Safe Exploration and Optimization of Constrained MDPs using Gaussian Processes

Akifumi Wachi
University of Tokyo
wachi@space.t.u-tokyo.ac.jp

Yanan Sui and **Yisong Yue**
California Institute of Technology
{ysui, yyue}@caltech.edu

Masahiro Ono
Jet Propulsion Laboratory,
California Institute of Technology
ono@jpl.nasa.gov

Abstract

We present a reinforcement learning approach to explore and optimize a safety-constrained Markov Decision Process (MDP). In this setting, the agent must maximize discounted cumulative reward while constraining the probability of entering unsafe states, defined using a safety function being within some tolerance. The safety values of all states are not known a priori, and we probabilistically model them via a Gaussian Process (GP) prior. As such, properly behaving in such an environment requires balancing a three-way trade-off of exploring the safety function, exploring the reward function, and exploiting acquired knowledge to maximize reward. We propose a novel approach to balance this trade-off. Specifically, our approach explores unvisited states *selectively*; that is, it prioritizes the exploration of a state if visiting that state significantly improves the knowledge on the achievable cumulative reward. Our approach relies on a novel information gain criterion based on Gaussian Process representations of the reward and safety functions. We demonstrate the effectiveness of our approach on a range of experiments, including a simulation using the real Martian terrain data.

Introduction

In many environments, an autonomous agent must discover both potential hazards as well as rewards on-the-fly. For example, in case of NASA's rovers exploring the surface of Mars, the driving safety and scientific gain of unvisited places cannot be perfectly known a priori. Hence, at the end of each Martian day, called Sol (approximately 24 hours and 40 minutes), the rover takes a set of panorama images and send it to the Earth. The ground operators identify science targets as well as hazards in the images, and plan a path for the next Sol, which is typically less than 100 meters. This process is exploratory by nature; the rover needs to explore unvisited regions in order to make discoveries, but in a way to assure safety. However, the agent is not fully autonomous in this case because there does not exist a good algorithm to fully drive the decision-making process in situ.

Traditional exploration approaches in reinforcement learning (Sutton and Barto 1998; Bertsekas and Tsitsiklis 1995) are agnostic to safety, and instead focus on efficient exploration under the assumption of ergodicity of the state transitions, i.e., that all states are assumed to be reachable. In practical applications, an agent is not necessarily able to reach all (safe) states because of (blocking) hazardous states. On the other hand, especially in the field of robotics, there is a significant body of work on safety-constrained decision-making (Fleming and McEneaney 1995; Schwarm and Nikolaou 1999; Blackmore et al. 2010; Ono et al. 2015). While these methods consider uncertainty what state one is in, they assume that unsafe states are known *a priori*. Removing this assumption requires exploration of safety, which is a central focus of our work.

To develop a tractable approach for exploring the safe region, we make regularity assumptions that similar states have similar safety levels. In particular, we model the safety function as a Gaussian process (GP) (Rasmussen 2006), with an appropriate kernel to capture similarity between states. Previous work on safe optimization using GP have largely focused on the state-less setting, such as in Bayesian optimization (Mockus 2012). Another typical limitation is when the reward function and the safety function are identical (Sui et al. 2015), e.g., an unsafe state is one where the reward function is too low.

In stateful settings, such as Markov Decision Processes (MDPs), previous work on safe optimization either ignore additional structure (Moldovan and Abbeel 2012), or can only exploit structure for exploring the safe region (Turchetta, Berkenkamp, and Krause 2016) (i.e., pure exploration). The former suffers from longer convergence times since it cannot exploit the GP structure, and the latter suffers from not actually optimizing the reward function.

Our Contributions. In this paper, we propose a novel safe-constrained exploration and optimization approach that maximizes discounted cumulative reward while guaranteeing safety. As demonstrated in Figure 1, we optimize over constrained MDPs with a priori unknown two functions, one for reward and the other for safety. A state is considered safe if the safety function value is above a threshold.

As alluded to above, compared to previous work on jointly exploring and optimizing safety-constrained MDPs (Moldovan and Abbeel 2012), our approach is unique in two aspects. First, (Moldovan and Abbeel 2012) puts no assumption on the structure of reward and safety functions. While that approach is more general, in practice, nearby states often have the similar levels of safety. For example, if a Mars

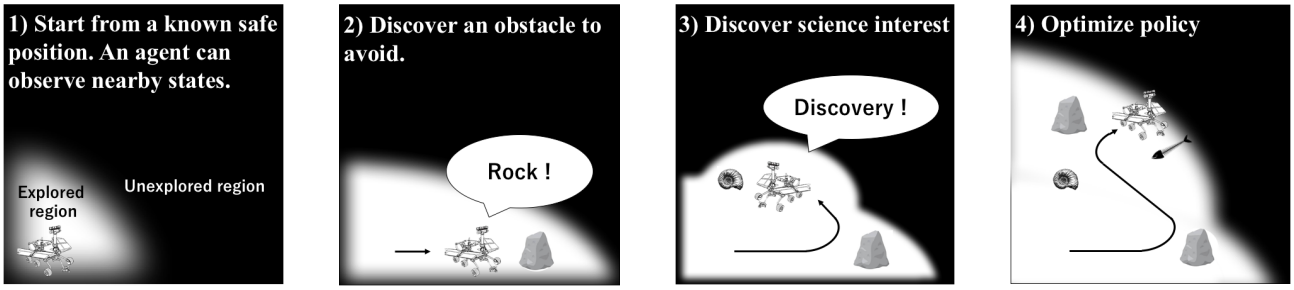


Figure 1: Conceptual images of the planning problem under uncertainty on reward and safety.

rover experiences high slippage at a certain location, it is likely to experience similar slip one meter away. Properly modeling this dependency can significantly reduce the cost of exploration. Second, our approach is able to estimate the long-term value of exploring new safe regions. In contrast, the approach by (Moldovan and Abbeel 2012) assigns uniform additional reward for visiting unvisited states, which can be substantially less efficient.

