Quantifying Uncertainty in Complex Reinforcement Learning Scenarios

Saeid Rezaei, Kenneth N.Brown

Confirm Centre for Smart Manufacturing, School of Computer Science and IT,
University College Cork, Cork, T12XF62, Ireland
saeid.rezaei@cs.ucc.ie , k.brown@cs.ucc.ie

Abstract

Reinforcement learning (RL) has proven itself as a powerful method for solving optimal decision-making problems by interacting with an environment. Particularly, when integrated with neural networks, RL has achieved remarkable successes in complex domains such as board games. we explore different theoretical approaches to understanding the existence of optimal policies under various conditions. The paper culminates in a comparative analysis of methodologies for classifying the complexity of RL problems, including logical, real-world benchmark, and utility theory approaches, providing a comprehensive perspective on the current state and future directions of RL research.

Introduction

Reinforcement learning has demonstrated its capability to identify optimal solutions through interaction with the environment in various applications. When combined with neural networks, notable achievements in board games, as highlighted in (Silver et al. 2016), have underscored the promise of this approach, motivating researchers to develop a variety of RL algorithms tailored to specific problem characteristics. Depending on the nature of the problem, the sets of actions or states can either be discrete or continuous. Algorithms are categorized into different sets based on these characteristics in (Liu et al. 2022):

Algorithm	Input	Output	State-action Spaces support
DQN	states	Q-value	Discrete only
Double DQN	states	Q-value	Discrete only
Dueling DQN	states	Q-value	Discrete only
DDPG	States action pair	Q-value	Continuous only
A2C	States action pair	Q-value	Discrete and Continuous
PPO	States action pair	Q-value	Discrete and Continuous
SAC	States action pair	Q-value	Continuous only
TD3	States action pair	Q-value	Continuous only
MADDPG	States action pair	Q-value	Continuous only

Table 1: RL algorithms

Automated reinforcement learning

Designing reinforcement learning models for a problem and making choices about algorithms and hyperparameters involves a need for thoughtful analysis. It's important to recognize that various configurations can yield vastly different results. AutoRL pipeline(Parker-Holder et al. 2022), a reinforcement learning pipeline has the potential to address numerous tasks while considerably cutting down on time by selecting appropriate configurations. Authors in (Liu et al. 2021) introduced a framework to enhance generalization and shorten training times.

In (Mussi et al. 2023), a general AutoRL pipeline that can be used to solve sequential decision-making problems is introduced. The most important research questions in AutoRL are optimizing hyperparameters with minimal resources and modelling a problem as MDP and generalizing the mapping from available information to an RL environment. In (Mussi et al. 2023) a pipeline for offline and one for online RL is proposed as following:

- **Online Pipeline.** The Online AutoRL Pipeline takes an environment ξ as input, which is modified by Feature Engineering stage to ease learning in the next stage. Using the features created in this stage, it outputs a transformed environment ξ'. Through the Policy Generation process, the environment ξ' is used to estimate the optimal policy $\hat{\pi}^*$. As a result of the Policy Evaluation phase, a performance estimate , $\eta(\hat{\pi}^*)$ is provided based on a performance index η. Feature engineering, Policy generation and Policy evaluation are three main parts in Online AutoRL pipeline.

- **Offline Pipeline.** The Offline AutoRL Pipeline, differently from the online one, two additional preliminary stages are included: Data Generation and Data Preparation. When an environment ξ is provided as input, the Data Generation stage creates a dataset D. When a dataset D is already available, for example from a real process, this step is omitted. As a result, the environment ξ is only used to evaluate the performance of a policy. By applying corrections over each row of the dataset D, the Data Preparation stage modifies the dataset D (i.e., the rows of the dataset) to obtain D' . After passing through the Feature Engineering stage, the environment ξ and dataset D' are transformed into a dataset D'' and an environment ξ' equipped with transformed states, actions, and rewards. In the Policy Generation stage, the dataset D'' is used to estimate the optimal policy $\hat{\pi}^*$. It differs from the online stage in that it uses the dataset D'', whereas the Policy Evaluation stage uses the environment ξ' for estimating $\eta(\hat{\pi}^*)$.

The fundamental definitions and notations in Reinforcement Learning, The diagrams for the online and offline pipelines and definitions for reward shaping and belief states can be found in Appendix A.

Challenges of Real-wprld RL

Despite the impressive achievements of RL and many proposed frameworks, it is highly tuned implementations that fail to generalize many of results. Several challenges need to be addressed to effectively apply reinforcement learning to real-world problems (Dulac-Arnold et al. 2021):

1. learn from limited samples.
2. Dealing with large delays in the observations
3. Learning in high-dimensional state and action spaces.
4. Reasoning about system constraints
5. Interacting with systems that are partially observable
6. Learning from multi-objective or poorly specified reward functions.
7. Being able to provide actions quickly
8. Training off-line from the fixed logs
9. Providing system operators with explainable policies

For each of these challenges, numerous solutions have been proposed, each demonstrating their enhancements over various baselines. Additionally, some articles have attempted to categorize all concepts related to specific aspects of automated reinforcement learning. For example, automated deep reinforcement learning method in (Franke et al. 2021) is supposed to perform an efficient and robust training of any off-policy RL algorithm while simultaneously tuning hyperparameters. In (Eschmann 2021), the study reviews and compares various approaches to reward function design, including reward shaping, the distinction between sparse and dense rewards, intrinsic motivation, curiosity-driven learning, and several other methodologies.

