Tailored Expectile Imputation Algorithms for Efficient Expectile-based Distributional Reinforcement Learning

Matthew Farrugia-Roberts
Master of Computer Science
The University of Melbourne
farrugiam@student.unimelb.edu.au

Word count*: 2199 words.

1. Introduction

Reinforcement learning (RL) is a framework for algorithms that leverage machine learning to solve decision-making problems [1]. RL is applicable in myriad important technological domains, from low-level robotic motor control [2] to high-level strategic control of autonomous systems such as self-driving vehicles [1,3]. Thus, improving the efficiency and efficacy of RL algorithms entails wide-ranging practical benefits.

In this research plan, we consider establishing one such improvement after excising an efficiency bottleneck from an otherwise-leading algorithm. The remainder of this section contextualises and motivates our approach within the RL field, and states our research question. In section 2, we detail our proposed methods of investigation and analysis for validating our hypothesis. We conclude this short plan in section 3 by summarising the hypothetical contribution of our innovation.

1.1. Goals of the field: *Efficacious* and *efficient* reinforcement learning

The leading evaluative dimension for RL algorithms is *efficacy*. Given experience interacting in a decision-making environment coupled with some notion of 'reward', *efficacious* RL algorithms are those whose learned behaviours lead to highly 'rewarding' consequences [1][1].

A second dimension weighs in appraising RL algorithms for practical use: One seeks *efficiency* alongside efficacy. For example, an algorithm may outpace another by requiring less computational effort or time to process each interaction during training.

Distributional RL (DRL) is a recent and promising paradigm for approaching RL [4,5]. DRL algorithms learn a reward *probability distribution* given each decision-making situation, and then make decisions based on these distributions [4]. As the author has reviewed [5], early DRL algorithms (for example [2,4,6–8]) have consistently matched or surpassed previous state-of-the-art RL algorithms in efficacy with, predominantly, comparable efficiency—a clear improvement. Moreover, in this nascent paradigm, refining these initial approaches may yet reveal further improvements [5]. We seek just such a refinement.

*Count by `texcount -1 -sum` (*à la* Overleaf). Excludes front matter and references. Breakdown at back.

[1] The formal details of RL, including notions of 'reward', are beyond our scope. We refer readers to Sutton and Barto [1] for a comprehensive introduction.

Of various initial approaches, Rowland et al.'s *expectile-based* DRL algorithm shows particularly strong efficacy [5,8]. Its eponymous *expectiles* are statistics for summarising probability distributions, analogous to the well-known *quantiles* [9,10]. The regression literature knows expectiles as well-suited for learning probability distributions [5, 11–14]. With Rowland et al.'s empirical results, this supports pursuing expectile-based DRL algorithms.

Despite its leading efficacy, Rowland et al.'s algorithm lags competitors in efficiency due to a critical bottleneck. To process each unit of experience, the algorithm must solve a sub-problem called *expectile imputation.* Here, Rowland et al.'s algorithm resorts to a *numerical optimisation* routine [8,15]. This general-purpose routine solves the sub-problem, but without taking advantage of its structure, and at significant computational cost[2].

A variant of Rowland et al.'s algorithm with a more efficient solution to this sub-problem could become a leading performer among DRL algorithms in both efficacy *and* efficiency. Below, we review the expectile imputation sub-problem and consider tailored replacements for Rowland et al.'s naive numerical optimisation sub-routine.

1.2. Removing the expectile imputation bottleneck

Expectile-based DRL involves summarising probability distributions with collections of *expectiles* (Figure 1a). However, Rowland et al. establish that such *expectile summaries* are inconvenient for incorporating decision-making experiences during training [8]. Thus, their expectile-based DRL algorithm converts expectile summaries into a more convenient form—a *sample of values* with a matching distribution—before assimilating each unit of experience. This conversion sub-problem is an instance of so-called *expectile imputation* [8] (Figure 1b).

