2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Online Model-free Safety Verification for Markov Decision Processes Without Safety Violation

Abhijit Mazumdar, Rafal Wisniewski and Manuela L. Bujorianu

***Abstract*— In this paper, we consider the problem of safety assessment for Markov decision processes *without* explicit knowledge of the model. We aim to learn probabilistic safety specifications associated with a given policy without compromising the safety of the process. To accomplish our goal, we characterize a subset of the state-space namely *proxy set*, which contains the states that are *near* in a probabilistic sense to the *forbidden set* consisting of all unsafe states. We compute the safety function using the single-step temporal difference method. To this end, we relate the safety function computation to that of the value function estimation using temporal difference learning. Since the given control policy could be unsafe, we use a safe *baseline sub-policy* to generate data for learning. We then use an *off-policy* temporal difference learning method with *importance sampling* to learn the safety function corresponding to the given policy. Finally, we demonstrate our results using a numerical example.**

***Index Terms*— Online safety verification, Markov decision processes, reinforcement learning, temporal difference, proxy set.**

I. Introduction

In safety-critical systems, assessing safety associated with a control policy is crucial during the deployment of the control policy. Safety verification for dynamical system is usually studied in two settings: worst-case [1]–[3] or stochastic [4]–[8]. In the worse-case set up, safety corresponds to the property of never visiting the unsafe region. While the systems may be subjected to uncertain disturbance input, a hard upper bound on the disturbance input is assumed to be known. In the stochastic setup, safety is defined as the probability of reaching the unsafe region with a small probability below a prescribed margin.

If the operational environment is changing or no prior information regarding the system model, or the environment is known, then safety needs to be verified during operation [9]–[11]. This set-up is called *online* safety verification.

The works described above are model-based, i.e., an appropriate system model is required. Data-driven safety verification methods are getting attention, of late, as they eliminate the requirement of a model of the systems [12]–[17]. Among these works, [17] considers a probabilistic safety notion, whereas [12]–[16] consider the worse-case safety definition. For systems with discrete-time and continuous states, [12], [13] proposed a data-driven method based on barrier certificate to verify safety formally. In [14], a data-driven approach with formal guarantees is presented for networks of discrete-time sub-systems. To this end, a sub-barrier function for each sub-system is computed, then the overall barrier function is derived from the individual sub-barrier function. Further, [13], [15], [16] converts the problem of finding barrier certificate as a robust convex problem.

Main Contributions: In the existing works on data-driven safety verification, it is assumed that an existing data set is available. This is called the *offline* set-up. However, many times, safety needs to be verified during the operating phase of a system in an *online* fashion [9]–[11]. If no prior data is available, the data-driven online set-up becomes more challenging as safety can be jeopardized during the learning.

To the best of our knowledge, the existing data-driven methods, except for [17], are offline. In this paper, we develop an online safety verification method for stochastic systems without jeopardizing the system's safety. We consider a Markov decision process framework to represent the stochastic dynamics. Unlike [17], in this work, we do not need to know even a partial model of the system. This relaxation makes the problem much harder compared to [17]. We use a single-step temporal difference method (TD(0)) to learn the safety function corresponding to a given *target control policy* π. If the TD(0) method is used naively, then the target policy π, which needs to be assessed, must be used. However, since the policy π is arbitrary and could be unsafe, employing it during the learning phase can lead to violation of the safety constraints. To circumvent this issue, we use an off-policy TD(0) method with *importance sampling* [18], [19]. We assume that at least one *safe baseline sub-policy* for each state of a sub-set, called *proxy set*, of the state-space is known. This assumption is an essential requirement in safe reinforcement learning. The safe baseline sub-policy is needed to use the off-policy TD (0) method to learn the safety function without violating the safety constraints.

The organization of the paper is as follows. In Section II, we set up the relevant notations. We present the system description and the problem formulation in Section III. The main results are described in Section IV. In Subsection IV-A, we presented the algorithm following a thorough discussion. With a numerical example, we demonstrate our results in Section V. Finally, in Section VI, we conclude the paper and highlight a future extension to this work.

II. Notations

We consider an MDP with a set of finite states denoted by $\mathcal{X}$ and a finite set of finite actions represented by $\mathcal{A}$. We consider a sample space Ω of all sequences of the form $\omega =$

The work of the first and the second authors has been supported by the Poul Due Jensens Foundation under project SWIft.The work of the third author has been funded by the EPSRC project EP/R006865/1: Interface reasoning for interacting systems (IRIS).