Nevertheless, challenges persist in identifying difficult benchmarks within current reinforcement learning environments. Could classification methods aid in discerning which aspects of reinforcement learning to prioritize based on the problem at hand? The key focus of this article is on exploring diverse approaches to categorizing the complexity of reinforcement learning issues. The structure for the remainder of the document is laid out as follows: Section 2 revisits basic notations and definitions; Section 3 explores the criteria for optimality in various scenarios; Section 4 sheds light on various classification concepts concerning complexity; concluding with a final section that summarizes our findings.

Preliminaries

Uncertainty quantifiation

Based on the structure of algorithms the type of uncertainty is classified for some proposed methods. There are two types of uncertainty which are classified as following (Powell 2022):

Aleatoric uncertainty This class is fine time-scale uncertainty which refers to uncertainty that varies from time-step to time-step which is assumed to reflect the dynamics of the problem. Whether a time step is minutes, hours, days or weeks, fine time-scale uncertainty means that information from one time-step to the next is either uncorrelated, or where correlations drop off quickly.

Epistemic uncertainty All This class is time-scale uncertainty which reflects uncertainty in an environment which occurs over long-time scales. This might reflect new technology, changes in market patterns, the introduction of a new disease, or an unobserved fault in machinery for a process.

Authors in (Franke et al. 2021) presented 14 different algorithms with different structures which are classified based on the type of uncertainty. This classification is summarized in table 2:

Existence of Optimal policy

Within this section, we discuss five theories that demonstrate the presence of an optimal policy in various reinforcement learning modeling scenarios. A concise overview of the characteristics of each theory is presented. Following this, we introduce our own theory, suggesting that under certain conditions, an optimal policy may not exist. The fundamental notations for MDP and POMDP are included in Appendix A and B respectively. Based on these notations, the necessary definitions for the theories are provided as follows:

Optimal value function: A policy π^* is total reward optimal whenever the value function has the highest value for each state in MDP state set or POMDP belief set:

$$v^{\pi^*}(s) \geq v^{\pi}(s) \qquad for\, each\, s\, \in S\, and\, all \quad \pi \in \Pi^{HR}$$

The value function for MDP:
$\mathrm{V}^{\delta}(i) = \sum_{j \in S} T\left(i, \delta(i), j\right) \Big(r\left(i, \delta(i), j\right) \quad + \gamma V^{\delta}(j)\Big)$
And the value function for POMDP is:
$\mathrm{V}^{\delta}(i) = \sum_{j \in S} T(i, \delta(i), j) \left[r(i, \delta(i), j)\right]$
$+ \sum_{o \in O} M(o, j, a) V^{\delta}(B_{a,o})$

Table 2: Uncertainty quantifiation

Base Algorithm	Online/ Offine	Uncertainty Method	Type of Uncertainty
Distributional DQN	Online	MC-Dropout and Variance Networks	Epistemic and Aleatoric
Model Based DDPG	Online	Bootstrapped Q	Epistemic
DQN	Online	Bootstrapped Q and Prior Networks	Epistemic
QR-DQN	Online	Bootstrapped Q	Epistemic and Aleatoric
SAC	Offine	Variance of Dynamics Model	Epistemic
DQN	Online	Bootstrapped Q	Epistemic
SAC Actor-Critic	Offine Offine	Bootstrapped Q MC-Dropout	Epistemic Epistemic
SAC	Online	MC-Dropout and Boot-strapped Q	Epistemic
SAC	Online	Bootstrapped Variance Q Networks	Epistemic and Aleatoric
SAC + CQL	Offine	Bootstrapped Q and Policy	Epistemic
SAC	Offine	Bootstrapped Q	Epistemic
Actor-Critic	Offine	Bootstrapped Q	Epistemic
PPO	Online	Variance Networks	Aleatoric

Table 3: uncertainty quantification

Theory 1

Given a MDP (S, A, T, R) and a stationary policy δ, the sequence V_t generated by

$$V_{t+1}(i) = V_t(i) + a_t(i)z_t(i)d_t$$

converges w.p.1 to the value function V^{δ} along an infinite trajectory under policy δ, as long as every state is visited an infinite number of times, the step-sizes α_t verify:

$$\sum_{t=0}^{\infty} \alpha_t = \infty \qquad \sum_{t=0}^{\infty} {\alpha_t}^2 < \infty$$

To prove it is assumed that all states are visited infinitely often, the Markov chain (S, P_{δ}) is ergodic. (S, P_{δ}) represents the Markov chain with state-space S and transition probabilities. (Melo, Ribeiro, and Norte 2005)

$$P_{\delta}(i, j) = T\left(i, \delta(i), j\right),$$

for $i, j \ \in S$.

