Safe Exploration in Markov Decision Processes

Teodor Mihai Moldovan
Department of Computer Science
University of California at Berkeley
Berkeley CA 94720, USA
moldovan@cs.berkeley.edu

Pieter Abbeel
Department of Computer Science
University of California at Berkeley
Berkeley CA 94720, USA
pabbeel@cs.berkeley.edu

Abstract

We explain the need for safe exploration methods in Markov Decision Processes and present three different formulations of safety that might be relevant in specific practical applications. Our framework allows restricting any previously proposed exploration method to consider only safe policies, as long as it relies on some exploration bonus. All the formulations we consider can be expressed as instances of *policy coupling* problems which we investigate. We show that policy coupling is, unfortunately, NP-hard, so we propose two approximate solutions based on a first order approximation of the problem. The first solution is an iterative algorithm similar to gradient descent that is guaranteed to find local optima eventually. We show how to leverage recent advances in parallel computating in this case. The second solution relies on further approximation, but is computationally efficient. Finally, our grid world exploration experiments show that the second method works well in practice and illustrate key differences between safe and regular exploration.

1 Motivation

Our article addresses the issues of safety and exploration in planning problems when the parameters of the system, or system dynamics, are known only approximatively. We want to find policies that will perform acceptably well under all likely dynamics, while, at the same time, collecting information about the system that is relevant to improving performance in the future. For example, consider a robot helicopter practicing an aerobatic maneuver it has just learned by watching a expert perform it [1]. Its dynamics model is inaccurate because it has observed only a few executions so far, and the demonstrations might have been suboptimal. We would like it to experiment with variations around the current policy to learn more about its dynamics, and, thus, improve performance, but, at the same time, we want to avoid crashing because repairs are expensive. Since it's impossible to completely avoid all accidents, we are willing to accept some low probability of crashing. This is an interesting, practical example of a *safe exploration* problem.

Without safety guarantees, current exploration methods remain theoretical exercises, since, in practice, nobody would trust them to control critical systems. For example, a personal robot doing housework would have to explore its environment to be efficient, but it should not be damage objects or hurt people in the process. Similarly, a self-driving car would have to continuously learn about road conditions, but should not compromise passenger safety at any time. For example, it would need to check break effectiveness when it encounters rain or show, but should only do so when safe.

The first issue that makes safe exploration difficult is lack of ergodicity. We call a Markov Decision Process *ergodic* if any state is reachable from any other state by following a suitable policy. This assumption rarely holds in practice, yet the guarantees of most exploration algorithms depend on it. The second difficulty concerns correlated parameter ambiguity which is a necessary feature in safe exploration since it allows a system to make informed guesses about unseen states based on properties of the previously seen neighboring states. However, dealing with correlations is more difficult than assuming independent ambiguity, which is the usual practice. Finally, safe exploration is necessarily a multi-objective optimization problem where the objective being optimized relates to exploration and the objective being constrained encodes safety. By contrast, all previous safety or exploration methods solve single objective optimization problems.

The issue of safety, or risk aversion in MDPs has been addressed before [2, 3, 4, 5, 6, 7, 8], but few of the proposed methods provide strong probabilistic guarantees [6], and they tend to assume uncorrelated ambiguity in the system dynamics. Exploration in MDPs has also been addressed before [2, 9, 10, 11, 12, 13], but most methods tend to rely on the assumption of ergodicity and, consequently, explore too aggressively. Few authors have attempted to provide safety guarantees for exploration methods [14], but their approaches are difficult to generalize since they were engineered specifically to particular problems.

2 Notation and assumptions

For brevity, we will not give a general introduction to MDPs here. Instead, we direct readers who are not familiar with the topic to [15]. We follow the Bayesian approach when accounting for model ambiguity and we the transition probabilities and rewards be random variables. In our notation, capital letters denote random variables. For example, the usual value function recursion, conditional on dynamics, takes the following form:

$$V := \sum_{t=0}^{\infty} R_{S_t, A_t}, \quad \text{and the Markov property implies:}$$

$$E_{\delta(s),\pi}\left[V \mid P, R\right] = \sum_{a,s'} \pi_{s,a} R_{s,a} + P_{s,a,s'} E_{\delta(s'),\pi}[V \mid P, R]$$

Technically, both P and R are random tensors indexed by the state and action processes, S and A. We subscript the expectation functional, E, to make the measure explicit. For simplicity of notation, and without loss of generality, we choose not to represent discounting explicitly. Instead, we implicitly assume that there exists at least one absorbing state and that all states in absorbing sets generate no rewards.