A. Mazumdar and R. Wisniewski are with the Section of Automation & Control, Aalborg University, 9220 Aalborg East, Denmark (e-mail: {abma, raf}@es.aau.dk).

Manuela L. Bujorianu is with the Department of Computer Science, University College London, UK (e-mail: l.bujorianu@ucl.ac.uk).

$(x_0, a_0, x_1, a_1, \ldots) \in (\mathcal{X} \times \mathcal{A})^\infty$ with $x_i \in \mathcal{X}$ and $a_i \in \mathcal{A}$. By $\mathcal{F}$, we denote the σ-algebra generated by coordinate mappings: $X_t(\omega) = x_t$ and $A_t(\omega) = a_t$. By upper case X_t and A_t, we denote random variables, while we use x_t and a_t for deterministic values, i.e., their realizations, at time-step t. Further, we assume that the initial state X_0 has a distribution μ. In this work, we consider stationary policies, i.e., maps $\pi : \mathcal{X} \to \Delta(\mathcal{A})$, with $\Delta(\mathcal{A}) = \{(p_1, \ldots, p_{|\mathcal{A}|}) \in [0,1]^{\mathcal{A}} \mid p_1 + \ldots + p_{|\mathcal{A}|} = 1\}$. A sub-policy π' for a subset of $W \subseteq \mathcal{X}$ is defined as $\pi' : W \to \Delta(\mathcal{A})$. For a fixed initial distribution μ and a policy π, we define recursively the probability $\mathbb{P}_\pi^\mu$ on $\mathcal{F}$ by

$$\mathbb{P}_\pi^\mu[X_1 = x] = \mu(x)$$
$$\mathbb{P}_\pi^\mu[A_t = a \mid X_t = x] = \pi(a|x)$$
$$\mathbb{P}_\pi^\mu[X_{t+1} = y \mid X_t = x, A_t = a] = p(x, a, y)$$

We write $\mathbb{P}_\pi^y := \mathbb{P}_\pi^{\delta_y}$ for the delta distribution concentrated at y. The expectation with respect to $\mathbb{P}_\pi^y$ is denoted $\mathbb{E}_\pi^y$.

III. System Description and Problem Formulation

We consider an MDP with a set of states $\mathcal{X}$ and a set of actions $\mathcal{A}$. Suppose the set of states is partitioned into a target set $E \subset \mathcal{X}$, a set of forbidden states U, and $H := \mathcal{X} \setminus (E \cup U)$ be the set of living (taboo) states.

This work deals with probabilistic safety. For any target control policy, or simply target policy π, in order to assess safety, a safety function $S_\pi(x)$ is defined as follows [8].

Definition 1: (Safety Function) For each state $x \in H$, the safety function is the probability that the realizations hit the forbidden set U before reaching the target set E, i.e., for a fixed policy π,

$$S_\pi(x) := \mathbb{P}_\pi^x[\tau_U < \tau_E],$$

where τ_A is the first hitting time of a set A. •

We consider a probabilistic safety notion called p-safe [7], [8]. Following definitions are central to this work and are inspired from [7], [8].

Definition 2: (p-Safe State, p-Safe MDP and p-Safe Policy) For a given policy π, a state $x \in H$ is called p-safe if the safety function does not exceed p, i.e., $S_\pi(x) \leq p$. Similarly, an MDP is called p-safe with a policy π if: $\max_{x \in H} S_\pi(x) \leq p$. In this case, π is called a p-safe policy. •

It should be noted that we use safety and *p-safety* synonymously throughout the paper.

We now formally state the problem that we address in this work as follows.

> *Problem P:* Estimate the safety function for the given *target policy* π without rendering the MDP unsafe, i.e., ensuring that $S_\pi(x) \leq p$ for each state $x \in H$.

Since the target policy, π, could be arbitrary; if we apply it, we might jeopardize the safety of the MDP. Hence, we must use an indirect way to learn about the safety function corresponding to π. In order to solve Problem P, we now introduce a *proxy set* as follows. If we know the proxy states, we can learn the safety function faster than not knowing them.

Definition 3: (Proxy Set) We call the subset $U' \in H$ as a *proxy set* of an MDP, if it has the following properties:

N.1 $\tau_{U'} < \tau_U$, almost surely.