Theory 2

Given a MDP (S, A, T, R) , the sequence Q_k, generated by

$$Q_{t+1}(i, a) = Q_t(i, a) + a_t(i, a)(R_t + \gamma \max_{b \epsilon A} Q_t(j, b) - Q_t(i, a))$$

converges w.p.1 to the Q-function Q^* along an infinite trajectory under policy δ, as long as every state-action pair is tried an infinite number of times and step-sizes α_t verify (Melo, Ribeiro, and Norte 2005):

$$\sum_{t=0}^{\infty} \alpha_t = \infty \qquad \sum_{t=0}^{\infty} {\alpha_t}^2 < \infty$$

The purpose of this algorithm is to compute the Q-functions given by:

$$Q^*(i, a) = \sum_{j \in S} T(i, a, j)(r(i, a, j) + \gamma V^*(j))$$

Theory 3

Consider a POMDP $(S, A,\ O,\ T, M\ , R)$ and suppose that the underlying Markov chain is ergodic for any stationary policy. If the stationary policy δ used to generate the learning trajectory assigns for non-zero probability to all actions and step-sizes α_t verify

$$\sum_{t=0}^{\infty} \alpha_t = \infty \qquad \sum_{t=0}^{\infty} {\alpha_t}^2 < \infty$$

Algorithm SQ-Learning
(The procedure for SQ-Learning should be fixed)
then the SQ-Learning algorithm converges to the optimal approximation of Q^* given the sample set $\{\widetilde{\pi}_1, \dots\ ,\ \widetilde{\pi}_N\ \}$(Melo, Ribeiro, and Norte 2005).

Theory 4

This theory uses the following definitions:

History-based Decision Process (HDP) A history-based decision process P is a stochastic mapping from a history-action pair to observation-reward pairs. Formally, $P \ \ : H^* \ \times A \ \rightarrow O \ \times R$ where denotes a stochastic mapping. We use Q to denote an action-value function of a HDP, and

Q^* denotes the optimal Q-value function.

$$Q^*(h,a) \ \sum_{\acute{o}\acute{r}} P\left(\acute{o}\acute{r}|ha\right)(\acute{r} + \ \gamma \max_{\breve{a}} Q^*\left(ha\acute{o}\acute{r}, \breve{a}\right))$$

State-uniformity condition: For any action a, if any two histories h and $\acute{h}$ map to the same state s, then the optimal Q-values of the underlying HDP of these histories are the same, i.e., state-uniform; $Q^*(h,a) = Q^*\left(\acute{h}, a\right)$. It is easy to see that in this case $q^*(s,a) = Q^*(h,a)$. The state-uniformity condition is weaker than the MDP condition.

Q-Value Uniform Decision Process (QDP): A model is a QDP if it satisfies the state-uniformity condition.

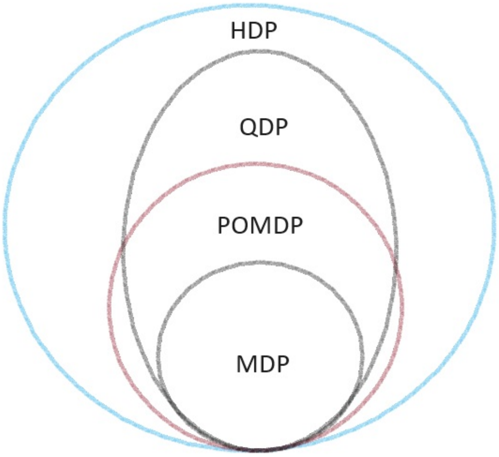

Figure – QDP in the perspective of other decision problem classes

Convergence in QDP: Assuming the state process is ergodic and QDP, the agent learns the optimal action-value function of a QDP state-process(Majeed and Hutter 2021).

Theory 5

Given the following assumptions, an optimal policy does not exist.

1-The MDP' state or POMDP' belief state is fully connected

2- The reward random variable varies across different period $t < N$ Let $R_t \equiv r_t(X_t, Y_t$ denote the random reward received in period $t < N$, $R_N \equiv r_N(X_N)$ denote the terminal reward, $R \equiv (R_1, R_2, \ldots, R_N)$ denote a random sequence of rewards, and R the set of all possible reward sequences.)

