The Promise of Distributional Reinforcement Learning

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

Word count*: 2087 words.

1. Introduction

Reinforcement learning concerns the design of algorithms that learn to make better decisions from experience. This broad framework permits many approaches and finds diverse applications, from systems for game-playing and robotic control to models of animal learning [1].

The *value-based* approach to reinforcement learning is as follows: First, observe the outcomes of decisions in various situations. Then, estimate a *value function* capturing the average long-run consequences of each decision in each situation. With this function, make decisions based on the estimated *value* of each available option in the current situation [1]. The 'value' in 'value function' refers to *expected value*—one averages out random variation while estimating long-run consequences. Accordingly, recent work [2] proposes an extension to value-based reinforcement learning, asking: If one estimated the *probability distribution* over long-run consequences itself (rather than its average), would that lead to more effective learning? This *distributional* paradigm shift reveals a large space of new reinforcement learning algorithms for exploration, with early experiments yielding impressive empirical performance [2–9].

In this review, we attempt to orient ourselves in this new fertile landscape, seeking directions towards realising the potential of distributional reinforcement learning. We begin in section 2 by summarising the fundamentals of the traditional value-based paradigm. We then outline its distributional extension in section 3, and discuss its particular empirical and theoretical properties. In section 4, we review several recently-proposed distributional algorithms. We conclude by highlighting paths for further investigation.

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

2. Value-based reinforcement learning

We review the fundamental notions informing our discussion of value-based and distributional methods. See Sutton and Barto [1], on which this section is based, for a more comprehensive and formal introduction to reinforcement learning.

In reinforcement learning (RL), one models a sequential decision-making task with three entities: An *agent*, its *environment*, and a *reward function*. In each of a sequence of *interactions*, the agent observes the current *state* of the environment, and selects an available *action* using an action-selection *policy*. In response, the environment transitions to a new state. Finally, the reward function assigns a numerical *reward* to the action and transition. Importantly, each step (the action selection, the state transition, and the reward assessment) may involve randomness.

RL algorithms aim to produce action-selection policies that maximise, in expectation, a sum of the rewards assigned to each interaction in an interaction sequence. This total cumulative reward is called the *return*. With it, one can express succinctly the goal of RL algorithms: *Find action-selection policies achieving high expected return.*

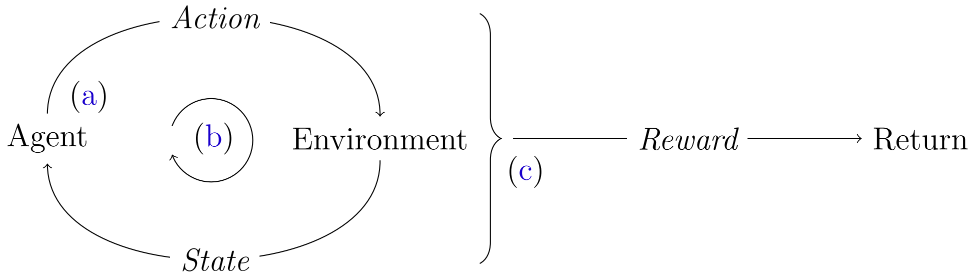

Figure 1: Reinforcement learning (RL): Find an *action-selection* policy (a) leading to *interaction sequences* (b) that, according to the *reward function* (c), accumulate high *return*.

A common approach for finding return-maximising policies is to estimate (from sample interactions) the *conditional expected value* of the return[1] given each environmental state and prospective action. This is the so-called *value function*: A function with states and actions as input, and the corresponding conditional expected return value as output. Return values inherently account for the long-term consequences of actions. Thus, with a value function, a policy can be comparatively simple—a 'greedy' policy selecting the maximum-value action given the environment's current state often suffices.

Value functions come in different sizes, and one distinguishes several learning *settings*: In environments with a finite number of states and available actions, one may separately estimate the function's output for each possible input; as if filling in a table. This is the so-called *tabular* setting. Where the number of states or actions is infinite (or intractable), one must resort to estimating a finitely-parametrised approximation of the value function with regression techniques. In particular, with a multi-layered artificial neural network for non-linear function approximation, one speaks of *deep reinforcement learning*. However, linear function approximators are often sufficient and more amenable to theoretical analysis [1].