N.2 For all $x \in U'$, there exists $a \in A$ and $y \notin U$ such that $p(x, a, y) > 0$. •

The proxy set U' can be considered a neighborhood of the forbidden set U as the probability of hitting U' before hitting the forbidden set U is 1.

Remark 1: To motivate the need for defining proxy states, we can think of autonomous robot navigation. The robot hits an obstacle only if it crosses certain states (speed, angular velocity, etc.). Not all states directly lead to an unsafe state without visiting other states.

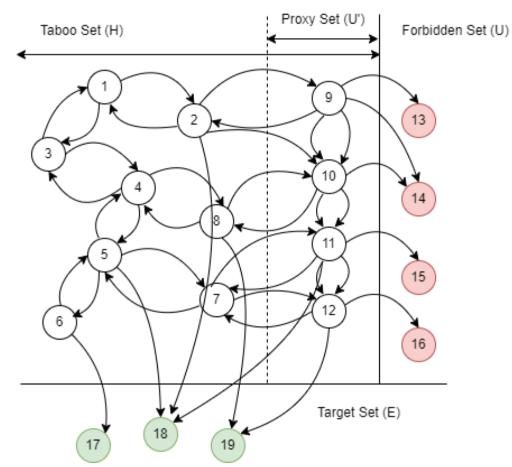

Fig. 1. Illustration of the taboo set, proxy set, forbidden set, and target set.

We now introduce the concept of *safe action* and *safe baseline sub-policy* for the proxy states $x' \in U'$. These will enable us to learn the safety function of a given policy without violating the safety constraint.

Definition 4: (Safe Action) For each proxy state $x' \in U'$, we call an action a safe action, denoted by $a^{safe}(x')$, if

$$p(x', a^{safe}(x'), y) = 0, \forall y \in U.$$

•

Definition 5: (Safe Baseline Sub-Policy) We call a sub-policy defined for the proxy set U' a *safe baseline sub-policy*, denoted by π^S, if

$$S_{\pi^S}(x') \leq p, \forall x' \in U'.$$

•

Throughout the paper, we have the following assumptions:

i) The proxy set U' is given.

ii) A safe baseline sub-policy π^S, for each proxy state $x' \in U'$, is known.

Remark 2: Knowledge of proxy states is not mandatory for the results presented in this paper. However, it makes learning faster than not having knowledge of proxy states.

Remark 3: A safe baseline policy is a standard assumption in safe reinforcement learning. We cannot guarantee safety throughout the learning process without the knowledge of a safe policy.

IV. Online Safety Verification

In this section, we present the algorithm for online safety verification. Before presenting the algorithm for estimating the safety function, we need to establish the following results.

As given in [8], safety function can be expressed as follows.

Lemma 1: ([8]) Suppose, $\tau = \tau_{U \cup E}$ is almost surely finite. The safety function for each state $x \in H$, with a policy π, is then given by

$$S_\pi(x) = \mathbb{E}_\pi^x \sum_{t=0}^{\tau-1} \kappa(X_t, A_t),$$

where $\kappa(x,a) = \sum_{y \in U} p(x,a,y)$. •

Remark 4: If we knew the transition probabilities $p(x,a,y)$ from the proxy set to the forbidden set, as in [17], we could use the standard TD(0) method to estimate the safety function. Since we do not know the transition probabilities, we need to express the safety function as follows. •

Proposition 1: If $\tau = \tau_{U \cup E} < \infty$, almost surely, then the safety function for any state $x \in H$ can be expressed as follows:

$$S_\pi(x) = \mathbb{E}_\pi^x \sum_{t=0}^{\tau-1} c(X_t, A_t), \tag{1}$$

where,

$$c(X_t, A_t) = \begin{cases} 1, & \text{if } X_{t+1} \in U \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Proof: Observe the following:

$$\begin{aligned} &\mathbb{E}_\pi^x[c(X_t,A_t)|X_t = \tilde{x}, A_t = a] \\ &= \mathbb{E}_\pi^x\big[\mathbb{E}_\pi[c(X_t,A_t)|X_t = \tilde{x}, A_t = a]\big] \\ &= \mathbb{E}_\pi^x[\sum_{y \in U} p(\tilde{x},a,y)] \\ &= \mathbb{E}_\pi^x[\kappa(X_t,A_t)|X_t = \tilde{x}, A_t = a] \end{aligned} \tag{3}$$

Thus, the expressions of the safety function given in Lemma 1 and Proposition 1 are equivalent. Hence, using Lemma 1, the safety function can be expressed as given in the Proposition. ■

The following property relates the safety function of proxy states $x' \in U'$ with the safety function of states $x \in H \setminus U'$. This property will be used to ensure safety during the learning phase.