At a high level, our approach employs two MDPs, which we call *Optimistic* and *Pessimistic* MDPs, and uses the difference in the value functions as the information gain criterion. Our GP safety function yields three classes of states: safe, unsafe, and uncertain. The only difference between the Optimistic and Pessimistic MDPs is that uncertain states are considered safe former and unsafe in the latter. Using this criterion, the agent is motivated to explore uncertain states that could result in high cumulative reward if they are determined safe.

We demonstrate empirically that our approach yields more efficient safe exploration and optimization compared to (Moldovan and Abbeel 2012) and (Turchetta, Berkenkamp, and Krause 2016). The simulations are performed on both synthetic environments, as well as real environments of two locations on Mars.

Problem Statement

We formulate a constrained MDP, defined as a tuple $\langle \mathcal{S}, \mathcal{A}, f(\boldsymbol{s}, a), r(\boldsymbol{s}, a), g(\boldsymbol{s}), \gamma \rangle$. $\mathcal{S}$ is a set of states $\{\boldsymbol{s}\}$, $\mathcal{A}$ is a set of actions $\{a\}$, $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is a deterministic transition model, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a (bounded) reward function, $g : \mathcal{S} \to \mathbb{R}$ is a safety function, and $\gamma \in \mathbb{R}$ is a discount factor. We assume that reward and safety functions are *not* known a priori. Hence, an agent must explore the state space to learn these functions. At every discrete time step t, the agent is required to be in a safe state, that is, the safety function value $g(\boldsymbol{s}_t)$ of state $\boldsymbol{s}_t$ must be above some threshold $h \in \mathbb{R}$. A policy $\pi : \mathcal{S} \to \mathcal{A}$ maps a state to actions. The value of a policy is defined as the discounted cumulative reward over the next N steps while guaranteeing the satisfaction of the safety constraint. The optimal planning problem in this MDP is represented as follows:

$$\text{maximize} : V_N^{\pi}(\boldsymbol{s}) = E\left[\sum_{t=1}^{N} \gamma^{t-1} r(\boldsymbol{s}_t, a_t)\right]$$
$$\text{subject to} : g(\boldsymbol{s}_t) \geq h, \forall t = [1, N].$$

In conventional reinforcement learning, the exploration/exploitation trade-off is a central issue. Compared to the conventional setting, the key difference of our setting is the need to also explore the safety function g. In the rest of the paper, we first describe how we use Gaussian processes to model the safety function and classify the states as safe, unsafe, or uncertain. Then, we describe our algorithm for leveraging this information to efficiently explore the safe region to the extent necessary for optimizing the reward function.

Characterization of Safety

We employ Gaussian processes (GPs) to represent and estimate the safety and reward functions, g and r. For clarity of exposition, we focus our description of GPs with respect to modeling g.[1] In many practical applications, safety exhibits regularity; that is, similar states possess the similar level of safety.[2] Such regularities can be naturally modeled using GPs.

Using the GP model, the value of g at unvisited states are predicted based on the observations at visited states. The GP-based predictions have uncertainty, which is represented by Gaussian distributions. As a result, unvisited states are naturally categorized into three classes: safe, unsafe, and uncertain. Roughly speaking, an unvisited state is considered safe if the safety function value is above the threshold almost certainly (i.e., with a greater probability than a pre-defined confidence level), is considered unsafe if the safety function value is below a threshold almost certainly, and is considered uncertain otherwise. These concepts are shown in Figure 2.

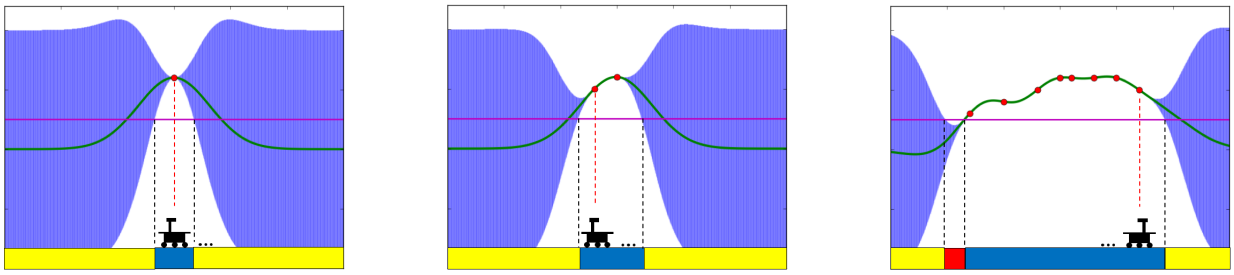
We assume that state space $\mathcal{S}$ is endowed with a positive semidefinite kernel function $k(\cdot, \cdot)$, and that the safety function $g(\boldsymbol{s})$ has bounded norm in the associated Reproducing Kernel Hilbert Space (RKHS). The kernel function k is used to capture similarity between states, and formalizes the assumption that similar states have similar levels of safety. The safety function can thus be modeled as:

$$g(\boldsymbol{s}) = \mathcal{GP}(\mu^g(\boldsymbol{s}), k^g(\boldsymbol{s}, \boldsymbol{s}')).$$

A GP is fully specified by its mean, $\mu^g(\boldsymbol{s})$, and covariance, $k^g(\boldsymbol{s}, \boldsymbol{s}')$. Without loss of generality, let $\mu^g(\boldsymbol{s}) = 0$

[1] Our approach does not prohibit other regression methods from being used for the reward function.

[2] We do not focus on the situations where environmental hazards are expressed as binary function (e.g., rocks and cliffs).



