory of nonparametric regression, with application to system identification," *The Annals of Statistics*, pp. 139–149, 1979.

[17] A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Delyon, L. Ljung, J. Sjöberg, and Q. Zhang, "Nonlinear black-box models in system identification: Mathematical foundations," *Automatica*, vol. 31, no. 12, pp. 1725–1750, 1995.

[18] L. Ljung, "Some aspects on nonlinear system identification," in *IFAC Symposium on Identification and System Parameter Estimation*, 2006, pp. 553–564.

[19] G. Mzyk and P. Wachel, "Wiener system identification by input injection method," *International Journal of Adaptive Control and Signal Processing*, vol. 34, no. 8, pp. 1105–1119, 2020.

[20] G. Wood and B. Zhang, "Estimation of the Lipschitz constant of a function," *Journal of Global Optimization*, vol. 8, pp. 91–103, 1996.

[21] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Online least squares estimation with self-normalized processes: An application to bandit problems," *arXiv preprint arXiv:1102.2670*, 2011.

[22] P. Wachel, K. Kowalczyk, and C. R. Rojas, "Decentralized diffusion-based learning under non-parametric limited prior knowledge," *European Journal of Control*, 2022, under review.

[23] S. Dean and B. Recht, "Certainty equivalent perception-based control," in *Learning for Dynamics and Control*, 2021, pp. 399–411.

[24] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," *IEEE Trans. Inform. Theory*, vol. 58, no. 5, 2012.