3 Different ways to formulate safe exploration

We propose three different definitions of what safe exploration might mean, depending on application. The safety guarantees are understood to hold with hight probability, as specified by the user, during exploration. Of course, safe exploration might not be possible within the specified safety parameters. In this case, we ignore the exploration objective and simply maximize the safety objective.

The first, and most general safety formulation amounts to forcing ergodicity by ensuring that the system remain in the same connected component of the state space during exploration. We necessarily assume that the objective is withing the connected component of the starting state, so there is a way to achieve the objective and then return to the start state. This natural assumption will hold in most interesting real world problems. The exploration efficiency guarantees will now hold for the current component, so, eventually, all of it will be explored. For example, a crawler robot exploring an earthquake site would have to avoid losing ergodicity by getting stuck in rubble.

The second variant is guaranteeing that the expected reward exceed some user specified minimum while exploring. This is suitable in situations where we already know a reasonably good policy, and we want to improve it by exploring, but the system must continuously remain in production and the returns may not diminish too much. A good example is a chemical plant trying to reduce costs by optimizing process parameters. Safety amounts to ensuring that revenue remains hight enough to cover expenses during exploration.

The third option is ensuring that the system remain within some user specified set of safe states. This is appropriate whenever the state space is simple enough that an expert could explicitly formulate a set of safe states. An robot helicopter is such a system because experienced pilots can specify ranges for pitch, roll, yaw, positions, velocities and angular velocities such that the helicopter would safely land if control were shut down.

Even though the three proposed safety criteria seem different, they all share the same type of structure; they can all be formalized as *policy coupling* problems:

$$\max_{\pi} \; E_{\rho,\pi,\mu} \sum_t R_{S_t,A_t} \quad \text{such that:} \quad E_{\rho,\pi,\sigma} \sum_t R_{S_t,A_t} \geq b \tag{1}$$

The same state and action processes, S and A, appear in both the objective and the constraint. Their distribution is determined by the measures ρ, which describes the start state distribution, and π, which encodes the policy. On the left hand side, the measure μ assigns mass to rewards and transitions policies, R and P, such that they encode the exploration objective. On the right hand side, the measure σ "picks out" rewards and transition probabilities to encode the safety objective. The following identity serves to further clarify clarify notation and show how to compute these expectations in practice.

$$E_{\rho,\pi,\mu} \sum_t R_{S_t,A_t} = E_\mu \left[E_{\rho,\pi} \left[\sum_t R_{S_t,A_t} \,|\, P, R \right] \right]$$

We can view the proposed safety constraints as restrictions on the space of allowable policies. The exploration objective would, then, be optimized on this restricted space of safe policies. However, the way we define safety does not influence the type of exploration objective we may use. In fact, it remain completely arbitrary, as long as it amounts to optimizing some reward in an MDP defined on the same state-action space as the safety objective. Two popular exploration algorithms, R-max [11] and Near-Bayesian Exploration [13], rely on such objectives, and, thus, can be made safe.

4 Complexity and parametrization of policy coupling

Since we have shown that safe exploration reduces to policy coupling in all our formulations, we will now focus on the policy coupling problem itself. The following complexity results can be shown by reduction to the Hamiltonian path problem:

1. The policy coupling problem is NP-hard in general.
2. The problem remains NP-hard even if the rewards are known and only the transition probabilities are ambiguous.
3. The problem is tractable when rewards are known.

These results seem natural once we notice the connection between policy coupling and *Partially Observable Markov Decision Processes* (POMDPs), which are known to be NP-hard. Regardless, it is still possible to find good approximate solutions, as we will show later on. We begin by choosing a suitable parametrization for the problem based on the special case that can be solved efficiently: the case of known transition probabilities.