Proposition 2: For any state $x \in H$, the following is true:

$$S_\pi(x) \le \max_{x' \in U'} S_\pi(x').$$

Proof: Following the definition of the safety function, for any $x \in H$, we get the following:

$$\begin{aligned} S_\pi(x) &= \mathbb{P}_\pi^x[\tau_U < \tau_E] \\ &= \sum_{x' \in U'} \mathbb{P}_\pi^{x'}[\tau_U < \tau_E]\mathbb{P}_\pi^x[X_{\tau'} = x'] \text{ (let } \tau' = \tau_{U' \cup E}) \\ &= \sum_{x' \in U'} S_\pi(x')\mathbb{P}_\pi^x[X_{\tau'} = x'] \\ &\le \max_{x' \in U'} S_\pi(x'). \end{aligned}$$

The second equality is a direct consequence of the first property of the proxy states. ■

Remark 5: As a consequence of Proposition 2, if we apply a safe baseline policy only for the proxy states U' and use a target policy π for other states $H \setminus U'$, the MDP will be safe throughout the learning phase. •

A. Safe learning of the safety function

We notice that the safety function in (1) resembles the value function considered in reinforcement learning. The single-step temporal difference method, TD(0), is one of the most widely used methods to compute the value function. Hence, we also use the TD(0) method to estimate the safety function. Suppose $S_t(x)$ is the estimated safety function for state x in the tth learning step, and after applying A_t according to π the process reaches state y. Then, according to the TD(0) method, the update rule for the estimated safety function is as given below:

$$\begin{aligned} &S_{t+1}(x) \leftarrow S_t(x) + \alpha_t(x)[c_t + S_t(y) - S_t(x)], \\ &\text{where, } c_t = \begin{cases} 1, & \text{if } y \in U \\ 0, & \text{if } y \notin U \end{cases}. \end{aligned} \tag{4}$$

In the above expression, $c_t + S_t(y)$ is called the *TD target*.

Now, suppose the learning rate $\alpha_t(x)$ is chosen such that following conditions are satisfied:

$$\begin{aligned} (i) \sum_t^\infty \alpha_t(x) = \infty \\ (ii) \sum_t^\infty \alpha_t^2(x) < \infty. \end{aligned} \tag{5}$$

Then, using the results presented in [20], it can be inferred that $S_t(x)$ converges to the true safety function $S(x)$ for each $x \in H$.

Since the hitting time is finite (almost surely), we consider an episodic temporal difference TD(0) algorithm. In an episodic learning framework, whenever the process hits the terminal states, learning is resumed from an arbitrary initial state. Since the given target policy π could render the MDP unsafe, we use a safe baseline sub-policy π^S for the proxy set to generate data. However, the goal is to learn the safety function $S_\pi(x)$ with the target policy π. Since π^S is safe by definition, the safety is always maintained for the proxy set U'. Further, if we ensure that the safety function for the proxy set U' is less than p, then from Proposition 2, it is made sure that the MDP is safe. The policy that is used to generate necessary data during learning is called the *behavior policy*, denoted by π^b. The *behavior policy* π^b that we use is as follows:

$$\pi^b = \begin{cases} \pi, & \text{for } x \in H \setminus U' \\ \pi^S & \text{for } x \in U'. \end{cases} \tag{6}$$

Since the behavior policy π^b is chosen differently than the target policy π, if we use the standard TD(0) naively, then we would only learn $S_{\pi^b}(x)$, and not $S_\pi(x)$. To resolve this issue, we use a variant of TD(0) with *per-decision importance sampling*, which is an *off-policy* value function estimation method as given in [18]. In this method, for the proxy states $x' \in U'$, the update rule for the estimated safety function takes the following form:

$$S_{t+1}(x') \leftarrow S_t(x') + \alpha_t(x')[\frac{\pi(a|x')}{\pi^S(a|x')}\big(c_t + S_t(y) - S_t(x')\big)], \tag{7}$$

where, c_t is as given in (4) and $\frac{\pi(a|x')}{\pi^S(a|x')}$ is called the *importance-sampling ratio*. The learning rate is kept similar to (5). From the results presented in [19], it follows that $S_{t+1}(x')$ converges to the true safety function $S_\pi(x')$ with the target policy π, almost surely.