With this assumption, we would say that the values are anticipated to fluctuate over the next n period.

$$\exists\, m, n\ , o \forall\, T > m \quad :$$

The expected value of reward for each state (or belief state) will satisfy the following condition:

$E\left(R(s)_{T+n}\right) = \ r\ , \ \ r \ \in [R_{\min}\ , \ R_{\max}]$, based on this condition it is possible to assume that:

$$E\left(R(s)_{T+o}\right) > E\left(R(s)_{T+n}\right)$$

and

$$E\left(R(s)_{T+n}\right) > E\left(R(s)_{T+m}\right)$$

Proof. The main idea regarding the proof is as following: Based on the assumptions the expected value of the reward value, represented as a random variable, changes across different timesteps. Referring back to the definition of an optimal policy presented in Chapter 2, the value function for each state should be maximized for it to align with the optimal value function.Based on the assumption about the fully connected state set, a contradiction arises between the existence of an optimal policy and the foundational assumption regarding the class features.

consider the following:

Optimal Policy:

$$V^\delta \quad : \quad S \rightarrow R$$

$$V^\delta(i) = E_\delta[\sum_{k=0}^{\infty} \gamma^k \quad R_{t+k}|\, \phi_t = i]$$

where R_{t+k} is the reward received at time $t+k$, given by

$$R_{t+k} = r(\phi_{t+k}, \delta(\phi_{t+k}), \phi_{t+k+1})$$

and E_δ is the expected value operator associated with the transition probabilities defined by policy δ.

$\mathrm{V}^\delta(i) = E\left[R\left(i, \delta(i)\right)\right] + \gamma \sum_{k=0}^{\infty} \sum_{j \in S} T\left(i, \delta(i), j\right)$
$\times\, E_\delta\left[R_{t+k+1}|\phi_{t+1} = j\right]$

$\mathrm{V}^\delta(i) = E\left[R\left(i, \delta(i)\right)\right]$
$+\, \gamma \sum_{k=0}^{\infty} \sum_{j \in S} T\left(i, \delta(i), j\right) E_\delta\left[R_{t+k+1}|\, \phi_{t+1} = j\right]$

$V^\delta(i) = E\left[R\left(i, \delta(i)\right)\right]$
$+\, T\left(i, \delta(i), j\right) \gamma V^\delta(j)$

$\mathrm{V}^\delta(i) = E\left[R\left(i, \delta(i)\right)\right]$
$+\, T\left(i, \delta(i), j\right) \gamma V^\delta(j)$

That is, we seek a policy π^* for which.

$$v_N^{\pi^*}(s) \geq v_N^\pi(s), \ s \ \in S$$

Based on the definition of optimal policy the following equation should be satisfied for this policy:

$$V^\delta(s) \geq V^\pi(i), \ s \ \in S$$

$$E\left[R\left(i, \ \delta(i)\right)\right] + T\left(i, \ \delta(i), j\right) \gamma V^\delta(j) \geq$$

$$E\left[R\left(i, \ \pi(i)\right)\right] + T\left(i, \ \pi(i), j\right) \gamma V^\pi(j)$$

Based on the problem assumption regarding the conditions there exist policy which:

$E\left[R\left(i, \ \delta(i)\right)\right] \ < \ E\left[R\left(i, \ \pi(i)\right)\right]$ and $T\left(i, \ \ \delta(i), j\right) \ = T\left(i, \ \pi(i), j\right)$

Then:

$$T\left(i, \ \delta(i), j\right) \gamma V^\delta(j) < T\left(i, \ \pi(i), j\right) \gamma V^\pi(j)$$

Then $V^\delta(j) < V^\pi(j)$ which is a contradiction with Definition of optimal policy for state j.

Theory 6

Given the assumptions in Theory 5, ε optimal policy does not exist. Proof. The main idea regarding the proof is as following: Based on the assumptions the expected value of the reward value, and optimal policy and using Markov's inequality , a contradiction arises between the existence of an ε optimal policy and the foundational assumption regarding the class features.

Markov's inequality:

If X is a non-negative random variable, then for all $\alpha > 0$, which is called Markov inequality:

$$P(X \geq a) \leq \frac{E[X]}{a}$$

$$1 - P(X \geq a) \geq 1 - \frac{E[X]}{a}$$

$$P(X < a) \geq 1 - \frac{E[X]}{a}$$

Based on the definition of ε optimal policy, considering δ as an optimal policy then there exists a policy:

$$V^{\delta}(j) - V^{\pi}(j) < \varepsilon$$

Let us define the random variable.