[1] Though return is a function of interaction *sequences*, not individual interactions, one can nevertheless efficiently estimate return using so-called *temporal difference learning*, an instance of *stochastic semi-gradient descent*. The details are beyond our scope—see [1].

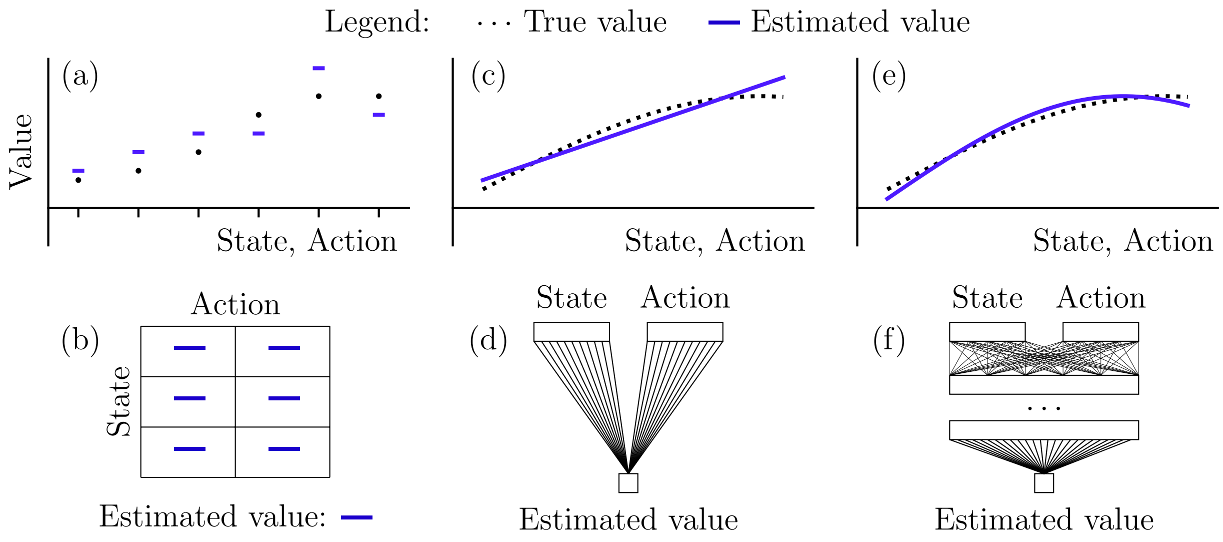

Figure 2: Value function estimation settings: *Tabular* (a, b): estimating finite functions (tables). *Non-tabular* (c–f): for infinite functions, in which one represents states and actions as *vectors* and one learns finitely-parametrised *linear* (c, d) or *non-linear* (e, f) transformations of these.

3. Reinforcement learning with distributions

In recent work, Bellemare et al. propose that algorithms should estimate the conditional probability distribution of the return for each state and action, instead of estimating its expected value [2]. Thus, in so-called *distributional*[2] RL, the traditional *value* function is replaced by a function from states and actions to conditional return *probability distributions*.

Given these return distributions, one may still compute averages for action selection. However, it appears that the estimation process benefits from adopting this richer learning target, unlocking improved performance. To better understand the potential of the distributional paradigm, we discuss these and other uniquely distributional properties below.