From Proposition 2, it is clear that if we ensure that the proxy states $x' \in U'$ are safe, then the states $x \in H \setminus U'$ will be safe irrespective of the sub-policy used for them. However, since we use the policy π that needs to be assessed for any state $x \in H \setminus U'$, the update rule is the standard one as given in (4).

Further, to estimate the safety function with the behavior policy π^b, the update rule is given by:

$$S^b_{t+1}(x) \leftarrow S^b_t(x) + \alpha_t(x)[c_t + S^b_t(y) - S^b_t(x)],$$
$$\text{where, } c_t = \begin{cases} 1, & \text{if } y \in U \\ 0, & \text{if } y \notin U \end{cases}. \tag{8}$$

•

Algorithm 1 : Safe TD(0) with importance sampling:

Input: The given target policy π for which safety is needed to be evaluated, a safe baseline sub-policy π^S, a safe behavior policy π^b, learning rate $\alpha_t(x)$ for each $x \in H$, safety parameter p, proxy set U';
Initialize: $S_1(x)$ for each $x \in H$ arbitrarily, $S_1(x) = 0$ for each $x \in U \cup E$, $t = 1$;
for Episodes ($k = 1, 2, ..., \mathcal{L}$) **do**
    Draw an initial state x uniformly from H;
    for Iterations ($i = 1, 2, ..., \mathcal{T}$) **do**
        if $x \in U'$ **then**
            Apply a safe action $A_t = a^{safe}(x)$ according to the safe baseline sub-policy π^S;
        else
            Apply action A_t according to the target policy π;
        end if
        Observe the new state y, and c_t according to (4);
        if $x \in U'$ **then**
            Update the safety function as follows:

$$S_{t+1}(x) \leftarrow S_t(x) + \alpha_t(x)[\frac{\pi(a|x)}{\pi^S(a|x)}(c_t + S_t(y) - S_t(x))]; \tag{9}$$

        else ($x \in H \setminus U'$)
            Update the safety function as follows:

$$S_{t+1}(x) \leftarrow S_t(x) + \alpha_t(x)[c_t + S_t(y) - S_t(x)]; \tag{10}$$

        end if
        Set $x \leftarrow y$;
        Set $t \leftarrow i + 1$;
        if x is a terminal state, i.e., $x \in U \cup E$ **then**
            Terminate the Episode.
        end if
    end for
end for

V. ILLUSTRATING EXAMPLE

Consider the MDP as shown in Figure 2. There are 12

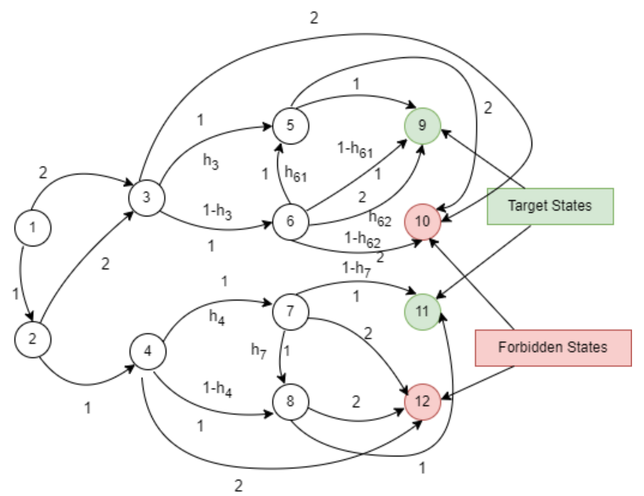

Fig. 2. Example MDP.

states, of which two are forbidden, and two are target states. Specifically, the set of states is $\mathcal{X} = \{1, 2, ..., 12\}$, target set is $E = \{9, 11\}$, forbidden set is $U = \{10, 12\}$, the taboo set is $H = \{1, 2, ..., 8\}$, and the set of actions is $\mathcal{A} = \{1, 2\}$.