X = $\left|V^{\delta}(j) - V^{\pi}(j)\right|$

$$= \Big| E\left[R\left(i, \delta(i)\right)\right] + T\left(i, \delta(i), j\right)\gamma V^{\delta}(j)$$

$$- E\left[R\left(i, \pi(i)\right)\right] - T\left(i, \pi(i), j\right)\gamma V^{\pi}(j)\Big|$$

For state j: Based on the existence of same transition function for two policy it is possible to assume $V^{\delta}(j) = V^{\pi}(j)$ .Then:

$$M = E\left[R\left(i,\ \delta(i)\right)\right] + T\left(i,\ \delta(i), j\right)\gamma V^{\delta}(j)$$

$$N = E\left[R\left(i,\ \pi(i)\right)\right] - T\left(i,\ \pi(i), j\right)\gamma V^{\pi}(j)$$

$$P\left(|M - N| < a\right)$$

$$\leq 1 - \frac{E[E\left[R\left(i,\ \delta(i)\right)\right] - E\left[R\left(i,\ \pi(i)\right)\right]]}{a}$$

Based on the Theory assumption there exist a policy :

$$E\left[R\left(i,\ \delta(i)\right)\right] - E\left[R\left(i,\ \pi(i)\right)\right] > a$$

Then: $p\left(\left|V^{\delta}(j) - V^{\pi}(j)\right|\right) < 0$ which is a contradiction with the assumption and probability axioms.

The distinctions and commonalities among Theories 1 through 6 are encapsulated in the table 4:

Approach to Complexity Classification

Various methodologies can be employed to categorize the complexity levels within different domains. This section encapsulates these classification strategies, comparing their distinctive features and applications.

Logical approach

The concept of reward serves as the cornerstone for reinforcement learning agents in their quest to learn optimal policies. In their work, Abel et al. (2021) discuss the expressivity of rewards as a mechanism to define the tasks an agent should undertake. Similarly, Dulac-Arnold et al. (2021) explore the application of Multi-Objective Reward Functions to address real-world challenges, underlining the complexity of catering to multiple goals simultaneously. This perspective is further supported by research, including Silver et al. (2021), who propose that intelligence and its related capabilities can essentially be seen as the pursuit of reward maximization within an environment. This notion suggests that navigating complex environments requires an array of skills aimed at optimizing rewards.

Contrarily, Vamplew et al. (2022) challenge this view by highlighting the importance of considering trade-offs between multiple conflicting objectives for intelligent decision-making. They argue that an agent motivated solely by scalar rewards may need to focus on optimizing one goal at the expense of others or attempt to balance them through a scalar combination. This debate centers on the ”Reward-is-Enough” hypothesis, positing that an effective reward-maximizing agent could inherently develop intelligence-related abilities by striving to achieve its objectives, potentially using unknown algorithms to excel in maximizing cumulative rewards in its environment. Despite this, Vamplew et al. (2022) also acknowledge limitations, noting that intelligent decision-making often necessitates navigating between competing objectives, thus questioning the sufficiency of scalar rewards in capturing the complexity of real-world decision-making.

Real-world benchmarks approach

Some frameworks tried to address some of these challenges. For example, automated deep reinforcement learning method in (Franke et al. 2021) is supposed to perform an efficient and robust training of any off-policy RL algorithm while simultaneously tuning hyperparameters. However, there should be some benchmarks which are like real-world settings to evaluate these frameworks and be able to compare performance of different methods. One example is the three levels of complexity which are proposed based on the Deepmind control suit in (Dulac-Arnold et al. 2021). While it is easy to show the convergence of reinforcement learning algorithms in easy and medium level of complexity, convergence of algorithms in hard level is a difficult task which needs to be demonstrated. Authors in (Taitler et al. 2023) used pyRDDLGym framework to present benchmark for twelve different domains. These benchmarks have different complexity with different parameters and RL algorithms may have poor performance in some settings. However, there should be some bench-marks which are like real-world settings to evaluate these frameworks and be able to compare performance of different methods. One example is the three levels of complexity which are proposed based on the Deepmind control suit in (Dulac-Arnold et al. 2021). While it is easy to show the convergence of reinforcement

Feature	invariant MDP	Fully connected	Non stationary	$r_t' s$ vary across	$p_t' s$ vary across	Additional Conditions
Theory 1	×	×	×	×	×	Operator conditions
Theory 2	×	×	×	×	×	Operator conditions
Theory 3	×	×	×	×	×	Operator conditions
Theory 4	×	×	×	×	×	QDP condition
Theory 5,6	+	+	+	+	+	Reward condition

Table 4: Summary of Theoretical Conditions

learning algorithms in easy and medium level of complexity, convergence of algorithms in hard level is a difficult task which needs to be demonstrated. Authors in (Taitler et al. 2023) used pyRDDLGym framework to present benchmark for twelve different domains. These benchmarks have different complexity with different parameters and RL algorithms may have poor performance in some settings.

Utility theory approach

The foundational idea of reward in reinforcement learning is succinctly captured by Sutton's reward hypothesis, which posits that ”all of what we mean by goals and purposes can be effectively conceptualized as the maximization of the expected value of the cumulative sum of a received scalar signal (reward).” This article further explores the domain of reward signals through the lens of utility theory, providing a structured analysis of these concepts.

Von Neumann-Morgenstern (VNM) Axioms

The discourse on utility introduces several key axioms, essential for understanding decision-making under uncertainty:

Axiom of Completeness: For any two lotteries $L, M \in \mathcal{L}$, it holds that either $L \succeq M$ or $M \succeq L$, ensuring that any pair of outcomes is comparable.

