Distributional Monte Carlo Tree Search for Risk-Aware and Multi-Objective Reinforcement Learning*

Extended Abstract

Conor F. Hayes
National University of Ireland Galway
Ireland
c.hayes13@nuigalway.ie

Mathieu Reymond
Vrije Universiteit Brussel
Belgium
mathieu.reymond@vub.be

Diederik M. Roijers
AI lab, Vrije Universiteit Brussel (BE)
& HU Univ. of Appl. Sci. Utrecht (NL)
diederik.yamamoto-roijers@hu.nl

Enda Howley
National University of Ireland Galway
Ireland
enda.howley@nuigalway.ie

Patrick Mannion
National University of Ireland Galway
Ireland
patrick.mannion@nuigalway.ie

ABSTRACT

In many risk-aware and multi-objective reinforcement learning settings, the utility of the user is derived from the single execution of a policy. In these settings, making decisions based on the average future returns is not suitable. For example, in a medical setting a patient may only have one opportunity to treat their illness. When making a decision, just the expected return – known in reinforcement learning as the value – cannot account for the potential range of adverse or positive outcomes a decision may have. Our key insight is that we should use the distribution over expected future returns differently to represent the critical information that the agent requires at decision time. In this paper, we propose Distributional Monte Carlo Tree Search, an algorithm that learns a posterior distribution over the utility of the different possible returns attainable from individual policy executions, resulting in good policies for risk-aware settings. Moreover, our algorithm outperforms the state-of-the-art in multi-objective reinforcement learning for the expected utility of the returns.

KEYWORDS

Multi-objective; risk-aware; decision making; distributional; reinforcement learning; Monte Carlo tree search

ACM Reference Format:
Conor F. Hayes, Mathieu Reymond, Diederik M. Roijers, Enda Howley, and Patrick Mannion. 2021. Distributional Monte Carlo Tree Search for Risk-Aware and Multi-Objective Reinforcement Learning: Extended Abstract. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

In reinforcement learning (RL) settings, the expected return is used to make decisions. In many scenarios, the utility of a user is derived from the single execution of a policy [12]. In this case the expected return does not provide the agent with sufficient critical information about the potential positive or adverse outcomes a decision may have. In order for an agent to have sufficient critical information at decision time, it is crucial to replace the expected return with a posterior distribution over the expected utility of returns.

In the real world, decision-making often involves trade-offs based on multiple conflicting objectives [5, 8]. Many approaches to multi-objective decision-making only consider linear utility functions; this limitation severely restricts the real-world applicability of these methods, given that utility in many real-world problems is derived in a non-linear manner.

In the multi-objective case, optimising under the expected utility is known as the expected scalarised returns (ESR) [10, 12]. If the utility function is non-linear strictly multi-objective methods are required to find optimal solutions.

We propose a novel algorithm, Distributional Monte Carlo Tree Search (DMCTS), which learns a posterior distribution over the expected utility of the returns. DMCTS learns a posterior distribution over the utility of the returns by executing multiple individual policies and calculating the utility of the returns obtained from each policy execution. DMCTS builds upon Monte Carlo Tree Search (MCTS) [7, 14, 17]. Our key insight is that learning a posterior distribution over the utility of the returns is essential when optimising for risk-aware RL and under the MORL ESR criterion. We implement and demonstrate DMCTS for both risk-aware and multi-objective problems under the ESR criterion. DMCTS learns good polices in risk-aware settings. Moreover, DMCTS outperforms the state-of-the-art in MORL under ESR.

2 DISTRIBUTIONAL MONTE CARLO TREE SEARCH

The majority of RL research focuses on learning an optimal policy based on the expected returns, known as the value. Under the expected scalarised returns (ESR), a single execution of a policy is used to derive the utility of a user. A distribution over the expected utility of returns must be used when making decisions under the ESR criterion as a distribution provides the agent with sufficient information at decision time to exploit positive outcomes and avoid negative outcomes.

We present our Distributional Monte Carlo Tree Search (DMCTS) algorithm which learns a posterior distribution over the expected returns. DMCTS builds an expectimax search tree through the same

*An extended version of this paper is available [6].

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

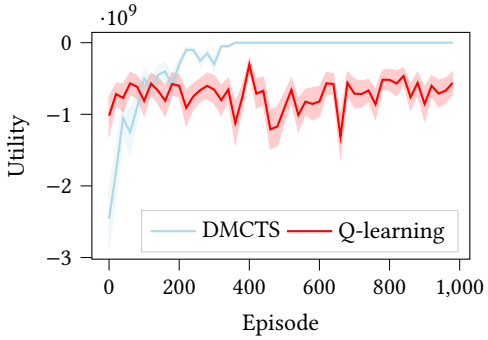

Figure 1: Results from the risk-aware environment.

