RoVA: A Method for Robust Value Approximation

Colin R. Small,
Brendan J. Crowe

April 2021

1 Introduction

COVID-19 policy response can be modeled as a control problem: a government (the actor) observes the spread of the disease throughout its population (the state) and implements mandates, lockdowns, and other means of controlling the spread of the disease (the action and the policy). However, optimizing such a policy in a reinforcement learning setting is an inherently difficult problem. High dimensionality of underlying, policy-relevant data (including, but certainly not limited to, the number of infected, exposed, dead, and recovered individuals), makes solving of the problem's value function through traditional methods (e.g. dynamic programming) computationally infeasible. This publicly-available policy-relevant data is often incomplete, inaccurate, and insufficiently covers the state space of our COVID-response control problem to enable the solving of a value function.

There are, however, ways to combat each of these problems. Approximate linear programming (ALP) solves the curse of dimensionality by approximating a value function given a subset of real, observed states and their rewards. However, such an approximated value function is beholden to the accuracy of the observed states used to create it and thus is not robust with regard to uncertainty in these states (such as the inaccuracy of COVID data reporting). Optimizing safe policies in the face of uncertainty in rewards and states can be accomplished with methods such as Bayesian Robust Optimization for Imitation Learning (BROIL)**?**, but such approaches require knowing a set of already-solved reward functions. In our COVID-response control problem, we have neither an already-solved reward function nor trust in the publicly available data used to approximate one.

To the best of our knowledge, we unaware of any efficient methods that combine solutions to these two problems. Thus, we present Robust Value Approximate (RoVA), a method for robust value function approximation.

1.1 ALP

As introduced in the previous section, ALP can be a useful tool for solving reinforcement learning problems when a tabular MDP formulation of the problem is incomplete or infeasible to solve.

This is achieved by approximating the value function $v(s)$ of the problem with a set of state features Φ and corresponding feature weights w. This grants the modeler a much larger degree of flexibility and poses a much easier problem to solve as compared with a tabular MDP. where we fit weights to best approximate a known set of values rather than solving the value function of every possible state directly.

The linear program begin solved is:

$$\begin{aligned} \min_{w} \quad & c^T \Phi w \\ \text{s. t.} \quad & (I - \gamma P_i) \quad \Phi \geq r_i \, i = 1, \ldots, A \end{aligned} \tag{1}$$

Here, we define c to be the distribution over starting states, γ is the discount rate, P_i, r_i are the transition probabilities and reward for each action i, and $A = |\mathcal{A}|$ where $\mathcal{A}$ is the action space.

As may be obvious, this formulation works in the reinforcement learning paradigm is which the reward function $r(s, a)$ is known.

For our problem, we have a large degree of uncertainty in our known reward function we are attempting to approximate. In practive, We may have have some set of candidate reward functions or a distribution over reward functions, but the true reward function is unknown.

For this reason, the problem we are actually trying to solve is an Inverse Reinforcement Learning (IRL) problem, which brings us BROIL**?**.

1.2 BROIL

BROIL is an IRL method that allows for robust optimization over over a set of possible reward functions. This is precisely what we wish to solve within our problem.

However the formulation of BROIL explicitly outlined within the paper assumes a tabular MDP. The problem being solved can be written as the following linear program:

$$\begin{aligned} \min_{u \in \mathbb{R}^{\mathcal{S}, \mathcal{A}}} \quad & p_R^T \rho(\pi, R) \\ \text{s. t.} \quad & \sum_{a \in \mathcal{A}} (I - \gamma P_a^T) u^a = p_0 \\ & u \geq 0 \end{aligned} \tag{2}$$

1.3 RoVA

Our goal to to unify the ideas briefly presented in ALP and BROIL. In doing so we hope to obtain, through solving a linear program, a flexible value function

approximated policy that is robust to the worst case reward in a set of reward functions.

We solve:

$$\begin{aligned} \max_{u,z} \quad & z \\ \text{s.t.} \quad & z \leq r^T u \quad r \in R \\ & \Phi^T A u = \Phi^t c \\ & u \geq 0 \end{aligned} \tag{3}$$

Or its dual

$$\begin{aligned} \min_{w,\xi} \quad & c^T \Phi w \\ \text{s.t.} \quad & A\Phi w \geq \hat{\Phi} \tilde{R} \xi \\ & \mathbb{1}^T \xi = 1 \\ & \xi \geq 0 \end{aligned} \tag{4}$$

1.4 Derivation of the Primal and Dual