Introduction

- In many risk-aware and multi-objective reinforcement learning (MORL) settings the utility of a user is derived from the single execution of a policy.
- In such settings the expected return, or value, does not provide sufficient critical information about the potential positive or adverse effects a decision may have.
- In this case, it is essential to replace the expected value with a posterior distribution over the expected utility of the returns (ESR).
- We propose a novel algorithm, Distributional Monte Carlo Tree Search (DMCTS), which learns a posterior distribution over the expected utility of the returns.
- We implement and demonstrate DMCTS for both risk-aware and multi-objective problems under the ESR criterion.

A full version of the paper can be found at the following link: https://arxiv.org/abs/2102.00966.

Distributional Monte Carlo Tree Search

- To compute the distribution we first calculate the accrued returns, $R_i^t$. The accrued returns is the sum of rewards received during the execution phase as far as timestep, $t$, where $r_i$ is the reward received at each timestep, $\sum_{t=0}^{c_i} r_i$.
- Secondly, we must calculate future returns, $R_{t+1}^t$. The future returns is the sum of the rewards received when traversing the search tree during the learning phase and Monte Carlo simulations from timestep, $t$, to a terminal node, $t_{\text{root}}$, $\sum_j r_{t+1}^t$.
- The cumulative returns, $R_i$, is the sum of the accrued returns, $R_i^t$, and the future returns, $R_{t+1}^t$.

We use a bootstrap distribution to approximate the posterior [2]. To update the bootstrap distribution at each node we use Algorithm 1.

The agent then executes the action, $a^*$, which corresponds to the following:

$$a^* = \arg \max_i \frac{a_i}{\beta_j}.$$

Experiment

- We evaluate DMCTS in a risk-aware problem domain [4] under ESR using the following non-linear utility function: $u = 1 - e^{-\alpha i}$. (1)
- To evaluate DMCTS in the risk-aware domain, we compare DMCTS against Q-learning [5].
- To evaluate DMCTS in a multi-objective setting under ESR, we use the Fishwood problem [3] with the following non-linear utility function: $u = \min \left( \text{fish}, \frac{\text{wood}}{2} \right)$. (2)
- To evaluate DMCTS in the Fishwood domain, we compare DMCTS against C51 [1], EUPG [3], and Q-learning [5].
- As shown in Figure 1 and Figure 2, DMCTS learns good policies in risk-aware settings and achieves state-of-the-art performance in MORL under ESR.

Algorithm 1: Update bootstrap distribution

Input: $i \leftarrow$ Node in the tree
Input: $R_t \leftarrow$ Cumulative Returns
Input: $J \leftarrow$ node bootstrapDistribution
for $j \leftarrow 1$ bootstrap replicates do
Sample $d_j$ from Bernoulli$(\alpha_j/\beta_j)$
if $d_j = 2$ then
$\alpha_j = \alpha_j + w(R_t)$
$\beta_j = \beta_j + 1$
end
end

References