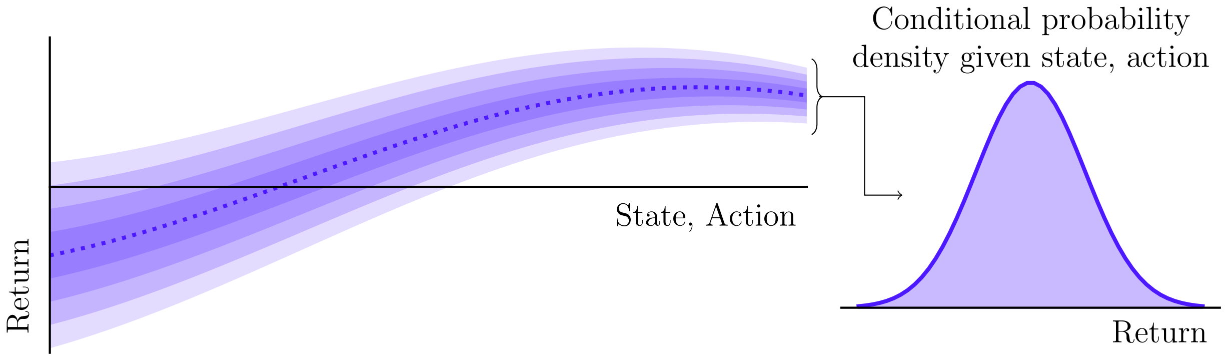

Figure 3: Distributional reinforcement learning: Estimate the *conditional probability distribution* of the return as a function of state/action combinations, rather than its *expected value* (dotted).

[2] Distributional reinforcement learning must not be confused with *distributed* reinforcement learning, wherein one distributes learning algorithm execution over multiple computers. The distribution of concern is rather a *probability* distribution. Of course, one may distribute distributional reinforcement learning [4].

3.1. Distributional performance Early distributional algorithms (viz. [2–9], reviewed in section 4) consistently match and out-perform state-of-the-art value-based algorithms on multiple standard benchmarks. Each benchmark comprises a diverse suite of RL environments. For example, most of these algorithms were evaluated on the challenging and competitive *Atari 2600* benchmark [10], comprising dozens of different arcade games. Moreover, most of these experiments involved large-scale implementations in the complex *deep RL* setting. This broad empirical trend supports the potential of the distributional extension of value-based RL as a promising path towards more effective RL algorithms.

3.2. Distributional analysis Despite the abundant and impressive empirical support for their efficacy, no work has yet provided a principled explanation of the exact mechanism by which distributional algorithms improve over their value-based counterparts. At best, Bellemare et al. propose and *informally* justify some potential mechanisms [2]. However, their proposals are yet to receive even direct empirical investigation.

Complicating matters, further analysis shows that some distributional algorithms behave exactly equivalently to existing value-based algorithms in the tabular and linear function-approximation settings [11]. This leaves one with the historically difficult task of analysing the non-linear function approximation setting (cf. [1]) for a theory of how estimating distributions helps learning at all.

Moreover, it's not obvious that distributional estimation will always stabilise near the true distribution function—one must prove the *convergence* of each distributional algorithm. So far, many tabular distributional algorithms have accompanying convergence proofs. Notably, variations of Bellemare et al.'s original algorithm are convergent in both the tabular [12] and linear function-approximation [13] settings. Current results do not satisfactorily transfer from one algorithm to another, however. A general result on the convergence of distributional algorithms could accelerate exploration.

3.3. Distributional flexibility In work preceding the recent wave of empirical results, Morimura et al. advocate another benefit of learning return distributions: Given distributions, one can readily implement action-selection policies embodying *alternative risk preferences* [14, 15]. We elaborate below.

Pursuing average return 'at all costs' may lead to algorithms accepting rare but ruinous outcomes that are otherwise avoidable (with small concessions in average return). In many practical applications of RL, one seeks to control exposure to this kind of *risk*. Hence, one may prefer actions selected for alternative statistical measures such as *value at risk* or *conditional value at risk* [16]. While such measures may not form viable learning targets alone, they can readily be derived from the distributional information learned by distributional algorithms [14–16]. Therefore, Morimura et al. view distributional RL as a unifying framework for implementing various *risk-sensitive* RL algorithms.

Notably, Morimura et al. eschew formal treatment of the effect of alternative action-selection policies on learning dynamics [14, 15]. Considering the small scale of their experiments, the *algorithmic* safety of this risk-mitigation strategy requires justification. More recent work involving larger-scale experiments [6] echoes this uncertainty, finding unexplained impacts of varying risk preferences. Therefore, this dimension of distributional reinforcement learning should be subject to further empirical and theoretical investigation.

4. Representations for distributional algorithms

We come to the task of representing the distributions central to distributional algorithms. In general, a probability distribution is an infinite object. Therefore, one can only estimate some finite parametrisation of such distributions. There are myriad ways to parametrise and represent distributions. For each representation, there may be several estimation procedures—each a potential distributional RL algorithm. Here we review several recently-proposed representations, seeking promising directions for further exploration.