With $P = p$, deterministic, the state the policy coupling problem expressed by Equation 1 reduces to:

$$\max_{\pi} \; E_{\rho,\pi} \sum_t r^{\mu}_{S_t,A_t} \quad \text{such that:} \quad E_{\rho,\pi} \sum_t r^{\sigma}_{S_t,A_t} \geq b \tag{2}$$

where $r^x := E_x R$. This can be formulated as a linear program based on the dual linear program representation of plain MDPs:

$$\begin{aligned} \max_{n} \quad & u_\mu \quad \text{subject to:} \quad u_\sigma \geq b \quad \text{and} \\ x \in \{\mu, \sigma\} \quad & u_x = \sum_{s,a} n_{s,a} E_x[R_{s,a}] \\ \forall s \quad & \sum_a n_{s,a} = \rho_s + \sum_{a,s'} p_{s',a,s} n_{s',a} \end{aligned} \tag{3}$$

where $n_{s,a} = E_{\rho,\pi} \sum_t 1_{S_t=s} 1_{A_t=a}$ is the expected number of times action a is executed in state s. If the corresponding expectation is infinite, as would be the case for sink states, we fix $n_{s,a} = 0$ since we assumed $R_{s,a} = 0$ for those states. The policy can be recovered as $\pi_{s,a}(n) = n_{s,a}/n_s$ when the denominator, $n_s = E_{\rho,\pi} \sum_t 1_{S_t=s} = \sum_a n_{s,a}$, is positive. Otherwise, if $n_s = 0$, simply let $\pi_{s,a}(n)$ be the uniform categorical distribution over the actions available at state s.

5 First order approximation of policy coupling

The most important idea in this article is using special case of policy coupling with known transition probabilities to construct an approximation for the general case. We will pretend the transition probabilities are known and equal to their expectation, $p = E[P]$, and we will, then, apply a correction to the rewards R to compensate. The correction will be such that we obtain a first order approximation when parametrizing with respect to n. That is, we will match both the value function and the gradient of the original problem in the approximation, for all policies, states and actions. We start by looking at the policy gradient [16]:

$$\frac{\partial V^\pi_\rho}{\partial \pi_{s,a}} = n_s Q^\pi_{s,a} \quad \text{where} \quad Q^\pi_{s,a} = R_{s,a} + \sum_{s'} P_{s,a,s'} V^\pi_{\delta(s')} - V^\pi_{\delta(s)},$$

and $V^\pi_\rho := E_{\rho,\pi}[V \mid P, R]$ is the conditional value function. Next, we change variables to $n_{s,a}$:

$$\begin{aligned} n_s Q^\pi_{s,a} = \frac{\partial V^\pi_\rho}{\partial \pi_{s,a}} &= \sum_{s',a'} \frac{\partial V^\pi_\rho}{\partial n_{s',a'}} \frac{\partial n_{s',a'}}{\pi_{s,a}} = \sum_{s',a'} \frac{\partial V^\pi_\rho}{\partial n_{s',a'}} \frac{\partial (n_{s'} \pi_{s',a'})}{\pi_{s,a}} \\ &= n_s \frac{\partial V^\pi_\rho}{\partial n_{s,a}} + \sum_{s'} \left(\sum_{a'} \frac{\partial V^\pi_\rho}{\partial n_{s',a'}} \pi_{s',a'} \right) \frac{\partial n_{s'}}{\pi_{s,a}} \end{aligned}$$

Using the fact that $\sum_a \pi_{s,a} Q^\pi_{s,a} = 0$ for all s, we can easily check that the following is a solution:

$$\frac{\partial V^\pi_\rho}{\partial n_{s,a}} = Q^\pi_{s,a} \tag{4}$$

This also happens to be the *natural policy gradient*, which is known to be informative [17]. Finally, we find a corrected reward, $\bar{R}^\pi_{s,a}$, such that $Q^\pi_{s,a}$ and $V^\pi_{\delta(s)}$ remain the same for all π, s and a when changing transition probabilities from P to $p = E[P]$:

$$\bar{R}^\pi_{s,a} = R_{s,a} + \sum_{s'} (P_{s,a,s'} - p_{s,a,s}) V^\pi_{\delta(s')} \tag{5}$$