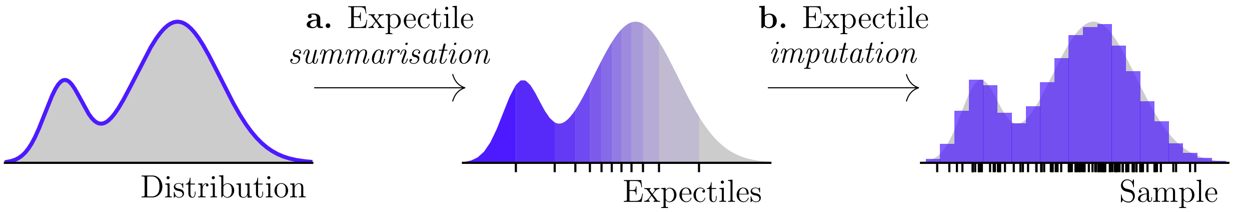

Figure 1: a. Like *quantiles*, a set of *expectiles* summarises a probability distribution. **b.** *Expectile imputation problem:* Given such a set of expectiles, find a *sample* with a matching distribution.

We consider averting this conversion bottleneck in Rowland et al.'s algorithm. By retaining most of their algorithm, but replacing their expensive numerical optimisation sub-routine with an algorithm that exploits the expectile imputation sub-problem's structure, we should achieve a more efficient but equally efficacious DRL algorithm.

Work from outside DRL may reveal such an expectile imputation algorithm. For example, Schnabel and Eilers [16] present an algorithm for an equivalent problem (their 'nonparametric density estimate' is our 'sample of values'). Additionally, Newey and Powell [9] analytically connect a distribution's expectile summary with its density function, an insight that may lead to a second algorithm. The relevance of this work has not yet been adequately acknowledged within DRL [5,8], which leads to our research question:

> **Research question:** Do these results [9,16] (1) represent more efficient *expectile imputation* algorithms, (2) thereby giving us more efficient but equally efficacious *expectile-based DRL* algorithms, compared to Rowland et al.'s algorithms [8]?

[2] Rowland et al. report training their algorithm in comparable time to competing algorithms, but they parallelise training across multiple computers [8]. With fixed available hardware, training time would increase. Thus, we consider this algorithm less efficient despite the equivalent absolute training time.

2. Methods

Mirroring our two-part question, we separate our study into two stages: First, we will validate our alternative algorithms for the expectile imputation sub-problem (section 2.1). Second, we will study the resulting modified expectile-based DRL algorithm (section 2.2).

This division is prudent: The expectile imputation sub-problem is self-contained, so we can establish our new algorithms independently of the complicating DRL context. Moreover, as is common in RL (for example [4, 7, 8]), we may thereby demonstrate the conceptual feasibility of our approach before committing the significant computational resources our second stage demands. This subsequent investment is still necessary, however, in line with the field's practical expectations [1,6]. Below, we outline our proposed methods for investigating each stage and for analysing the results of these investigations.

2.1. Tailored expectile imputation algorithms

Our two alternative expectile imputation algorithms are a new algorithm based on Newey and Powell's theorem [9, Theorem 1iv] and the algorithm from Schnabel and Eilers [16]. In our first stage, we seek to establish (1) that these algorithms indeed solve the expectile imputation problem, and (2) their improved efficiency over Rowland et al.'s numerical baseline [8].

Establishing correctness: Before comparing our algorithms' efficiency we must establish that they indeed solve the expectile imputation problem. Fortunately, mathematical derivations of our algorithms' behaviour should be within reach: Our algorithms are already based on analytical results that—alongside the existing theory of expectile statistics [9, 10, 17]—present a convenient starting point. Therefore, we will pursue *proofs of correctness* as a first step in establishing our algorithms.

After securing this analytical support, we may still desire empirical validation of our algorithms' correctness. However, any practical issues should emerge during our empirical evaluation of their *efficiency*, which we consider below.

Establishing improved efficiency: Unfortunately, analytical tractability may not extend to our efficiency investigation. Our baseline, a general-purpose numerical optimisation routine, permits conservative analysis [18, as cited in 19], but may resist yielding results relevant to our special case. Since we must compare against this baseline, theoretical run-time bounds on our alternatives will be fruitless. Therefore, we will eschew formal analysis and instead perform a simple empirical run-time experiment.