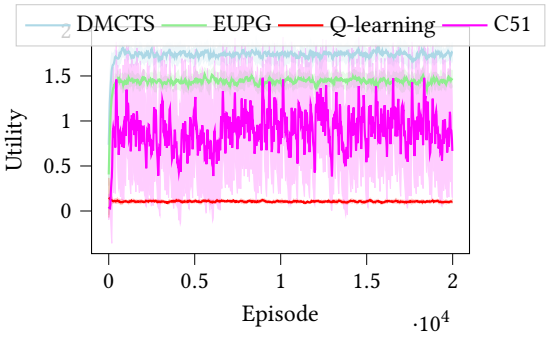

Figure 2: Results from the Fishwood environment.

process as MCTS [16]. Learning a posterior distribution over the utility of the returns can be used to replace the expected future returns (of vanilla MCTS) at each node.

To compute the distribution we first calculate the accrued returns, $\mathbf{R}_t^-$. The accrued returns is the sum of rewards received during the execution phase as far as timestep, t, where $\mathbf{r}_t$ is the reward received at each timestep,

$$\mathbf{R}_t^- = \sum_{0}^{t-1} \mathbf{r}_t.$$

Secondly, we must calculate future returns, $\mathbf{R}_t^+$. 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_n,

$$\mathbf{R}_t^+ = \sum_{t}^{t_n} \mathbf{r}_t.$$

The cumulative returns, $\mathbf{R}_t$, is the sum of the accrued returns, $\mathbf{R}_t^-$, and the expected future returns, $\mathbf{R}_t^+$. $\mathbf{R}_t$ is backpropagated to each node in the search tree, where the utility is computed, $u(\mathbf{R}_t)$.

At each node we aim to maintain a posterior distribution over the expected utility of the returns. However, because the utility function may be non-linear, a parametric form of the posterior distribution may not exist. Since a bootstrap distribution can be used to approximate a posterior [4, 9], it is much more suitable to maintain a bootstrap distribution over the expected utility of the returns at each node.

Each bootstrap distribution contains a number of bootstrap replicates, $j \in \{1, ..., J\}$ [2]. On initialisation of a new node, for each bootstrap replicate, j, the parameters α_j and β_j are both set to 1.

During the backpropagation phase the bootstrap distribution at each node is updated. At node i, for each bootstrap replicate, j, a coin flip is simulated. If the result of the coin flip is equal to 1 (heads), α_{ij} and β_{ij} are updated:

$$\alpha_{ij} = \alpha_{ij} + u(\mathbf{R}_t)$$

$$\beta_{ij} = \beta_{ij} + 1$$

To select actions while learning, we use the previously computed statistics. At node n, we select an action by sampling the bootstrap distribution at each child node, i. For each sampled bootstrap replicate, j, the α_{ij} and β_{ij} values are retrieved and $\frac{\alpha_{ij}}{\beta_{ij}}$ is computed. Since the following is true,

$$\frac{\alpha_{ij}}{\beta_{ij}} \equiv \mathbb{E}[u(\mathbf{R}_t^- + \mathbf{R}_t^+)], \tag{1}$$

by maximising over i in Equation 1, we select an action corresponding to j approximately proportionally to the probability of that action being optimal – as per the Bootstrap Thompson Sampling [2, 3] exploration strategy. The agent then executes the action, a^*, which corresponds to the following:

$$a^* = \arg\max_i \frac{\alpha_{ij}}{\beta_{ij}}.$$

We note that at execution time we can simply select the overall maximising action by averaging over all the acquired data, thereby maximising the ESR criterion:

$$ESR = \mathbb{E}[u(\mathbf{R}_t^- + \mathbf{R}_t^+)]. \tag{2}$$

3 EXPERIMENTS

Before testing DMCTS on benchmark problems from the MORL literature, we evaluate DMCTS in a risk-aware problem domain under ESR. Shen et al. [13] define a risk-aware MDP where an agent must decide from a number of stocks in which to invest. To evaluate our DMCTS algorithm we use the following risk-averse non-linear utility function:

$$u = 1 - e^{-r_t}. \tag{3}$$

As shown in Figure 1, DMCTS consistently learns the optimal policy for the above risk-averse utility function. The policy, which avoids all risk, has a cumulative utility of 0. Q-learning struggles to learn a stable policy for the given utility function.