Using this corrected reward, we obtain the first order approximation of the policy coupling problem parametrized by $n_{s,a}$:

$$\begin{aligned} &\max_n \quad u_\mu \quad \text{subject to:} \quad u_\sigma \geq b \quad \text{and} && (6) \\ &x \in \{\mu, \sigma\} \quad u_x = \sum_{s,a} n_{s,a} E_x \left[R_{s,a} + \sum_{s'} (P_{s,a,s'} - E[P_{s,a,s}]) V^{\pi(n)}_{\delta(s')} \right] \\ &\forall s \quad \sum_a n_{s,a} = \rho_s + \sum_{a,s'} E[P_{s',a,s}] n_{s',a} \end{aligned}$$

6 Locally optimal iterative solution

If we could find a fixed point of the first order approximation in Equation 6, then that fixed point would be the solution to policy coupling. The simplest way to look for one is to start with some arbitrary n, solve Equation 6, then plug in the solution, solve again and iterate until convergence. In most cases it is difficult to compute the expectations involved in closed form, so, in practice, we would estimate them by importance sampling. At first sight, doing so seems to increase the computational costs significantly. However, the problem structure allows samples to be processed in parallel with minimal communication. If we allow one computing node per sample, then the running time of the importance sampler is essentially the same as that of processing a single sample. This solution is made practical by recent advances in parallel computing software infrastructure and increased public availability of large numbers of compute nodes at reasonable prices as offered, for example, by the Amazon Elastic Compute Cloud.

Even though, in our experience, this naive iteration has worked for some simple problems, it will not converge in general. Instead, it will keep oscillating among a set of policies that might be arbitrarily bad. To prevent such oscillations we need to reduce the amount of change allowed on each iteration, so we impose $\|n^i - n^{i-1}\| \leq k(i)$ where i is the iteration number and $k(i)$ is some "cooling" schedule. If we choose to use the L_2 norm, then we are, essentially performing gradient descent and convergence is easier to analyze, but the complexity of solving the optimization in Equation 6 increases because it becomes a quadratic program. In this case, choosing $k(i) = 1/i$ will guarantee eventual convergence to a local minimum [18]. The L_1 and L_∞ norms seem to work well in practice and they do not increase the complexity of solving the optimization; it remains a linear program. Unfortunately, controlling the number of iterations until convergence seems to be difficult.

7 Approximate but computationally efficient solution

Another idea is to approximate further by using upper and lower bounds on V. For example, if the safety objective encodes the probability of remaining in the safe set, these limits would be 0 and 1. Let $L \geq V_\rho^\pi \geq U$ for any policy π. Then we can see that:

$$\begin{aligned}(P_{s,a,s'} - E[P_{s,a,s}])V_\rho^\pi &\geq \max(0, P_{s,a,s'} - E[P_{s,a,s}])L + \min(0, P_{s,a,s'} - E[P_{s,a,s}])U \\ &:= \Delta^+ P_{s,a,s'} L + \Delta^- P_{s,a,s'} U\end{aligned}$$

Using this lower bound in Equation 6 we get the conservative approximation:

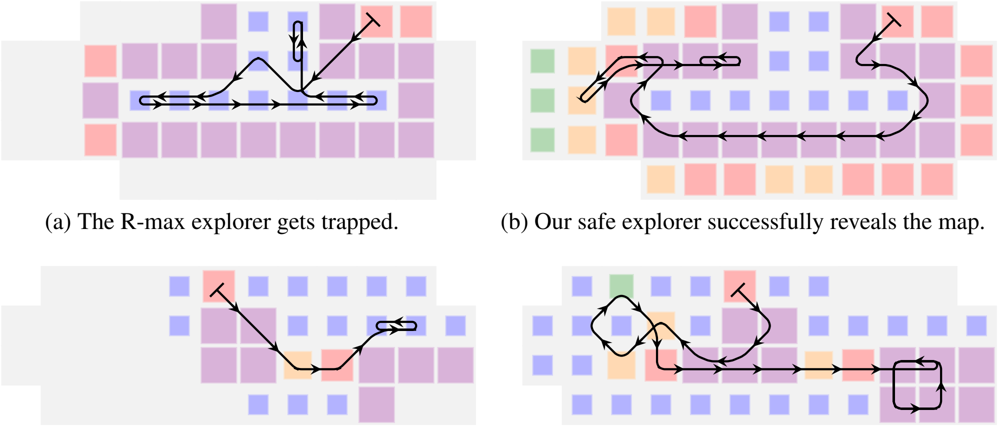
$$\begin{aligned} &\max_n \quad u_\mu \quad \text{subject to:} \quad u_\sigma \geq b \quad \text{and} \\ x \in \{\mu, \sigma\} \quad &u_x = \sum_{s,a} n_{s,a} E_x \left[R_{s,a} + \sum_{s'} \Delta^+ P_{s,a,s'} L + \sum_{s'} \Delta^- P_{s,a,s'} U \right] \\ \forall s \quad &\sum_a n_{s,a} = \rho_s + \sum_{a,s'} E[P_{s',a,s}] n_{s',a} \end{aligned} \tag{7}$$

This is now a plain linear program after removing the non-linear dependence in $V_{\delta(s')}^{\pi(n)}$, and we can solve it efficiently. At the same time, computing the expectations is much easier, and we expect that they will often have closed form solutions; otherwise, we can still approximate them by parallel importance sampling. Overall, solving this conservative approximation of policy coupling has the same theoretical complexity as solving a plain MDP on the underlying state-action space in the dual linear program formulation.

8 Experiments

Our experiment models a terrain exploration problem where the agent has limited sensing capabilities. We consider a simple rectangular grid world, where every state has an integer height between 1 and 5. The agent has eight actions available at any time, $\{N, NE, E, SE, S, SW, W, NW\}$, that represent an attempt to move in the specified direction. Such move will always succeed unless the destination state is higher than the current state by more than 1. In other words, the agent can always go down cliffs, but is unable to climb up if they are too steep. Initially, the heights of all states are ambiguous except for the states immediately neighboring the agent. From our Bayesian standpoint, heights, $H_{x,y}$, are independent, uniformly distributed categorical random variables on the space of allowed heights. Whenever the agent enters a new state it can see the heights of all immediately surrounding states.

Moving from state s to state s' is considered safe if there exists a policy that is guaranteed to bring the system from state s' back to state s based only on the currently known height map. This definition implies that the agent will remain withing the same connected component of the state space



(a) The R-max explorer gets trapped.

(b) Our safe explorer successfully reveals the map.

(c) The R-max explorer acts irreversibly twice.

(d) Our safe explorer first checks reversibility.

Figure 1: Exploration experiments in a simple grid world. See text for full details. Square sizes are proportional to corresponding state heights between 1 and 5. The large, violet squares have a height of 5, while the small, blue squares have a height of 1. Gray spaces represent states that have not yet been observed.

at all time, thus forcing ergodicity. Note that transitions originally considered unsafe might become safe after the height map becomes better known. In other words, we require all actions to be *eventually reversible*, but not necessarily immediately reversible. The exploration criterion is a modified version of R-max exploration [11], where the exploration bonus of moving between two states is proportional to the number of neighboring unknown states that would be uncovered as a result of the move, $m_{s,a}$.

We use the approximate method presented in Section 7 to solve the policy coupling problem corresponding to this exploration scenario. The exploration criterion value function, corresponding to the measure σ, represents the probability that, under the proposed policy, the agent will only take actions that are eventually reversible. Since this safety objective represents a probability, we can immediately bound it between 0 and 1. We set the safety bound, b, to 1. The measure corresponding to the exploration criterion, μ, is such that $P = E[P]$ and $R_{s,a} = E[R_{s,a}] + r_{\text{max}} m_{s,a}$, so it picks out the expected transition probabilities and adds the exploration bonus to the expected rewards.