Our investigation must begin with a faithful reproduction of Rowland et al.'s baseline routine—a crucial step in performing a meaningful comparison. We will closely follow the details in the works involving this baseline [8, 15] in our implementation.

We will then measure the run-time of our baseline and our two alternative algorithms. For a robust comparison, we will fix the available hardware and aggregate results from multiple trials. To elicit any hardware-independent run-time trends, and to ensure relevant results, we will explore a range of input and output parameters typical of Rowland et al.'s DRL experiments [8]. We will synthesise inputs using Gaussian mixture modules—an expressive distribution family which itself underpins competing DRL algorithms [2, 20]. Here we diverge from Schnabel and Eilers' precedent: They demonstrate their algorithm on real-world regression data [16]. However, they emphasise correctness, whereas we require a flexible data source for our wide-ranging run-time experiments. Furthermore, RL works often establish conceptual results like those of our first stage experiments on similar 'toy' data [1, 7, 8].

Analysis: From our correctness analysis, we seek a guarantee that our algorithms always give suitable outputs. A derivation as in Rowland et al.'s baseline [8] would suffice. We would permit typical mathematical assumptions, such as indefinitely large input sets, if our empirical efficiency investigation reveals no behavioural problems given typical parameters.

From the efficiency experiments themselves, we seek a clear speed-up over our baseline, such as an order-of-magnitude improvement over the full parameter range. Such a resounding improvement seems possible due to the baseline's naivety [8]. This would positively answer the first part of our research question and position us to resolve the second in kind.

However, while we have isolated the expectile imputation sub-problem, the broader DRL context remains our main motivating focus: Ultimately, we need improvements for the inputs these algorithms would face *within a DRL algorithm, during training*. The nature of such inputs is unclear *a priori*—we require richer experiments, to which we now turn.

2.2. Efficient expectile-based distributional reinforcement learning

Having established alternative algorithms for solving the expectile imputation sub-problem, we will restore the full DRL context, and address the second part of our question. In particular, we will investigate (1) whether Rowland et al.'s DRL algorithm sustains its efficacy when modified with our replacement expectile imputation algorithms, and (2) whether the expectile imputation algorithms sustain their speed-up in the DRL setting.

Both aspects of this investigation begin by carefully reproducing Rowland et al.'s DRL algorithm following published details [8], to establish our baseline and host our modifications. Using Drummond's terminology [21], such a 'reproduction' is harder than a mere 'replication' based on Rowland et al.'s *code*, but this code is unavailable, and a first independent confirmation of their efficacy results would itself contribute to the emerging DRL field.

Establishing equivalent efficacy: Following Rowland et al. [8], we will employ the popular *Atari 2600* environment suite [22, 23] to measure efficacy. This benchmark contains dozens of decision-making tasks based on video games from the classical console, and is suitable for evaluating the efficacy of RL algorithms because these games are, by design, both difficult for humans and diverse [22, 23]. While we could adapt our algorithm to other benchmark environment suites, this choice will also allow direct comparison to Rowland et al.'s results.

We will follow the best practices laid out by Machado et al. [23], as followed by Rowland et al. and similar works [4, 6–8]. This includes tuning the parameters of our baseline and modified algorithms on just 6 games before evaluating a final choice of parameters on a gauntlet of 57 games [23], and computing the progression of human-normalised scores [24] throughout training (the so-called *learning curves*) to measure and report our algorithms' efficacy [23]. Following these standards will help us compare to Rowland et al.'s results to validate our reproduction, and will help others compare to our results in the future [23].