Axiom of Transitivity: For any lotteries $L, M, N \in \mathcal{L}$, if $L \succeq M$ and $M \succeq N$, then $L \succeq N$, establishing a consistent preference order.

Axiom of Continuity: Given lotteries $L \succeq M \succeq N$, there exists a probability $p \in [0, 1]$ such that a mix of L and N with probability p is equivalent in preference to M.

Axiom of Independence: For any lotteries $L, M, N \in \mathcal{L}$ and a probability $p \in [0, 1]$, preference between L and M remains unchanged even when mixed with N in the same proportion.

These axioms enable the definition of a **Utility Function**, a real-valued function $u : \mathcal{L} \to R$, mapping lotteries to utilities such that $L \succeq M \Leftrightarrow u(L) \geq u(M)$.

Von Neumann-Morgenstern Utility Theorem: This theorem asserts that a preference relation adheres to the VNM axioms if and only if it can be represented by a utility function, where the utility of any lottery equals the expected utility of its outcomes. This utility function is uniquely determined up to a positive affine transformation.(Bowling et al. 2023)

Controlled Markov Process (CMP): This concept models the interaction between an agent and the environment, where the outcome of each action is determined by the current state and the action chosen, following the Markov property. The CMP is defined by a tuple (S, A, P, T), encapsulating states, actions, transition probabilities, and termination probabilities.

Extension to Sequential Decision Making: The VNM utility theorem extends these principles to the realm of sequential decision-making in a CMP, illustrating how preferences over finite trajectories can be represented through rewards and reward multipliers. This forms the basis for identifying an optimal, memoryless policy, highlighting the utility theory's applicability to reinforcement learning scenarios.(Shakerinava and Ravanbakhsh 2022)

Comparative Analysis of Various Approaches

We can now contrast the three traditional methodologies with the strategy rooted in optimality criteria. The differences and similarities among these approaches are concisely summarized in the table 5:

Conclusion

This paper has traversed the landscape of reinforcement learning, from its algorithmic foundations to the forefront of automated reinforcement learning techniques. Through a systematic review of theories surrounding optimal policy existence, we provided insights into the conditions under which optimal policies can be ascertained or shown to be elusive. Our exploration underscored the significance of complexity classification in reinforcement learning, presenting a variety of approaches, including logical reasoning, utility theory, and real-world benchmarks. Each approach offers unique perspectives on dissecting the intricacies of RL problems, suggesting that a blend of theoretical and practical considerations is essential for advancing RL applications.

Acknowledgments

This research was supported by a grant from Science Foundation Ireland under Grant number 16/RC/3918 which is cofunded under the European Regional Development Fund.

References

Agarwal, M.; and Aggarwal, V. 2021. Blind Decision Making: Reinforcement Learning with Delayed Observations. *Pattern Recognition Letters*, 150: 176–182.

Bowling, M.; Martin, J. D.; Abel, D.; and Dabney, W. 2023. Settling the Reward Hypothesis. In *International Conference on Machine Learning*, 3003–3020. PMLR.

level	Benchmark Approach	Utility theory	MDP and POMDP features
Easy	Easy setting	VNM axioms	Existence of optimal policy is proved (Theory1-4)
Medium	Medium setting	VNM+ axioms	Existence of optimal policy is unknown
Hard	Hard setting		Theory 5,6

Table 5: Complexity approach

Dulac-Arnold, G.; Levine, N.; Mankowitz, D. J.; Li, J.; Paduraru, C.; Gowal, S.; and Hester, T. 2021. An Empirical Investigation of the Challenges of Real-World Reinforcement Learning. arXiv:2003.11881.

Eschmann, J. 2021. *Reward Function Design in Reinforcement Learning*, 25–33. Cham: Springer International Publishing. ISBN 978-3-030-41188-6.

Franke, J. K. H.; Köhler, G.; Biedenkapp, A.; and Hutter, F. 2021. Sample-Efficient Automated Deep Reinforcement Learning. arXiv:2009.01555.

Gangwani, T.; Zhou, Y.; and Peng, J. 2020. Learning Guidance Rewards with Trajectory-Space Smoothing. *Advances in Neural Information Processing Systems*, 33: 822–832.

Han, B.; Ren, Z.; Wu, Z.; Zhou, Y.; and Peng, J. 2022. Off-Policy Reinforcement Learning with Delayed Rewards. In *International Conference on Machine Learning*, 8280–8303. PMLR.

Icarte, R. T.; Klassen, T. Q.; Valenzano, R.; and McIlraith, S. A. 2022. Reward Machines: Exploiting Reward Function Structure in Reinforcement Learning. *Journal of Artificial Intelligence Research*, 73: 173–208.

Igl, M.; Zintgraf, L.; Le, T. A.; Wood, F.; and Whiteson, S. 2018. Deep Variational Reinforcement Learning for POMDPs. In *International Conference on Machine Learning*, 2117–2126. PMLR.