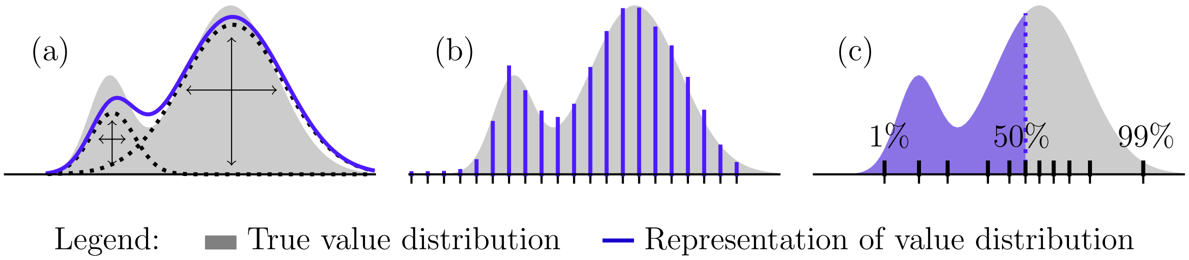

Figure 4: Example distribution representations: (a) Parametric (Gaussian mixture model, 2 components); (b) Categorial (21 discrete categories); (c) Quantiles (11 quantiles, 50%-quantile shown).

4.1. Parametric approaches A simple way to finitely represent distributions is to restrict oneself to *parametrised families* of distributions. For example, the family of Gaussian probability distributions is indexed by two parameters: *Mean* and *variance*. A *parametric* distributional algorithm estimates such parameters as functions of states and actions.

Morimura et al. derive a distributional algorithm for general parametrised families, such as the Gaussian or Laplacian families [15]. More recent works derive distributional algorithms for the expressive family of *mixtures of Gaussians* [4, 8], each validating this approach in large-scale experiments. Currently, these approaches all lack thorough convergence analyses. Moreover, the latter works seem to have under-appreciated the relevance of the former. By combining or comparing their results, it may be possible to reach general conclusions about parametric distributional algorithms.

4.2. Categorical approaches Alternatively, one may finitely approximate a general distribution by bucketing its values into discrete 'categories'. This leads to the *categorical* approach to distributional RL of Bellemare et al. [2], in which one learns a function capturing the probabilities for returns in each of the various categories.

Categorical distributional RL has demonstrated impressive performance [2–4] and yielded more theoretical results than other approaches [11–13]. However, the need to specify categories in advance is a salient limitation since the nature of the return distributions is rarely known—or static—in practice. More advanced algorithms with adaptive categories [9] could be a path forward.

4.3. Quantile-based approaches The p-quantile of a distribution is the value below which the total probability density is p. A suite of p-quantiles captures the approximate shape of a distribution. Indeed, the full set of p-quantiles forms the *quantile function*, or *inverse cumulative distribution function*, perfectly characterising a distribution. Quantile-based distributional algorithms estimate quantile suites, or even the quantile function itself, as a function of states and actions. The results are two of the most empirically impressive distributional algorithms explored to-date [5, 6].

However, it appears these works have not adequately addressed the existing literature on the broader *quantile regression* task, despite its relevance. In particular, *crossing quantile curves* [17, 18] may also plague quantile-based distributional RL algorithms.

4.4. Expectile-based approaches *Expectiles* are summary statistics generalising the expected value in the same way as quantiles generalise the *median* [19, 20]. In particular, a suite of expectiles characterises a distribution, leading to expectile-based distributional RL.

In the first study of expectile-based algorithms, Rowland et al. demonstrate their excellent learning capacity, even compared to quantile-based approaches [7]. Perhaps this is not surprising—the regression literature knows expectiles as a competitive alternative to quantiles [17, 18], even for estimating quantiles themselves [16, 21].

A salient bottleneck in Rowland et al.'s algorithm is the expensive *imputation* step, in which one converts a suite of expectiles into a matching sample. Here, Rowland et al. resort to numerical optimisation. However, more efficient imputation may be achievable using insights from the wider expectile literature [19, 22]. Combined with the intriguing performance results, this highlights expectile-based methods as particularly promising for further investigation.