Establishing improved efficiency: To investigate efficiency, we will run *separate* timing experiments with our baseline and modified algorithms, using their most efficacious parametrisations, and exploring a range of *Atari 2600* games and training stages. Partially decoupling our efficiency and efficacy experiments (rather than timing our whole efficacy experiments) is pragmatic: Evaluating *efficacy* involves simulating dozens of games, each for millions of frames, allowing each algorithm enough time to demonstrate flexible *learning ability* [22, 23]. So much time would be redundant in measuring *efficiency*, even in diverse situations. Conversely, we must serialise our baseline's expensive expectile imputation steps to measure efficiency, but we can otherwise parallelise these steps to speed up training [8].

Analysis: To compare the efficacy of different RL algorithms across multiple games, one usually compares a robust summary of scores such as their *median* [4,6–8]. This accounts for each algorithm having different strengths and weaknesses, while discounting outlying scores from overly easy or hard games [6]. However, we face a unique situation in comparing algorithms with hypothetically identical behaviour, and therefore performance, in all games—aggregating results might even obscure differences. We will report these summaries to facilitate future comparisons [23], but to answer our question, we will instead perform a detailed comparison of the per-game learning curves for our baseline and modified algorithms. If these learning curves 'line up', then we can conclude that our modifications have preserved the algorithm's efficacy.

In contrast, we expect stark differences between our algorithms' *efficiency*. Echoing section 2.1, we seek a clear speed-up over our baseline in all games, though perhaps shy of an order-of-magnitude improvement, since other components may begin to dominate the DRL algorithm's run-time. Instead, we would consider reaching the efficiency of competing DRL algorithms (for example [4,6,7,25]) sufficient improvement to affirm our question, since our evaluation involves diverse and representative learning situations.

In summary, if our algorithms indeed display equivalent efficacy with increased efficiency, we will have successfully excised Rowland et al.'s bottleneck, affirming our research question, and achieving state-of-the-art performance in both of our crucial dimensions.

3. Contribution

We conclude by speculating on some likely contributions of our work. If we establish an alternative expectile imputation algorithm based on Newey and Powell's theorem [9], this first-stage result may be a contribution in its own right, with potential applications outside DRL as illustrated by Schnabel and Eilers [16]. Our first-stage results may also highlight the connections between the new field of DRL and the existing regression literature, which, as the author has argued, has not yet been sufficiently acknowledged [5].

Turning to the full DRL context, if we reproduce Rowland et al.'s baseline results [8], this will constitute a valuable first independent validation of the expectile-based approach to DRL [5,21]. If we additionally succeed in establishing a more efficient but equally efficacious expectile-based DRL algorithm, this algorithm would reach the state of the art in RL by eliminating the main drawback of Rowland et al.'s contribution [5].

DRL is still young, and a competing algorithm may eventually emerge ahead of our successful approach [5]. In any case, our improvements should reduce the computational cost of further experimentation with future *expectile-based* DRL algorithms, as long as they still involve expectile imputation. Furthermore, by following Machado et al.'s conventions for evaluating efficacy in the *Atari 2600* environment suite, we will facilitate comparisons to our results in future work [23]. Our work may, thereby, continue to help this new paradigm advance towards its exciting potential.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.

[2] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, T. Dhruva, A. Muldal, N. Heess, and T. Lillicrap, "Distributed distributional deterministic policy gradients," in *International Conference on Learning Representations*, poster, 2018.

[3] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robotics and Autonomous Systems*, vol. 114, pp. 1–18, 2019.

[4] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 449–458, 2017.

[5] M. Farrugia-Roberts, "The promise of distributional reinforcement learning." Submission for *COMP90044 Research Methods*, The University of Melbourne, Semester 2 2020.

[6] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. G. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 3215–3222, 2018.

[7] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pp. 2892–2901, 2018.

[8] M. Rowland, R. Dadashi, S. Kumar, R. Munos, M. G. Bellemare, and W. Dabney, "Statistics and samples in distributional reinforcement learning," in *International Conference on Machine Learning*, pp. 5528–5536, 2019.

[9] W. K. Newey and J. L. Powell, "Asymmetric least squares estimation and testing," *Econometrica: Journal of the Econometric Society*, pp. 819–847, 1987.