(a) $t = 1$ (b) $t = 2$ (c) $t = 8$

Figure 2: Conceptual image of the characterization of safety. Green curve and blue band represent mean and confidence interval, respectively. The magenta line means the safety threshold, h. Blue, yellow, and red bars represent the safe region, uncertain region, and unsafe region, respectively.

for all $\boldsymbol{s} \in \mathcal{S}$. We model observation noise as $y^g = g(\boldsymbol{s}) + n^g$, where $n^g \sim \mathcal{N}(0, \omega^2)$. The posterior over $g(\boldsymbol{s})$ is computed analytically, based on t measurements at states $\boldsymbol{A}_t = \{\boldsymbol{s}_1, \cdots, \boldsymbol{s}_t\}$ with measurements, $\boldsymbol{y}_t^g = [g(\boldsymbol{s}_1) + n_1^g, \cdots, g(\boldsymbol{s}_t) + n_t^g]^\top$. The posterior mean, variance, and covariance are given as:

$$\begin{aligned} \mu_t^g(\boldsymbol{s}) &= \boldsymbol{k}_t^g(\boldsymbol{s})^\top (\boldsymbol{K}_t^g + \omega^2 \boldsymbol{I})^{-1} \boldsymbol{y}_t^g \\ \sigma_t^g(\boldsymbol{s}) &= k_t^g(\boldsymbol{s}, \boldsymbol{s}) \\ k_t^g(\boldsymbol{s}, \boldsymbol{s}') &= k^g(\boldsymbol{s}, \boldsymbol{s}') - \boldsymbol{k}_t^g(\boldsymbol{s})^\top (\boldsymbol{K}_t^g + \omega^2 \boldsymbol{I})^{-1} \boldsymbol{k}_t^g(\boldsymbol{s}'), \end{aligned} \tag{1}$$

where $\boldsymbol{k}_t^g(\boldsymbol{s}) = [k^g(\boldsymbol{s}_1, \boldsymbol{s}) \cdots, k^g(\boldsymbol{s}_t, \boldsymbol{s})]^\top$ and $\boldsymbol{K}_t^g$ is the positive semidefinite kernel matrix, $[k^g(\boldsymbol{s}, \boldsymbol{s}')]_{\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{A}_t}$. We further assume that the safety function g is Lipschitz continuous, with Lipschitz constant L, with respect to some metric $d(\cdot, \cdot)$ on $\mathcal{S}$. This assumption is naturally satisfied using common kernel functions (Ghosal and Roy 2006).

Safe space We employ the same definition of *safe space* as (Turchetta, Berkenkamp, and Krause 2016). First, let $\tilde{S}_t^{\text{safe}}$ be the set of states that satisfy the safety constraint. The safe space is a subset of $\tilde{S}_t^{\text{safe}}$ that is *reachable* and *returnable*. Intuitively, a state is reachable if it can be reached from a visited safe state. For example, an "island" of unvisited safe states that is isolated by unsafe states is not reachable. A state is returnable if the agent can return to a visited safe state from the state. For example, on a directed graph, a basin of unvisited safe states that "trap" the agent because there are no outgoing edges is not returnable.

First, we formally define $\tilde{S}_t^{\text{safe}}$, the set of states that satisfy the safety constraint. A GP model enables an agent to judge safety by providing it the confidence interval that has a form $Q_t(\boldsymbol{s}) = [\mu_{t-1}(\boldsymbol{s}) \pm \sqrt{\beta_t}\sigma_{t-1}(\boldsymbol{s})]$, where β_t is a positive scalar specifying the required level of safety. In other words, β inherently specifies the probability of violating the safety constraint. Following (Srinivas et al. 2010) and (Sui et al. 2015), for our experiments we use $\beta_t = 2$, $\forall t \geq 0$, and refer to the previous work for a more detailed discussion. We then consider the intersection of Q_t up to iteration t, which is recursively defined as $C_t(\boldsymbol{s}) = Q_t(\boldsymbol{s}) \cap C_{t-1}(\boldsymbol{s})$, $C_0(\boldsymbol{s}) = [h, \infty]$, $\forall \boldsymbol{s} \in \tilde{S}_0^{\text{safe}}$. We denote the lower and upper bounds on $C_t(\boldsymbol{s})$ by $l_t(\boldsymbol{s}) := \min C_t(\boldsymbol{s})$ and $u_t(\boldsymbol{s}) := \max C_t(\boldsymbol{s})$, respectively. With the Lipschitz constant L, $\tilde{S}_t^{\text{safe}}$ is defined as follows:

$$\tilde{S}_t^{\text{safe}} = \{\boldsymbol{s} \in \mathcal{S} \mid \exists \boldsymbol{s}' \in S_{t-1}^{\text{safe}} : l_t(\boldsymbol{s}') - Ld(\boldsymbol{s}, \boldsymbol{s}') \geq h\}.$$

Since the prediction by the GP is probabilistic, being in $\tilde{S}_t^{\text{safe}}$ means the safety constraint is satisfied with high probability. The probability of entering the safe states is guaranteed to be

$$\begin{aligned} &\Pr[g(\boldsymbol{s}) > h \mid l(\boldsymbol{s}') - Ld(\boldsymbol{s}, \boldsymbol{s}') > h] \\ \geq\ &\Pr[g(\boldsymbol{s}) > h \mid l(\boldsymbol{s}) > h] \\ \geq\ &\Pr[g(\boldsymbol{s}) > l(\boldsymbol{s}) \mid l(\boldsymbol{s}) > h] \\ =\ &\Pr[g(\boldsymbol{s}) > l(\boldsymbol{s})] \end{aligned}$$

The last equality exploits the conditional independence. Note that this probability can be tuned by β.