4.5. On the horizon There are, of course, distributional representations beyond those discussed above. For example, Morimura et al. explores a bespoke algorithm for estimating conditional cumulative distribution functions [14], and Farahmand lays the theoretical groundwork for a distributional algorithm representing return distributions in the frequency domain [23]. The horizon of possible distributional algorithms is yet to be fully explored.

5. Conclusion

Distributional RL is a nascent extension of value-based RL with intriguing potential based on a robust trend of impressive empirical performances. So far, analysis has centred around individual algorithms, particularly Bellemare et al.'s original categorical algorithm and its variations. We advocate for existing and new analyses to be carefully extended towards encompassing the myriad available distributional representations.

Of the many available representations, quantile- and expectile-based methods show particularly strong potential in terms of their performance and representation capacity. A specific direction for future work aiming to realise the potential of the distributional paradigm is to incorporate existing knowledge of quantile and expectile regression into more robust and efficient distributional algorithms. There is also room for wider exploration in the space of distributional representations itself—a single best representation is far from clear.

References

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

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

[3] 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 *AAAI*, 2018.

[4] 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*, 2018.

[5] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in *AAAI*, 2018.

[6] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit quantile networks for distributional reinforcement learning," in *International Conference on Machine Learning*, pp. 1096–1105, 2018.

[7] 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.

[8] 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.

[9] Y. Zhao, P. Liu, C. Bai, W. Zhao, and X. Tang, "Obtaining accurate estimated action values in categorical distributional reinforcement learning," *Knowledge-Based Systems*, p. 105511, 2020.

[10] M. C. Machado, M. G. Bellemare, E. Talvitie, J. Veness, M. J. 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.

[11] C. Lyle, M. G. Bellemare, and P. S. Castro, "A comparative analysis of expected and distributional reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4504–4511, 2019.

[12] M. Rowland, M. Bellemare, W. Dabney, R. Munos, and Y. W. Teh, "An analysis of categorical distributional reinforcement learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 29–37, 2018.

[13] M. G. Bellemare, N. Le Roux, P. S. Castro, and S. Moitra, "Distributional reinforcement learning with linear function approximation," in *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2203–2211, 2019.

[14] T. Morimura, M. Sugiyama, H. Kashima, H. Hachiya, and T. Tanaka, "Nonparametric return distribution approximation for reinforcement learning," in *ICML*, 2010.

[15] T. Morimura, M. Sugiyama, H. Kashima, H. Hachiya, and T. Tanaka, "Parametric return density estimation for reinforcement learning," in *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, pp. 368–375, 2010.

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

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

[18] 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.

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

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

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

[22] 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.

[23] A.-m. Farahmand, "Value function in frequency domain and the characteristic value iteration algorithm," in *Advances in Neural Information Processing Systems*, pp. 14808–14819, 2019.

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.

Command:

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

Output:

```
File: main.tex
Encoding: ascii
Sum count: 2087
Words in text: 1872
Words in headers: 30
Words outside text (captions, etc.): 180
Number of headers: 13
Number of floats/tables/figures: 4
Number of math inlines: 5
Number of math displayed: 0
Subcounts:
  text+headers+captions (#headers/#floats/#inlines/#displayed)
  229+1+0 (1/0/0/0) Section: Introduction
  355+3+98 (1/2/0/0) Section: Value-based reinforcement learning
  99+4+61 (1/1/0/0) Section: Reinforcement learning with distributions
  86+2+0 (1/0/0/0) Subsection: Distributional performance
  181+2+0 (1/0/0/0) Subsection: Distributional analysis
  200+2+0 (1/0/0/0) Subsection: Distributional flexibility
  69+4+21 (1/1/1/0) Section: Representations for distributional algorithms
  123+2+0 (1/0/0/0) Subsection: Parametric approaches
  97+2+0 (1/0/0/0) Subsection: Categorical approaches
  110+2+0 (1/0/4/0) Subsection: Quantile-based approaches
  136+2+0 (1/0/0/0) Subsection: Expectile-based approaches
  54+3+0 (1/0/0/0) Subsection: On the horizon
  133+1+0 (1/0/0/0) Section: Conclusion
```

Total: 2087 words.