Creating a Scalable Framework for Reinforcement Learning in Norm-Rich Environments

Stephanie Milani, University of Maryland, Baltimore County Katia Sycara, Carnegie Mellon University

Motivation

Goal:

- Enable agents to learn to act in complex environments with many **norms**, prescriptions of desired behavior.
- Develop a scalable framework for norm-rich environments.

Solution:

- Create a modified MDP that compactly represents norms with propositional functions to determine when they have been violated.
- Apply associated modifications for norm violations (reward penalties or transition changes) when they occur.

Background

- **Markov decision process (MDP**) - one of the standard problem representations in reinforcement learning (RL), consisting of:

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$$

(states, actions, transition probability function, rewards, and discount factor)

- Existing frameworks for normative reasoning:
  - **Normative MDPs (NMDPs)** [1] – construct an MDP that considers states with all possible norm violations.
  - **Modular Normative MDPs (MNMDPs)** [2] – construct a new MDP for each norm and norm interaction (avoid using full set of norm-modified states).

Norm Representation

A **norm** consists of:

- A condition function, which is a **propositional function** that determines whether or not a norm has been violated:

$$\mathcal{C} : s_n \times a \times s_{n+1} \rightarrow 0|1$$

- Modifications that will occur if the norm is violated:
  - A transition modification that is applied to $s \in S$:

$$\sigma_{\mathcal{T}} : s_{n+1} \rightarrow s_k$$

  - A reward modification:

$$\sigma_{\mathcal{R}} : r_n \rightarrow r_k$$

  - An associated penalty flag, p.

Norm Framework

- Our representation:

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma, \mathcal{N} \rangle$$

(MDP components with N, an ordered set of norms).

- Can either:
  - Precompute norm-modified MDP or
  - Calculate effect of norms with each transition.
- To determine transition, including effects of norms:
  1. Compute transition using original transition function (penalty flags can be stripped to produce state in original state space since flags are only used by norms).
  2. For next state, evaluate all norms in fixed order.
  3. For each violated norm, apply modifications.
- Application of norms → modified MDP which can be solved with standard RL methods.

State Space Size Analysis

- Time to compute effects of norms on a single transition is linear in number of norms, regardless of interactions.
- Learning time scales with number of states. Norms create at least polynomial more states, so is limiting factor.
- **Our method:**
  - Results in no more states than that created by NMDP.
  - If number of interactions is at most d, only some subsets of p possible penalties can be applied simultaneously.

$$|P| \leq 2^p \leq 2^n \text{ and } |P| \leq \sum_{i=0}^{d} \binom{p}{i} \approx O(p^d)$$

where $|P|$ = number of possible penalty combinations.

Approach	Number of States (order of)	Note
NMDP	2^n [2]	
MNMDP	n^d [2]	If d >n/lg(n), more states created than NMDP.
Our method	$2^p, p^d$	p ≤ n

Table 1: Here is the state space analysis for each of the three methods. The naïve approach, NMDP, is exponential in n, where n is the number of norms that can be either on or off. The MNMDP approach is polynomial in d, where d is the maximum number of interactions between norms. In our method, only penalty flags increase the number of states and each norm sets at most one penalty flag, p, so $p \leq n$. Results in no more states than that created by NMDP.

Conclusion

- Our framework:
  - Can modify transitions, as well as rewards.
  - Has number of states that depends on number of penalty flags, not on number of norms ($p \leq n$).
  - Has computed worst-case bound on number of states of the same order as the average cases for MNMDPs.
  - Avoids redundant states, adding no more states than are in the naïve case.

Future Work

- Our complexity bounds are not tight. Actual performance will likely be better. We will empirically evaluate our method to better estimate actual performance.
- We will conduct this evaluation on scenarios from previous work [2] to provide a better comparison to existing frameworks.
- Domestic service robots are a promising application of normative reasoning, so we will extend this work to a domain of this type.

References

[1] M.S. Fagundes, S. Ossowski, J. Cerquides, and P. Noriega, ``Design and evaluation of norm-aware agents based on normative Markov decision processes," *International Journal of Approximate Reasoning*, 2016.

[2] V. Krishnamoorthy, W. Luo, M. Lewis, and K. Sycara, ``A Computational Framework for Integrating Task Planning and Norm Aware Reasoning for Social Robots," in *IEEE International Conference on Robot and Human Interactive Communication*, 2018.

Acknowledgments

Steph thanks Vignesh Krishnamoorthy for his mentorship and Nicholay Topin for his advice.

THE ROBOTICS INSTITUTE