Next, we formally define the reachable and returnable sets. Given $\mathcal{S}$, the set that is reachable from $\mathcal{S}$ in one step is $R^{\text{reach}}(\mathcal{S}) = \mathcal{S} \cup \{\boldsymbol{s} \in \mathcal{S} \mid \exists \boldsymbol{s}' \in \mathcal{S}, a \in \mathcal{A} : \boldsymbol{s} = f(\boldsymbol{s}', a)\}$. Given sets $\mathcal{S}$ and $\bar{\mathcal{S}}$, the set that is returnable to $\bar{\mathcal{S}}$ with one step is given as $R^{\text{ret}}(\mathcal{S}, \bar{\mathcal{S}}) = \bar{\mathcal{S}} \cup \{\boldsymbol{s} \in \mathcal{S} \mid \exists a \in \mathcal{A} : f(\boldsymbol{s}, a) \in \bar{\mathcal{S}}\}$. The n-step returnability operator can be represented as $R_n^{\text{ret}}(\mathcal{S}, \bar{\mathcal{S}}) = R^{\text{ret}}(\mathcal{S}, R^{\text{ret}}(\mathcal{S}, \cdots))$. Using this operator, the returnable set is given by $\bar{R}^{\text{ret}}(\mathcal{S}, \bar{\mathcal{S}}) = \lim_{n \to \infty} R_n^{\text{ret}}(\mathcal{S}, \bar{\mathcal{S}})$. Finally, the set of safe states that fulfill reachability and returnability, S_t^{safe}, is defined as:

$$S_t^{\text{safe}} = \{\boldsymbol{s} \in \tilde{S}_t^{\text{safe}} \mid \boldsymbol{s} \in R^{\text{reach}}(S_{t-1}^{\text{safe}}) \cap \bar{R}^{\text{ret}}(\tilde{S}_t^{\text{safe}}, S_{t-1}^{\text{safe}})\}.$$

Unsafe space We extend the setting of (Turchetta, Berkenkamp, and Krause 2016) to also define the *unsafe space*, S_t^{unsafe}. As in the safe set, we first define the set of states that violate the safety constraint with a high confidence as:

$$\tilde{S}_t^{\text{unsafe}} = \{\boldsymbol{s} \in \mathcal{S} \mid \exists \boldsymbol{s}' \in \mathcal{S} : u_t(\boldsymbol{s}') + Ld(\boldsymbol{s}, \boldsymbol{s}') \leq h\}.$$

Using the returnable set defined above, the unsafe space, denoted by S_t^{unsafe}, is defined as:

$$S_t^{\text{unsafe}} = \{\boldsymbol{s} \in \mathcal{S} \mid \boldsymbol{s} \in \tilde{S}_t^{\text{unsafe}} \cup (\tilde{S}_t^{\text{safe}} \cap \bar{R}_c^{\text{ret}}(\tilde{S}_t^{\text{safe}}, S_{t-1}^{\text{safe}})\},$$

where X_c represents the complementary set of a set X. Note that we do *not* take into account reachability. In other words, an "island" of unvisited states should be regarded as unsafe.

Uncertain Space Finally, the uncertain space is simply the set of remaining states that belong to neither the safe nor unsafe spaces:

$$S_t^{\text{uncertain}} = S \setminus \left(S_t^{\text{safe}} \cup S_t^{\text{unsafe}}\right).$$

Intuitively, uncertain states can be either safe or unsafe, and an agent must explore the uncertain space to find new safe states with high reward. However, uncertain states cannot be guaranteed to be safe, so the agent must leverage the structure of the safety function (i.e., the similarity kernel) and explore the boundary of the safe space to collect measurements that are informative of uncertain states.

Method

Our proposed algorithm is named SAFEEXPOPT-MDP , which is summarized in Algorithm 1. At each iteration, the agent observes the reward and safety function values of the current state, and update the GP models. Using the updated GPs, the agent then updates the safe, unsafe, and uncertain sets (Line 3). Given these sets, the agent solves Optimistic and Pessimistic MDPs (explained in the following subsections) using a PAC-MDP or Near-Bayesian Optimal algorithm, and obtain their value functions (Line 4). Then, the agent chooses the next action such that the linear combination of the two value functions is maximized (Line 5). Note that the linearly combined value functions in Line 5 can also be represented as $\bar{J}_N + \eta(\hat{J}_N - \bar{J}_N)$.

Therefore, the resulting policy favors an action to visit states that have a large difference in value between the Optimistic and Pessimistic MDPs. We interpret this difference corresponds to an information gain criterion, because it becomes zero when the level of uncertainty on safety is similar along with the paths obtained in two MDPs. Hence, this mechanism motivates the agent to selectively explore the states which could result in significant increase in the cumulative reward.

Value of Exploration of Safety

Ultimately, the agent's objective is to maximize the amount of reward collected. Thus, the value of exploration should be directly tied to the potential reward that could be collected. Then, how should we evaluate the value of exploration of a given unvisited state? As alluded to above, we quantify the value of exploration as the difference between the optimistic case (where uncertain states are all safe) and the pessimistic case (where uncertain states are all unsafe), as described in Line 5 of Algorithm 1. The Bayes-optimal, optimistic, and pessimistic value functions are obtained by solving the following Bellman equation:

$$V_N^*(s, b^r, b^g) = \max_a \Big[\sum_{s'} I(s' \mid s, a, b^g) \cdot P_{s,a,s'} \cdot \{E[r(s')] + \gamma V_{N-1}^*(s', b^r_{s,a,s'}, b^g)\} \Big],$$

where I is a safety indicator function (explained shortly), P is the deterministic transition probability, b^r and b^g are the belief of reward and safety functions, respectively, at time step N, and $b^r_{s,a,s'}$ is the posterior belief of reward after taking the action a and moving to s'.

Algorithm 1 SAFEEXPOPT-MDP

1: **loop**
2: Observe reward and safety values of the current state, and update GP models
3: Update $\mathcal{S}_t^{\text{safe}}$, $\mathcal{S}_t^{\text{uncertain}}$, and $\mathcal{S}_t^{\text{unsafe}}$
4: Compute $\hat{J}_N$ for Optimistic MDP and $\bar{J}_N$ for Pessimistic MDP by Eq. (5) and Eq. (6)
5: Derive the optimal policy to maximize $\eta\hat{J}_N + (1-\eta)\bar{J}_N$ by Eq. (8)
6: Execute the next optimal action
7: **end loop**

Optimistic MDP assumes that all the uncertain states are safe. Hence, the safety indicator function for Optimistic MDP is defined as:

$$\hat{I}(s' \mid s, a, b^g) := \begin{cases} 1 & \text{if } \boldsymbol{s'} \in \mathcal{S}^{\text{safe}} \cup \mathcal{S}^{\text{uncertain}} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

We denote the value function of the Optimistic MDP by $\hat{V}_N$.