Figure 1 contrasts the behaviour of safe exploration, as described above, and plain R-max exploration with the modified exploration bonus. We can see that the safe explorer manages to explore the entire connected component it starts in, while the plain R-max explorer risks getting stuck in regions that are impossible to escape.

Our solution has the same theoretical complexity as planning in the unambiguous MDP with known height map, so the same complexity as solving a linear program with $O(|\mathcal{S}|)$ constraints and $O(|\mathcal{S}|)$ variables. When implemented in an efficient programming language (C++ for example) and using a commercial grade LP solver (such as Mosek or CPLEX), our method should scale to grid worlds of 10000 states or more.

References

[1] Pieter Abbeel, A. Coates, and Andrew Y. Ng. Autonomous Helicopter Aerobatics through Apprenticeship Learning. *The International Journal of Robotics Research*, June 2010.

[2] Alexander L. Strehl and Michael L. Littman. A theoretical analysis of Model-Based Interval Estimation. In *Proceedings of the 22nd international conference on Machine learning - ICML '05*, pages 856–863, New York, New York, USA, August 2005. ACM Press.

[3] Shie Mannor and John N. Tsitsiklis. Mean-Variance Optimization in Markov Decision Processes. In *Proceedings of the 28 International Con- ference on Machine Learning*, 2011.

[4] S.I. Marcus, E. Fernández-Gaucherand, D. Hernández-Hernandez, S. Coraluppi, and P. Fard. Risk sensitive Markov decision processes. *Systems and Control in the Twenty-First Century*, 29:263–281, 1997.

[5] Daniel Hernández-Hernández and Steven I. Marcus. Risk sensitive control of Markov processes in countable state space. *Systems & Control Letters*, 29(3):147–155, November 1996.

[6] Erick Delage and Shie Mannor. Percentile optimization in uncertain Markov decision processes with application to efficient exploration. *ICML; Vol. 227*, page 225, 2007.

[7] Arnab Nilim and Laurent El Ghaoui. Robust Control of Markov Decision Processes with Uncertain Transition Matrices. *Operations Research*, 53(5):780–798, 2005.

[8] Oliver Mihatsch and Ralph Neuneier. Risk-Sensitive Reinforcement Learning. *Machine Learning*, 49(2), 2002.

[9] Lihong Li, Michael L. Littman, and Thomas J. Walsh. Knows what it knows: a framework for self-aware learning. In *Proceedings of the 25th international conference on Machine learning*, pages 568–575, 2008.

[10] Sham Kakade, Michael Kearns, and John Langford. Exploration in Metric State Spaces. In *Proceedings of the International Conference on Machine Learning*, page 306, 2003.

[11] Ronen I. Brafman and Moshe Tennenholtz. R-MAX - A General Polynomial Time Algorithm for Near-Optimal Reinforcement Learning. In *Journal of Machine Learning Research*, volume 3, pages 213–231, 2001.

[12] Michael Kearns and Satinder Singh. Near-Optimal Reinforcement Learning in Polynomial Time. *Machine Learning*, 49(2):209–232, November 2002.

[13] J. Zico Kolter and Andrew Y. Ng. Near-Bayesian exploration in polynomial time. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pages 1–8, New York, New York, USA, 2009. ACM Press.

[14] A Hans, D Schneegaß, AM Schäfer, and S Udluft. Safe exploration for reinforcement learning. In *ESANN 2008, 16th European Symposium on Artificial Neural Networks*, 2008.

[15] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction.* MIT Press, 1998.

[16] Richard S Sutton, David Mcallester, Satinder Singh, Yishay Mansour, Park Avenue, and Florham Park. Policy Gradient Methods for Reinforcement Learning with Function Approximation. *Advances in neural information processing systems*, 07932(3):24–24, 1996.

[17] S Kakade. A Natural Policy Gradient. *Advances in Neural Information Processing Systems 14*, 14(26):1531–1538, 2002.

[18] D.P. Bertsekas and J.N. Tsitsiklis. Gradient convergence in gradient methods. *SIAM J. on Optimization*, 10(3):627–642, 2000.