Absolute Safety in Mixed Observability Markov Decision Processes

April 18, 2019

The partially observable Markov decision process (POMDP) is a systematic framework for expressing sequential decision problems where state and outcome uncertainty are important. Although this class of problems is PSPACE complete, and thus unlikely to have efficient general solution methods, recent advances have demonstrated that useful approximate solutions to large POMDPs that represent problems of real-world importance can be found.

There has been little work towards providing absolute safety guarantees (i.e. guarantees with probability 1) for POMDPs. The goal of this document is to describe one specific case where guaranteeing safety through reachability analysis is straightforward. This specific case is useful because it describes a wide variety of real-world problems, most notably human-robot interaction problems.

Specifically, the class studied consists of MOMDPs (mixed observability Markov decision process) where the safety constraint depends only on the fully observable part of the state, though the dynamics can depend on the partially observable part. To guarantee safety, the action space is dynamically constrained so that all actions are safe and lead to safe states.

The key concept is to consider the non-deterministic dynamics to be adversarial from a safety point of view and stochastic, with a partially-observable state, from the optimization point of view. Details are explained below.

1 Problem Description

The general problem that we seek to solve is a multi-stage decision problem where an action $a \in \mathcal{A}$ is chosen at each step. The system dynamics are determined by a function F, which maps a fully observable component of the state, $x \in \mathcal{X}$, the action, and a non-deterministic input, $d \in \mathcal{D}$, to a new state[1]. That is, at each stage, k,

$$x_{k+1} = F(x_k, a_k, d_k). \tag{1}$$

[1]The term "state" is used loosely here. It should be noted that x is not the complete Markov state for the probabilistic problem. In most cases the context should resolve ambiguity, but, when necessary, the terms "fully observable state component", which is always denoted x, and "Markov state", which is always denoted s, will be used for clarity

This problem is approached from two vantage points. From the safety standpoint discussed in Section 1.1, we seek to choose actions so that safety constraints will not be violated for any sequence of d's. From an optimization standpoint, we seek to maximize the expected reward if the d's are random variables with a known joint distributions. This optimization problem is formulated as a partially observable Markov decision process in Section 1.2.

1.1 Safety (Discrete Time)

In this document, safety will be defined with an *instantaneous safety function*, $l(x)$, which maps every state in the fully observable state space, $\mathcal{X}$, to a real number. If $l(x) \geq 0$, then x is considered instantaneously safe. If $l(x) < 0$, then x is unsafe.

To guarantee future safety, a control policy that will keep x in an instantaneously safe region at all time in the future must exist. To reason about this concept, we introduce the *safety value function* or simply *safety function*, L, which will allow us to classify states and actions as safe or unsafe. Intuitively, L represents the minimum safety margin for all times in the future. If $L(x) \geq 0$, then it is possible to avoid instantaneously unsafe states in the future, and x is said to be a *safe state*.

Let the action be determined by a function $a_k = u(x_k, k)$ and the disturbance by $d_k = d(x_k, a_k, k)$. The state-action safety function is defined as follows:

$$L(x, a, k) = \sup_{u} \inf_{d} \min_{i} l(x_i) \tag{2}$$

where

$$\begin{aligned} x_{k+1} &= F(x, a, d(x, a, k)) \\ x_{i+1} &= F(x_i, u(x_i, i), d(x_i, u(x_i, i), i)) \quad \forall\, i \in \{k+1 \,..\, K-1\}. \end{aligned}$$

If $L(x, a, k) \geq 0$, a is said to be a safe action at x. The state safety function is simply

$$L(x, k) = \sup_{a \in \mathcal{A}} L(x, a, k). \tag{3}$$

Calculating $L(x, a, k)$ involves solving a Stackelberg game. Fortunately, in some cases, sufficient approximations may be calculated offline.

correct? Can we say anything about the computational complexity of this game? Is it any easier than a POMDP?

1.2 Mixed Observability Markov Decision Process

From an optimization perspective, we assume that $\{d_k\}_{k=1}^{K}$ are random variables with a known joint distribution. Specifically, the nondeterministic dynamics are governed by a function G, so that

$$d_k = G(s_k, v_k), \tag{4}$$

where $s_k \in \mathcal{S}$ is the Markov state of the system, consisting of the fully-observable state, x_k, and a partially observable state, $\theta_k \in \Theta$, and v_k is a random variable

that is independent at each time step. Crucially, the presence of the hidden state, θ, which links d_k between timesteps, allows for *online learning* and the opportunity to make better decisions by gathering information.

The optimization objective is to maximize the expected value of a reward function, R.

$$\text{maximize}\, \mathrm{E}\left[\sum_{k=1}^{K} \mathcal{R}(s_k, a_k)\right] \tag{5}$$

This problem is a partially observable Markov decision process (POMDP). More specifically, it belongs to the class of mixed observability Markov decision processes (MOMDPs). This MOMDP is defined as follows:

- The **state space** is $\mathcal{S} = \mathcal{X} \times \Theta$.
- The **action space** is $\mathcal{A}$.
- The **observation space** is $\mathcal{X}$. Here we assume that there is no further observation than the fully observable state x_k, but it is straightforward to extend the problem to include partial information about θ_k.
- The **reward function** is $\mathcal{R}$.
- The **transition distribution** is implicitly defined by F, G, and a hidden state transition function, F_θ, defined so that $\theta_{k+1} = F_\theta(s_k, a_k, w_k)$, where w_k is a random variable that is independent at each time step. The combination of these functions will be denoted with $\mathcal{T} = (F, G, F_\theta)$.
- The **observation distribution** is a Kronecker or Dirac delta function centered at x_k, i.e. the agent receives a perfect observation of x_k, and will be denoted compactly as $\mathcal{Z}$.

This POMDP will will be compactly represented as

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{R}, \mathcal{T}, \mathcal{Z}). \tag{6}$$

The optimal solution to $\mathcal{M}$ is a policy, π^* that maps the history of previous actions and observations to the next action to take to optimize Eq. (5). There has been a great deal of research into solving POMDPs approximately both online and offline, and the optimal solution will automatically learn about the hidden state θ. However, if the problem is solved as formulated above, the policy may enter unsafe states.

2 Safe MOMDPs

To guarantee safety, we can prune the action space of $\mathcal{M}$ to only include safe actions. At each state, x, let the safe action space be defined as

$$\mathcal{A}_{\text{safe}}(x, k) = \{a \in \mathcal{A} : L(x, a, k) \geq 0\}\,. \tag{7}$$

A new safe MOMDP, can be defined as

$$\mathcal{M}_{\text{safe}} = (\mathcal{S}, \mathcal{A}_{\text{safe}}, \mathcal{O}, \mathcal{R}, \mathcal{T}, \mathcal{Z}). \tag{8}$$

This problem has the safety property defined in Proposition 1.

Proposition 1. *If $L(x_0, 0) \geq 0$, then, for any policy for $\mathcal{M}_{safe}$, every state that can be reached is safe, i.e. $l(x_k) \geq 0 \, \forall \, k \in \{1..K\}$.*

TODO: write out proof. It is conceptually simple: No action can lead to a state where unsafe states can't be avoided in the future.

3 Continuous-time safety formulation

TODO

4 Potential objections

4.1 The optimal solution to the POMDP will automatically be safe if there is infinite cost assigned to instantaneously unsafe states.

- Yes, but it is hard to find optimal solutions to POMDPs.
- We don't usually know exact probability distributions, we may want to be safe with respect to disturbances that have zero probability.
- If the probability distributions are trained on data, we often will not see the worst-case d_k.