Pessimistic MDP assumes that all the uncertain state are unsafe. Hence, the safety indicator function for Pessimistic MDP is defined as:

$$\bar{I}(s' \mid s, a, b^g) := \begin{cases} 1 & \text{if } \boldsymbol{s'} \in \mathcal{S}^{\text{safe}} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

We denote the value function of the Pessimistic MDP by $\bar{V}_N$.

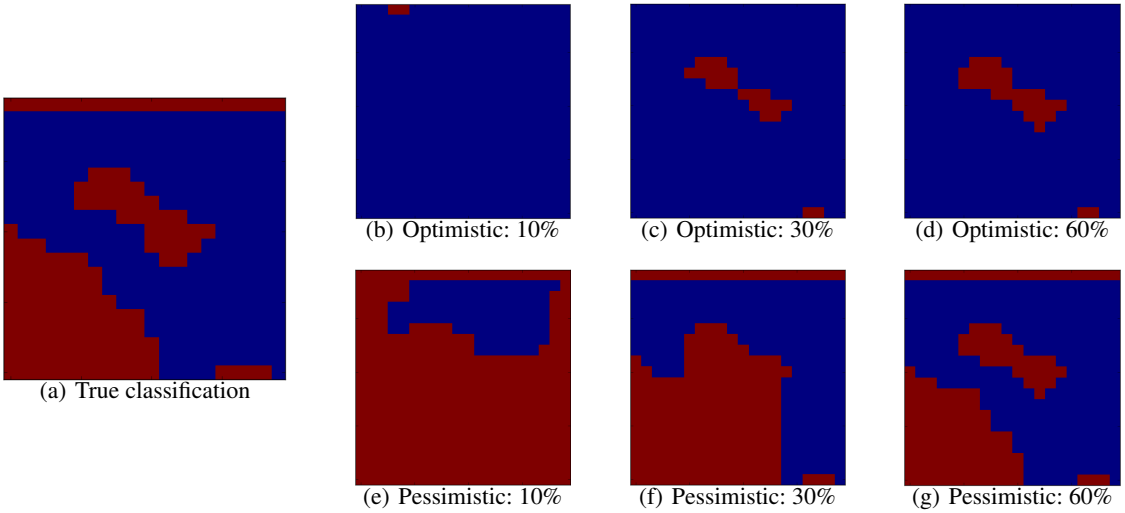
The two value functions always satisfy the relationship, $\bar{V}_N^* \leq V_N^* \leq \hat{V}_N^*$,where V_N^* is the value function of the Bayes optimal policy for the original problem. Figure 3 shows an example of the classification of the state space, and Table 1 concludes the definition of value functions.

Incentivizing Exploration of Reward Function

Practically, the above Bellman equations for the Optimistic and Pessimistic MDPs are very difficult to solve. Instead, we solve them approximately. Previous work proposed algorithms that have a property called *Probably Approximate Correct in Markov Decision Process (PAC-MDP)* (Kearns and Singh 2002; Brafman and Tennenholtz 2002; Kakade and others 2003; Strehl et al. 2006) or *Near-Bayes Optimal* (Strens 2000; Kolter and Ng 2009; Araya, Buffet, and Thomas 2012). The general idea of PAC-MDP and Near-Bayesian Optimal is that, with probability of $1-\delta$, an agent following the policy π can obtain the reward such that

$$V^\pi > V^* - \epsilon$$

for all but m time steps. For example, the Bayesian Exploration Bonus (BEB) algorithm (Kolter and Ng 2009) gives m that is $\mathcal{O}(\frac{|S||A|N^6}{\varepsilon^2} \log \frac{|S||A|}{\delta})$. In PAC-MDP and Near-Bayes Optimal approaches, the approximately optimal policy π is obtained by Delayed Q-learning or the exploration bonus algorithm, which adds "bonus" to the original reward function for exploring unexplored states. As a result, the value function including the bonus, denoted by J^π, is always greater



(a) True classification (b) Optimistic: 10% (c) Optimistic: 30% (d) Optimistic: 60%

(e) Pessimistic: 10% (f) Pessimistic: 30% (g) Pessimistic: 60%

Figure 3: An example of the classification of state space. Blue region represents $S^{\text{safe}} \cup S^{\text{uncertain}}$ for optimistic case and S^{safe} for pessimistic. The percentage for each figure means the time course against the time step large enough to explore $\mathcal{S}$.

Table 1: Definition of value (objective) functions. As for reward, we can consider three cases: actually obtainable, Bayes optimal, and value by algorithm. In the case of safety, we consider pessimistic, Bayes optimal, and optimistic cases.