To evaluate DMCTS in a multi-objective setting under ESR, we use the Fishwood problem [11]. For Fishwood we maximise the following non-linear utility function [11],

$$u = \min\left(\texttt{fish}, \left\lfloor \frac{\texttt{wood}}{2} \right\rfloor\right). \tag{4}$$

To evaluate DMCTS in the Fishwood domain, we compare DMCTS against C51 [1], Expected Utility Policy Gradient (EUPG) [11, 18], and Q-learning [15]. EUPG achieves state-of-the-art results in the Fishwood problem under ESR [11]. As shown in Figure 2, Q-learning and C51 fail to learn any meaningful policy. By contrast, DMCTS and EUPG outperform both C51 and Q-learning. DMCTS reaches a higher utility when compared to EUPG and achieves state-of-the-art performance under ESR in the Fishwood environment.

REFERENCES

[1] Marc G. Bellemare, Will Dabney, and Rémi Munos. 2017. A Distributional Perspective on Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17)*. JMLR.org, Sydney, NSW, Australia, 449–458.

[2] Dean Eckles and Maurits Kaptein. 2014. Thompson sampling with the online bootstrap. *CoRR* abs/1410.4009 (2014). arXiv:1410.4009 http://arxiv.org/abs/1410.4009

[3] Dean Eckles and Maurits Kaptein. 2019. Bootstrap Thompson Sampling and Sequential Decision Problems in the Behavioral Sciences. *SAGE Open* 9, 2 (2019), 2158244019851675. https://doi.org/10.1177/2158244019851675 arXiv:https://doi.org/10.1177/2158244019851675

[4] Bradley Efron. 2012. Bayesian inference and the parametric bootstrap. *Ann. Appl. Stat.* 6, 4 (12 2012), 1971–1997. https://doi.org/10.1214/12-AOAS571

[5] Conor F. Hayes, Enda Howley, and Patrick Mannion. 2020. Dynamic Thresholded Lexicograpic Ordering. *Adaptive and Learning Agents Workshop (AAMAS 2020)*.

[6] Conor F Hayes, Mathieu Reymond, Diederik M Roijers, Enda Howley, and Patrick Mannion. 2021. Risk-Aware and Multi-Objective Decision Making with Distributional Monte Carlo Tree Search. *arXiv preprint arXiv:2102.00966* (2021). https://arxiv.org/abs/2102.00966

[7] Levente Kocsis and Csaba Szepesvári. 2006. Bandit Based Monte-Carlo Planning. In *Machine Learning: ECML 2006*, Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 282–293.

[8] Patrick Mannion, Sam Devlin, Jim Duggan, and Enda Howley. 2018. Reward shaping for knowledge-based multi-objective multi-agent reinforcement learning. *The Knowledge Engineering Review* 33 (2018), e23. https://doi.org/10.1017/S0269888918000292

[9] Michael Newton and Adrian Raftery. 1994. Approximate Bayesian Inference by the Weighted Likelihood Bootstrap. *Journal of the Royal Statistical Society Series B-Methodological* 56 (01 1994), 3 – 48.

[10] Roxana Rădulescu, Patrick Mannion, Diederik M. Roijers, and Ann Nowé. 2020. Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems* 34, 10 (2020).

[11] Diederik M Roijers, Denis Steckelmacher, and Ann Nowé. 2018. Multi-objective Reinforcement Learning for the Expected Utility of the Return. In *Proceedings of the Adaptive and Learning Agents workshop at FAIM 2018*.

[12] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research* 48 (2013), 67–113.

[13] Yun Shen, Michael J. Tobia, Tobias Sommer, and Klaus Obermayer. 2014. Risk-Sensitive Reinforcement Learning. *Neural Computation* 26, 7 (2014), 1298–1328.

[14] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* (2016).

[15] Kristof Van Moffaert, Madalina M Drugan, and Ann Nowé. 2013. Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. IEEE, 191–199.

[16] Joel Veness, Kee Siong Ng, Marcus Hutter, William Uther, and David Silver. 2011. A Monte-Carlo AIXI Approximation. *J. Artif. Int. Res.* 40, 1 (Jan. 2011), 95–142.

[17] Weijia Wang and Michèle Sebag. 2012. Multi-objective Monte-Carlo Tree Search *(Proceedings of Machine Learning Research, Vol. 25)*, Steven C. H. Hoi and Wray Buntine (Eds.). PMLR, Singapore Management University, Singapore, 507–522.

[18] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3-4 (1992), 229–256.