While we do not need any model parameters, i.e., the transition probabilities, we assume the proxy set is known. In the above example, the proxy set is $U' = \{3, 4, 5, 6, 7, 8\}$. Suppose we are given to assess the safety function with the target policy π, which is a uniformly random policy for each state, i.e., $\pi(a|x) = 0.5$ for each $x \in H$ and $a \in \mathcal{A}$. Assume that the MDP must be p-safe with $p = 0.1$. We assume that a safe policy is known from each proxy state $x' \in \{3, 4, 5, 6, 7, 8\}$. For each $x' \in U'$, a safe sub-policy is as follows:

$$\pi^S(a|x') = \begin{cases} 0.96, & \text{if } a = 1 \\ 0.04, & \text{if } a = 2, \end{cases}$$

Assume that $h_3 = 0.4$, $h_4 = 0.6$, $h_{61} = 0.4$, $h_{62} = 0.6$ and $h_7 = 0.5$. We show the convergence of the estimated safety function with the target policy and the behavior policy in Figure 3 and 4, respectively. From the figures, it can be seen that the safety function with the safe behavior policy is less than p for all states, hence p-safe. Learning rate $\alpha_k(x)$ is chosen as follows:

$$\alpha_k = \begin{cases} 0.001, & \text{for all episodes } k \leq \mathcal{L}/2 \\ \frac{\alpha_{k-1}}{1+(10^{-6}\cdot\log(k+1)}), & \text{for all episodes } k > \mathcal{L}/2. \end{cases}$$

In Table I, we have shown the true value of the safety function for the target policy π and the behavior policy. These are estimated using the result given in [8]. Further, the estimated safety function for the policies π and π^b, at the end of the last episode $\mathcal{L} = 10^7$, are shown. It is observed that the final estimated safety functions $S_{\mathcal{L}}(x)$ and $S^b_{\mathcal{L}}(x)$ approach arbitrary close to the actual values.

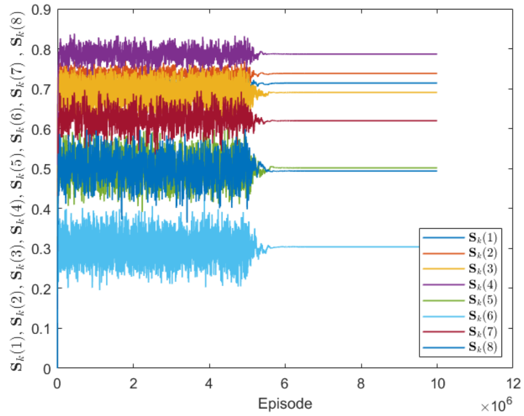

Fig. 3. Convergence of the safety function with the target policy π.

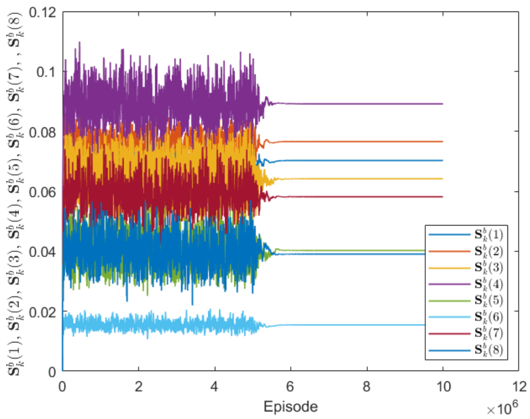

Fig. 4. Convergence of the safety function with the behavior policy π^b.

State (x)	True $S_\pi(x)$ using [8]	$S_{\mathcal{L}}(x)$ using Algorithm 1 $(\mathcal{L} = 10^7)$	True $S_{\pi^b}(x)$ using [8]	$S^b_{\mathcal{L}}(x)$ using Algorithm 1 $(\mathcal{L} = 10^7)$
1	0.7144	0.7140	0.0882	0.0703
2	0.7387	0.7381	0.0888	0.0766
3	0.69	0.6906	0.0734	0.0641
4	0.7875	0.7866	0.0895	0.0891
5	0.5	0.5017	0.04	0.0403
6	0.3	0.3041	0.0314	0.0155
7	0.625	0.6197	0.0592	0.0582
8	0.5	0.4939	0.04	0.0391

TABLE I

COMPARISON OF THE ACTUAL AND ESTIMATED SAFETY FUNCTION.