	$\bar{V}_N^\pi$	$\bar{V}_N^*$	$\bar{J}_N^\pi$	V_N^*	$\hat{V}_N^\pi$	$\hat{V}_N^*$	$\hat{J}_N^\pi$
Reward	Actually obtainable	Bayes optimal	Value by algorithm	Bayes optimal	actually obtainable	Bayes optimal	Value by algorithm
Safety	Pessimistic	Pessimistic	Pessimistic	Bayes optimal	Optimistic	Optimistic	Optimistic

than the actual value function and satisfies the following with the probability of $1-\delta$:

$$V^\pi \leq V^* \leq J^\pi \quad \text{and} \quad V^\pi \geq J^\pi - \epsilon \geq V^* - \epsilon, \tag{4}$$

for all but m time steps. With an exploration bonus algorithm, such as BEB, the value functions with bonus for the Optimistic and Pessimistic MDPs, $\hat{J}^\pi$ and $\bar{J}^\pi$, are given as follows:

$$\hat{J}^\pi(\boldsymbol{s}, b^r, b^g) = \max_a \Big[\sum_{s'} \hat{I} \cdot P_{s,a,s'} \cdot \{E[r] + \tilde{R} + \gamma \hat{J}^\pi(\boldsymbol{s}, b^r, b^g)\} \Big], \tag{5}$$

$$\bar{J}^\pi(\boldsymbol{s}, b^r, b^g) = \max_a \Big[\sum_{s'} \bar{I} \cdot P_{s,a,s'} \cdot \{E[r] + \tilde{R} + \gamma \bar{J}^\pi(\boldsymbol{s}, b^r, b^g)\} \Big], \tag{6}$$

where $\tilde{R}$ is the term for the exploration bonus. Note that the belief of reward b^r and safety b^g are not updated in these equations, which means that we can solve the equation using the standard value iteration or policy iteration algorithm. Our proposed method optimizes two MDPs, so has a similar tractability compared with conventional MDP algorithms.

Overall Policy

As mentioned above, our algorithm evaluates the value of exploration by the difference of the approximate value functions of the Optimistic and Pessimistic MDPs. Let ΔJ be the difference, and define

$$\Delta J_N := \hat{J}_N - \bar{J}_N.$$

Intuitively, the agent obtains a high information gain on safety when visiting a state with a large ΔJ. After computing $\hat{J}_N$ and $\bar{J}_N$, the agent chooses the next action by following the policy given as:

$$\pi_N = \arg\max_{\boldsymbol{s}' \in \mathcal{S}^{\text{safe}}} \{\bar{J}_N + \eta \Delta J_N\} \tag{7}$$

$$= \arg\max_{\boldsymbol{s}' \in \mathcal{S}^{\text{safe}}} \{\eta \hat{J}_N + (1-\eta)\bar{J}_N\}, \tag{8}$$

where $\eta \in [0,1]$ is a weight coefficient. Note that our approach optimizes the interpolating point between the optimistic and pessimistic policies. One can view η as controlling for the degree of optimism versus pessimism.[3] Also, note that the next state is chosen from $\mathcal{S}^{\text{safe}}$ in Equation (8).

Proof of Consistency

We prove a consistency result that the objective function in our approach converges as time horizon goes to infinity.

[3] η often has intuitive, physical meaning, which helps in tuning.

Table 2: Simulation result for a simple grid world (100 times Monte-Carlo simulation). Our approach achieves higher expected reward with less exploration, which suggests that our approach is able to better trade off between exploration and exploitation and avoid unnecessary exploration.

	Normalized cumulative reward	Percentage explored
SAFEEXPOPT-MDP	0.774 ± 0.183	7.4 ± 2.9
Moldovan & Abbeel (2012)	0.740 ± 0.213	8.6 ± 3.1
Turchetta et al. (2016)	0.701 ± 0.248	10.6 ± 2.1
Safe known	0.841 ± 0.089	-
Safe/reward known	1.00 ± 0.000	-

More specifically, the following theorem holds:

Theorem 1. *For given δ, there exists the upper bound of error, $\varepsilon = \eta\hat{\varepsilon} + (1-\eta)\bar{\varepsilon}$ for the value function (8) with a probability of $1-\delta$ for all but the first m time steps such that:*

$$0 \leq (\eta\hat{J}_N + (1-\eta)\bar{J}_N) - (\eta\hat{V}_N^* + (1-\eta)\bar{V}_N^*) \leq \varepsilon.$$

Proof. After the sufficiently large number of time steps, (4) holds with high probability for both the optimistic and pessimistic policies, and the following inequalities are satisfied:

$$\begin{aligned}\eta\hat{V}_N^* + (1-\eta)\bar{V}_N^* &\leq \eta\hat{J}_N + (1-\eta)\bar{J}_N \\ &\leq \eta(\hat{V}_N^{\pi} + \hat{\varepsilon}) + (1-\eta)(\bar{V}_N^{\pi} + \bar{\varepsilon}) \\ &\leq \eta(\hat{V}_N^* + \hat{\varepsilon}) + (1-\eta)(\bar{V}_N^* + \bar{\varepsilon}),\end{aligned}$$

where $\hat{\varepsilon}$ and $\bar{\varepsilon}$ are confidence intervals in the optimistic and pessimistic policies, respectively. The inequalities are satisfied with the high probability of $1-\delta = (1-\hat{\delta})(1-\bar{\delta})$, where $\hat{\delta}$ and $\bar{\delta}$ are the probability that Equation (4) does not hold against each policy. Hence, a novel value function in our paper approaches to $\eta\hat{V}_N^* + (1-\eta)\bar{V}_N^*$ as closely as $\varepsilon = \eta\hat{\varepsilon} + (1-\eta)\bar{\varepsilon}$ with high probability of $1-\delta$ for all but a certain time steps. □

Experiments

We evaluate our approach on two problem settings. One is with randomly generated environments on which we perform Monte-Carlo simulations, and the other is with environments created from real Martian terrain data. For each simulation, GP hyper-parameters are obtained through trial and error. As for the two Mars simulations, we refer to the previous work by (Turchetta, Berkenkamp, and Krause 2016) and use similar parameters.

Monte-Carlo Simulations with Randomly Generated Problems

We consider a 30×30 rectangular grid, where reward and safety values are randomly generated and assigned to states. At every state (except for the boundary), the agent can take one of five actions: *stay*, *up*, *down*, *left*, and *right*.

The exploring agent predicts the safety function g via GP with a Radial Basis Function (RBF) kernel with the lengthscales being 2.0 and the prior variance of safety being 1.5. The agent also models the reward function r as a GP with RBF kernel with the lengthscales of 2.0 and prior variance of 1.0. Throughout the simulation, the discount rate is set to 0.90 and beta is set as $\beta_t = 2,\ \forall t \geq 0$.