Liu, K.; Fu, Y.; Wu, L.; Li, X.; Aggarwal, C.; and Xiong, H. 2021. Automated feature selection: A reinforcement learning perspective. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, X.-Y.; Yang, H.; Chen, Q.; Zhang, R.; Yang, L.; Xiao, B.; and Wang, C. D. 2022. FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance. arXiv:2011.09607.

Majeed, S. J.; and Hutter, M. 2021. Reducing Planning Complexity of General Reinforcement Learning with Non-Markovian Abstractions. arXiv:2112.13386.

Melo, F. A.; Ribeiro, M. I.; and Norte, I. T. 2005. Convergence Results for Reinforcement Learning with Partial Observability. *Institute for Systems and Robotics, Tech. Rep. RT-602-05*.

Mussi, M.; Lombarda, D.; Metelli, A. M.; Trovò, F.; and Restelli, M. 2023. ARLO: A framework for Automated Reinforcement Learning. *Expert Systems with Applications*, 224: 119883.

Parker-Holder, J.; Rajan, R.; Song, X.; Biedenkapp, A.; Miao, Y.; Eimer, T.; Zhang, B.; Nguyen, V.; Calandra, R.; Faust, A.; et al. 2022. Automated reinforcement learning (autorl): A survey and open problems. *Journal of Artificial Intelligence Research*, 74: 517–568.

Powell, W. B. 2022. *Reinforcement Learning and Stochastic Optimization: A unified framework for sequential decisions*. John Wiley & Sons.

Shakerinava, M.; and Ravanbakhsh, S. 2022. Utility Theory for Sequential Decision Making. In *International Conference on Machine Learning*, 19616–19625. PMLR.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Taitler, A.; Gimelfarb, M.; Jeong, J.; Gopalakrishnan, S.; Mladenov, M.; Liu, X.; and Sanner, S. 2023. pyRDDLGym: From RDDL to Gym Environments. arXiv:2211.05939.

Wulfe, B.; Balakrishna, A.; Ellis, L.; Mercat, J.; McAllister, R.; and Gaidon, A. 2022. Dynamics-Aware Comparison of Learned Reward Functions. arXiv:2201.10081.

Yin, H.; Chen, J.; Pan, S. J.; and Tschiatschek, S. 2021. Sequential Generative Exploration Model for Partially Observable Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 10700–10708.

Appendix A

Below is a table summarizing the fundamental definitions in Reinforcement Learning (RL):

Table 6: fundamental definitions in Reinforcement Learning

Notation	**Definition**
A	Set of actions
S	Set of states
A_s	Set of actions in state s
T	A mapping from $S \times A$ into distributions over the states in S
R	A reward function that maps from S to real-valued rewards
Π	Set of policies $\{\pi_1, \pi_2, ..., \pi_N\}$
S_t	state at time t
A_t	action at time t
R_t	reward at time t
γ	discount rate (where $0 \leq \gamma \leq 1$)
I	Initial state of MDP
policy π	A mapping from S to actions in A
Value Function V	Maps from elements of S to real values
$r_t(s, a)$	Reward of choosing action $a \in A_s$ in state s at decision period t
$p(\hat{s}, r \mid s, a)$	Probability of next state $\hat{s}$ and reward r, given current state s and current action a
$r_t(s, a)$	Reward received in period t when action $a \in A_s$ is selected in $s \in S$
$r_t(s, a, j)$	Reward received in period t when action $a \in A_s$ is selected in $s \in S$ and next state is j
expected value at decision period t	$r_t(s, a) = \sum_{j \in S} r_t(s, a, j) p_t (j \vert s, a)$
$v_N^\pi(s)$	Value of Value function for state s in decision period N

The diagrams for online and offline reinforcement learning are presented as follows:

The definitions for reward shaping and belief state are provided as follows:

Reward Shaping

The core task of Feature Engineering is to select and generate a set of features that properly model the state-action space of the problem and perform reward shaping actions to facilitate the following learning phase. Reward shaping is a structured approach to develop a policy for a given Markov Decision Process (MDP) $M = (S, A, r, P, \lambda)$. To facilitate the learning process of optimal policy. The reward function $R = R + F$ in the transformed MDP $M = (S, A, r, P, \lambda)$ where $F : S \times A \times S \to R$ is a bounded real-valued function called the shaping reward function. So if in the original MDP M we would have received reward $R(S, A, S')$ for transitioning from s to S' on action a then in the new MDP M' we would receive reward $R(S, A, S') + F(S, A, S')$ on the same event. ()ng1999policy Improving the policy training performance in sparse rewarded POMDP problems is one popular example of reward shaping. SGEM (Yin et al. 2021), PIDR-MDP (Han et al. 2022), surrogate RL objective with smoothing in the trajectory-space(Gangwani, Zhou, and Peng 2020) are three examples which tried to model the sequential decision problem with delayed rewards with the guidance of shaping reward function instead of the environmental reward in different benchmarks. For some applications, proposed reward functions should be checked by reliable tools to make sure system's objectives are aligned with the human intentions (Wulfe et al. 2022). Inverse reinforcement learning (IRL) involves deducing the underlying reward function from the behavior of an expert. IRL's goal is to ascertain the reward function that a domain expert is maximizing, using data from their behavior and the environmental model. This method has potential applications in several fields related to reinforcement learning, including the study of animal and human behaviors, econometrics, and the development of intelligent agents. Most of the current inverse reinforcement learning (IRL) algorithms operate under the assumption that the environment follows a Markov decision process (MDP) model. However, there is a growing interest in adapting these algorithms for partially observable situations, as this would allow for more accurate representation of real-world scenarioschoi2011inverse. Reward Machines (RMs) offer an organized, automata-based framework for representing a reward function. This structure allows a Reinforcement Learning (RL) agent to break down an RL problem into well-defined subproblems. These subproblems can then be effectively tackled through off-policy learning techniques(Icarte et al. 2022).

Belief states

Diverse methods can be employed to revise the belief in various POMDP which is an important part of the defining state-action space in Feature Engineering. (Igl et al. 2018) proposed a method to update a belief distribution for problems which have incomplete and noisy observations by learning a generative model which is used to update a belief distribution. (Agarwal and Aggarwal 2021) focused on the issue of delayed observation updates in a reinforcement learning agent, where the agent does not instantly receive the current state of the environment. They demonstrated that the anticipated immediate rewards produced in an MDP (Markov Decision Process) with delays are equivalent to those generated in a corresponding extended MDP without such delays.

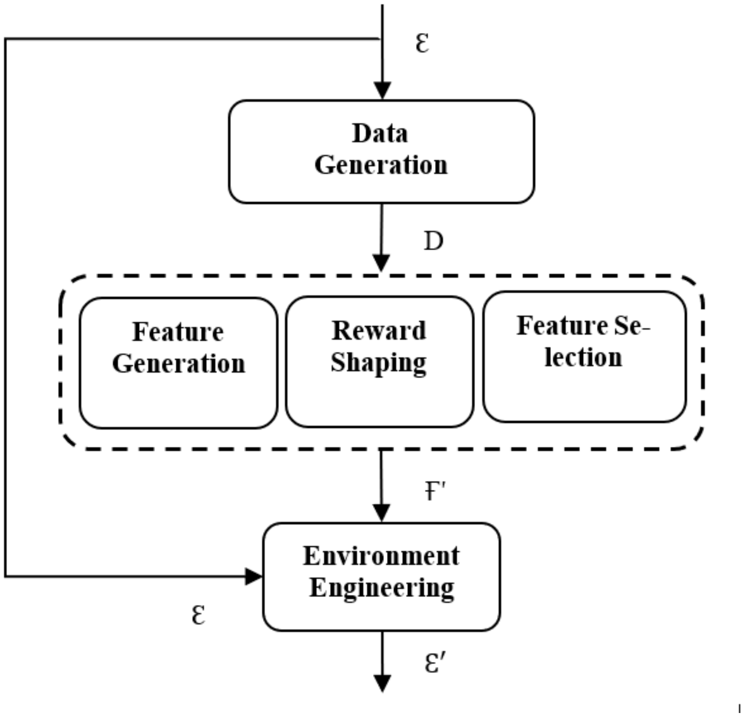

Figure 1: Feature Engineering stage in Online Pipeline

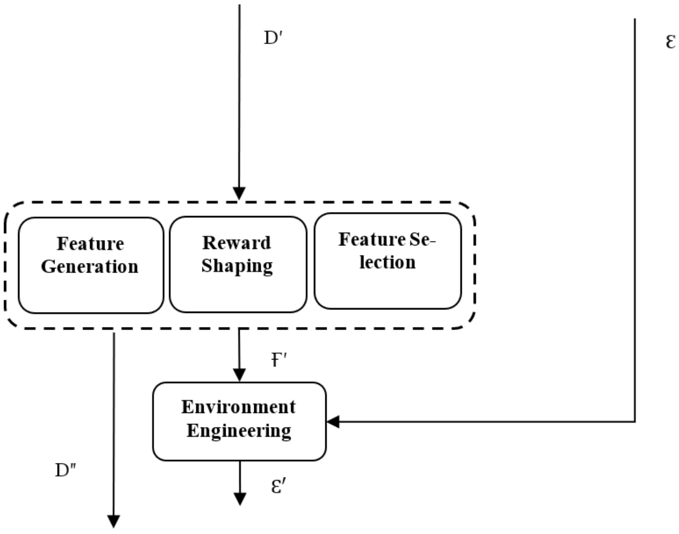

Figure 2: Feature Engineering stage in offline Pipeline