[10] M. C. Jones, "Expectiles and m-quantiles are quantiles," *Statistics & Probability Letters*, vol. 20, no. 2, pp. 149–153, 1994.

[11] T. Kneib, "Beyond mean regression," *Statistical Modelling*, vol. 13, no. 4, pp. 275–303, 2013.

[12] L. S. Waltrup, F. Sobotka, T. Kneib, and G. Kauermann, "Expectile and quantile regression—david and goliath?," *Statistical Modelling*, vol. 15, no. 5, pp. 433–456, 2015.

[13] B. Efron, "Regression percentiles using asymmetric squared error loss," *Statistica Sinica*, pp. 93–125, 1991.

[14] J. W. Taylor, "Estimating value at risk and expected shortfall using expectiles," *Journal of Financial Econometrics*, vol. 6, no. 2, pp. 231–252, 2008.

[15] W. Dabney, Z. Kurth-Nelson, N. Uchida, C. K. Starkweather, D. Hassabis, R. Munos, and M. Botvinick, "A distributional code for value in dopamine-based reinforcement learning," *Nature*, vol. 577, no. 7792, pp. 671–675, 2020.

[16] S. K. Schnabel and P. H. Eilers, “A location-scale model for non-crossing expectile curves,” *Stat*, vol. 2, no. 1, pp. 171–183, 2013.

[17] H. Holzmann, B. Klar, *et al.*, “Expectile asymptotics,” *Electronic Journal of Statistics*, vol. 10, no. 2, pp. 2355–2371, 2016.

[18] M. J. Powell, “A hybrid method for nonlinear equations,” *Numerical methods for nonlinear algebraic equations*, pp. 87–114, 1970.

[19] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, “User guide for minpack-1,” tech. rep., Argonne National Lab., IL, USA, 1980.

[20] Y. Choi, K. Lee, and S. Oh, “Distributional deep reinforcement learning with a mixture of gaussians,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9791–9797, IEEE, 2019.

[21] C. Drummond, “Replicability is not reproducibility: Nor is it good science,” *Proceedings of the Evaluation Methods for Machine Learning Workshop at the 26th ICML*, 2009.

[22] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, “The arcade learning environment: An evaluation platform for general agents,” *Journal of Artificial Intelligence Research*, vol. 47, pp. 253–279, 2013.

[23] M. C. Machado, M. G. Bellemare, E. Talvitie, J. Veness, M. Hausknecht, and M. Bowling, “Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents,” *Journal of Artificial Intelligence Research*, vol. 61, pp. 523–562, 2018.

[24] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[25] A. Gruslys, W. Dabney, M. G. Azar, B. Piot, M. G. Bellemare, and R. Munos, “The reactor: A fast and sample-efficient actor-critic agent for reinforcement learning,” in *International Conference on Learning Representations*, poster, 2018.

Word count (detailed breakdown)

This word count is computed by the `texcount` utility, which is the same tool used within Overleaf. The utility counts words in the main text, headings, footnotes, figure captions, equations, and in-text citations but omits those in the front matter, references, and this section, and counts some words in in-text citations like ‘et al.’ as single words.

Command:

```
texcount main.tex -sum
```

Output:

```
File: main.tex
Encoding: ascii
Sum count: 2199
Words in text: 2073
Words in headers: 39
Words outside text (captions, etc.): 87
Number of headers: 13
Number of floats/tables/figures: 1
Number of math inlines: 0
Number of math displayed: 0
Subcounts:
  text+headers+captions (#headers/#floats/#inlines/#displayed)
  121+1+0 (1/0/0/0) Section: Introduction
  295+9+59 (1/0/0/0) Subsection
  204+5+28 (1/1/0/0) Subsection: Removing the expectile imputation bottleneck
  118+1+0 (1/0/0/0) Section: Methods
  517+10+0 (4/0/0/0) Subsection: Tailored expectile imputation algorithms
  606+12+0 (4/0/0/0) Subsection: Efficient expectile-based distributional reinforce
  212+1+0 (1/0/0/0) Section: Contribution
```

Total:

2199 words.