The Optimistic and Pessimistic MDPs are solved by the BEB algorithm with the weight coefficient of 2.5 for the exploration bonus. In the MDPs, the estimated mean of reward function is used. The safety constraint is imposed by removing the unsafe states from $\mathcal{S}$ in the Optimistic MDP, and by removing unsafe and uncertain states from $\mathcal{S}$ in the Pessimistic MDP. The weight coefficient between optimistic and pessimistic policies is set to $\eta = 0.50$.

Our approach is compared with the ones by (Moldovan and Abbeel 2012) and (Turchetta, Berkenkamp, and Krause 2016), as well as with two non-exploratory cases where 1) safety function is known and 2) safety and reward functions are known *a priori*. In order to make a fair comparison, the MDPs are solved by the BEB algorithm in all the targets of comparison ((Moldovan and Abbeel 2012) used R-MAX in their paper). We performed a Monte-Carlo simulation with 100 samples of randomly generated reward and safety. The number of iteration is fixed and defined as 100.

Table 2 shows the simulation results. We see that our approach outperforms the competing baselines, although there is a sizable gap compared to more omniscient agents. Note also that our approach explored the fewest uncertain states compared to the baselines, and instead spent more effort finding a good reward function within a high quality safe region. These results show that our approach is able to better balance the three-way trade-off between exploring the safe region, exploring the reward function, and exploiting known rewards.

Simulated Mars Surface Exploration

We next demonstrate SAFEEXPOPT-MDP algorithm in simulated Mars surface exploration scenarios, as in (Moldovan and Abbeel 2012) and (Turchetta, Berkenkamp, and Krause 2016). Two scenarios are simulated. The first scenario uses the similar environment as in (Turchetta, Berkenkamp, and Krause 2016), where the elevation map of Mars over a 40 by 30 meters area at latitude 30°6' South and longitude 202°2' East is used. The elevation map is derived from the digital terrain models (DEMs), created from HiRISE camera on the Mars Reconnaissance Orbiter (McEwen et al. 2007). Figure 4 shows the elevation map whose step size is 1 meter.

Table 3: Simulation result for Martian terrain. As with Table 2, our approach achieves higher reward with less exploration.

	First Mars exploration scenario		Second Mars exploration scenario	
	Cumulative reward	Percentage explored	Cumulative reward	Percentage explored
SAFEEXPOPT-MDP	36	5.3 %	67	11.8 %
Moldovan & Abbeel (2012)	29	5.8 %	45	12.9 %
Turchetta et al. (2016)	5	6.7 %	0	15.3 %
Safe known	48	-	78	-
Safe/reward known	63	-	458	-

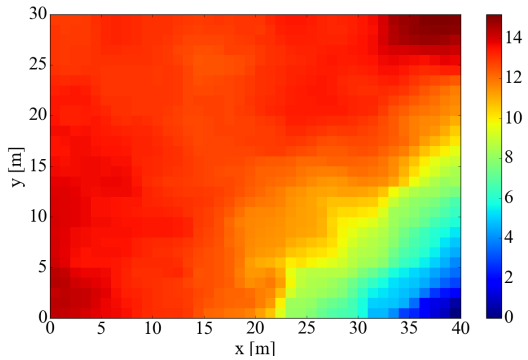


Figure 4: Mars digital elevation model by HiRISE data [m].

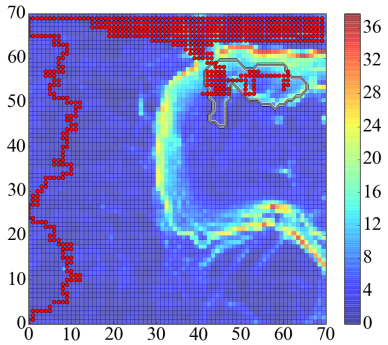


Figure 5: Data on the slope angle [deg] and ROI employed for the landing site selection of Mars 2020 rover.

The reward function is binary: one if a state is within a scientific region of interest (ROI), and zero otherwise. We assume a hypothetical rectangular ROI at $16 \leq x \leq 24$ and $21 \leq y \leq 30$. The rover (i.e., the agent) starts from $(0, 0)$, and at each time step it takes one of five actions: *stay*, *up*, *down*, *left*, and *right*. We also assume that any states where the slope is greater than 25 degrees are unsafe. Hence, the safety function g is defined as $g(\boldsymbol{s}, a) = H(\boldsymbol{s}) - H(\boldsymbol{s}')$, where H is the elevation of a given state. The safety threshold is $h = -\tan(25°)$.

The rover predicts the elevation using a GP with a Matern kernel with $\nu = 5/2$. The lengthscales are 15.0 m and the prior variance over elevation is 100 m^2. We assume a noise standard deviation of 0.075 m. To predict the reward function, it uses a GP with Radial Basis Function (RBF) kernel. The lengthscales are 10.0 m and the prior variance over reward is 50. We set the level of confidence as $\beta_t = 2$, $\forall t \geq 0$. Optimistic/Pessimistic MDPs are solved by BEB algorithm with the weight coefficient for exploration bonus at 2.5. The discount factor is set to 0.90, and the weight coefficient between optimistic and pessimistic policies is $\eta = 0.50$.

Our algorithm is compared against the same algorithms as in the previous simulation. The simulation result is summarized in the left half of Table 3. The result shows that our algorithm outperforms existing algorithms in terms of the cumulative reward even for the binary reward function. We observe a 24% relative improvement in reward compared to the next best baseline. Note that since the reward function is more peaked, the pure exploration approach by (Turchetta, Berkenkamp, and Krause 2016) performs substantially worse. We again observe that our approach conducts the least amount of exploration of the safety region, which indicates it is better able to balance the trade-off between exploring the safety region and the reward function.