VI. CONCLUSION AND FUTURE WORK

We have presented a TD(0) method for estimating the safety function without jeopardizing the safety of the MDP. Specifically, we have used an *off-policy* TD (0) with *per-decision importance sampling* to estimate the safety function. We have demonstrated that the estimated safety functions for all the states converge to the true value of the safety function. Further, we have shown that the safety functions with the *behavior policy*, that is followed during the learning, also converges to their true values.

We are working on extending these results to MDP with a large number of states, and continuous dynamical systems. To this end, we will use function approximation-based reinforcement learning techniques. Further, we shall study the convergence of the proposed algorithm.

REFERENCES

[1] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *International Workshop on Hybrid Systems: Computation and Control*. Springer, 2004, pp. 477–492.

[2] A. Chutinan and B. H. Krogh, "Computational techniques for hybrid system verification," *IEEE transactions on automatic control*, vol. 48, no. 1, pp. 64–75, 2003.

[3] C. Sloth, G. J. Pappas, and R. Wisniewski, "Compositional safety analysis using barrier certificates," in *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, 2012, pp. 15–24.

[4] M. L. Bujorianu and J. Lygeros, "Reachability questions in piecewise deterministic markov processes," in *Hybrid Systems: Computation and Control: 6th International Workshop, HSCC 2003 Prague, Czech Republic, April 3–5, 2003 Proceedings 6*. Springer, 2003, pp. 126–140.

[5] M. L. Bujorianu, "Extended stochastic hybrid systems and their reachability problem," in *Hybrid Systems: Computation and Control: 7th International Workshop, HSCC 2004, Philadelphia, PA, USA, March 25-27, 2004. Proceedings 7*. Springer, 2004, pp. 234–249.

[6] S. Prajna, A. Jadbabaie, and G. J. Pappas, "A framework for worst-case and stochastic safety verification using barrier certificates," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1415–1428, 2007.

[7] R. Wisniewski and L.-M. Bujorianu, "Safety of stochastic systems: An analytic and computational approach," *Automatica*, vol. 133, p. 109839, 2021.

[8] R. Wisniewski and M. L. Bujorianu, "Probabilistic safety guarantees for Markov decision processes," *IEEE Transactions on Automatic Control*, 2023.

[9] D. Althoff, M. Althoff, and S. Scherer, "Online safety verification of trajectories for unmanned flight with offline computed robust invariant sets," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3470–3477.

[10] F. Gruber and M. Althoff, "Anytime safety verification of autonomous vehicles," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1708–1714.

[11] A. G. Taye, J. Bertram, C. Fan, and P. Wei, "Reachability based online safety verification for high-density urban air mobility trajectory planning," in *AIAA AVIATION 2022 Forum*, 2022, p. 3542.

[12] A. Lavaei, A. Nejati, P. Jagtap, and M. Zamani, "Formal safety verification of unknown continuous-time systems: a data-driven approach," in *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, 2021, pp. 1–2.

[13] A. Salamati, A. Lavaei, S. Soudjani, and M. Zamani, "Data-driven verification and synthesis of stochastic systems through barrier certificates," *arXiv preprint arXiv:2111.10330*, 2021.

[14] N. Noroozi, A. Salamati, and M. Zamani, "Data-driven safety verification of discrete-time networks: A compositional approach," *IEEE Control Systems Letters*, vol. 6, pp. 2210–2215, 2021.

[15] A. Salamati and M. Zamani, "Data-driven safety verification of stochastic systems via barrier certificates: A wait-and-judge approach," in *Learning for Dynamics and Control Conference*. PMLR, 2022, pp. 441–452.

[16] ——, "Safety verification of stochastic systems: A repetitive scenario approach," *IEEE Control Systems Letters*, vol. 7, pp. 448–453, 2022.

[17] A. Mazumdar, R. Wisniewski, and M. L. Bujorianu, "Online learning of safety function for Markov decision processes," in *European Control Conference (ECC), 2023*. IEEE, 2023, pp. 1–6.

[18] E. Graves and S. Ghiassian, "Importance sampling placement in off-policy temporal-difference methods," *arXiv preprint arXiv:2203.10172*, 2022.

[19] D. Precup, R. S. Sutton, and S. P. Singh, "Eligibility traces for off-policy policy evaluation," in *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000, pp. 759–766.

[20] J. N. Tsitsiklis, "Asynchronous stochastic approximation and Q-learning," *Machine learning*, vol. 16, pp. 185–202, 1994.