The second scenario is based on the potential exploration scenario of *Mars 2020 rover*. We employ the real data of a region over 1.75 by 1.75 kilometers around Holden Crater, one of the previous candidates of the landing site of *Mars 2020 rover*. The safety function g is the slope angle, and we assume that every state where the slope is greater that 25 degrees is unsafe.

The reward function r is defined as a binary function meaning scientific gain: one if a state is within a ROI, and zero otherwise. The rover predicts the elevation using a GP with a Matern kernel with $\nu = 5/2$. The lengthscales are 50.0 m and the prior variance over elevation is 100 deg^2. We assume a noise standard deviation of 0.075 deg. To predict the reward function, the agent (rover) uses a GP with RBF kernel. The lengthscales are 50.0 m and the prior variance over reward is 50. Other simulation settings are equivalent with previous Mars simulation in this paper.

Figure 5 represents simulation environment in which the color map behind is the slope angle [deg] and the region surrounded by the black close curve represents ROI. The step size is 25 meters, and we consider 70×70 rectangular grid. The rover starts from $(0, 0)$, and takes a route represented by red circles. At the initial phase before arriving at the ROI, all the area around an agent is safe and reward-poor, so an agent considers that all the environment has similar characteristics. Hence an agent explores the state space all evenly by randomly choosing the next action. When the value functions are exactly equal between more than one state, an agent

randomly chooses the next state and action pair. However, once the agent discovers the ROI, it is likely to prefer to stay within the ROI. In addition, observe that the rover does not run through the states in which the slope angle is greater than 25 degrees, which shows that the rover succeeded in safely arriving at ROI. The simulation result is summarized in the right half of Table 3. There is a large gap between not knowing the safety/reward functions and knowing them (i.e., unrealistic case), which suggests there is still room for improving the algorithm.

Conclusion

We presented a novel approach for exploring and optimizing safety constrained MDPs. By modeling a priori unknown reward and safety via GPs, an agent can classify state space into safe, uncertain, and unsafe regions. The agent then optimizes the linear combination of optimistic policy and pessimistic policy. This algorithm automatically encourages exploration of safety while balancing exploration/exploitation of reward. In addition, the agent can guarantee safety with high probability by exploiting GP structure of the safety function. Finally, we demonstrated the effectiveness of our proposed method by theoretical analysis and numerical simulation. Moving forward, it would be interesting to develop rigorous finite-time rates of convergence (rather than just an asymptotic consistency result), as well as extend to multiple heterogeneous safety and reward functions.

Acknowledgement

The research was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology. This research was also supported in part by NSF Awards #1564330 & #1637598, JPL PDF IAMS100224, a Bloomberg Data Science Research Grant, and a gift from Northrop Grumman.

References

Araya, M.; Buffet, O.; and Thomas, V. 2012. Near-optimal BRL using optimistic local transitions. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 97–104.

Bertsekas, D. P., and Tsitsiklis, J. N. 1995. Neuro-dynamic programming: an overview. In *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, volume 1, 560–564. IEEE.

Blackmore, L.; Ono, M.; Bektassov, A.; and Williams, B. C. 2010. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE transactions on Robotics* 26(3):502–517.

Brafman, R. I., and Tennenholtz, M. 2002. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research* 3(Oct):213–231.

Fleming, W. H., and McEneaney, W. M. 1995. Risk-sensitive control on an infinite time horizon. *SIAM Journal on Control and Optimization* 33(6):1881–1915.

Ghosal, S., and Roy, A. 2006. Posterior consistency of Gaussian process prior for nonparametric binary regression. *The Annals of Statistics* 2413–2429.

Kakade, S. M., et al. 2003. *On the sample complexity of reinforcement learning*. Ph.D. Dissertation, University of London.

Kearns, M., and Singh, S. 2002. Near-optimal reinforcement learning in polynomial time. *Machine Learning* 49(2-3):209–232.

Kolter, J. Z., and Ng, A. Y. 2009. Near-Bayesian exploration in polynomial time. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 513–520. ACM.

McEwen, A. S.; Eliason, E. M.; Bergstrom, J. W.; Bridges, N. T.; Hansen, C. J.; Delamere, W. A.; Grant, J. A.; Gulick, V. C.; Herkenhoff, K. E.; Keszthelyi, L.; et al. 2007. Mars reconnaissance orbiter's high resolution imaging science experiment (HiRISE). *Journal of Geophysical Research: Planets* 112(E5).

Mockus, J. 2012. *Bayesian approach to global optimization: theory and applications*, volume 37. Springer Science & Business Media.

Moldovan, T., and Abbeel, P. 2012. Safe exploration in Markov decision processes. In *International Conference on Machine Learning (ICML)*.

Ono, M.; Pavone, M.; Kuwata, Y.; and Balaram, J. 2015. Chance-constrained dynamic programming with application to risk-aware robotic space exploration. *Auton. Robots* 39(4):555–571.

Rasmussen, C. E. 2006. Gaussian processes for machine learning.

Schwarm, A. T., and Nikolaou, M. 1999. Chance-constrained model predictive control. *AIChE Journal* 45(8):1743–1752.

Srinivas, N.; Krause, A.; Kakade, S. M.; and Seeger, M. 2010. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*.

Strehl, A. L.; Li, L.; Wiewiora, E.; Langford, J.; and Littman, M. L. 2006. PAC model-free reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 881–888. ACM.

Strens, M. 2000. A bayesian framework for reinforcement learning. In *International Conference on Machine Learning (ICML)*, 943–950.

Sui, Y.; Gotovos, A.; Burdick, J. W.; and Krause, A. 2015. Safe exploration for optimization with gaussian processes. In *International Conference on Machine Learning (ICML)*.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Turchetta, M.; Berkenkamp, F.; and Krause, A. 2016. Safe exploration in finite Markov decision processes with gaussian processes. In *Advances In Neural Information Processing Systems*, 4305–4313.