School of Computing
FACULTY OF ENGINEERING AND
PHYSICAL SCIENCES

UNIVERSITY OF LEEDS

Final Report

Planning to Explore in Reinforcement Learning

Jared William Swift

Submitted in accordance with the requirements for the degree of BSc Computer Science with Artificial Intelligence (Ind)

2022/23

COMP3931 Individual Project

The candidate confirms that the following have been submitted.

Items	Format	Recipient(s) and Date
Final Report	PDF file	Uploaded to Minerva (DD/MM/YY)
<Example> Scanned participant consent forms	PDF file / file archive	Uploaded to Minerva (DD/MM/YY)
<Example> Link to online code repository	URL	Sent to supervisor and assessor (DD/MM/YY)
<Example> User manuals	PDF file	Sent to client and supervisor (DD/MM/YY)

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of Student) ______________________

Summary

<Concise statement of the problem you intended to solve and main achievements (no more than one A4 page)>

Acknowledgements

<The page should contain any acknowledgements to those who have assisted with your work. Where you have worked as part of a team, you should, where appropriate, reference to any contribution made by other to the project.>

Note that it is not acceptable to solicit assistance on 'proof reading' which is defined as the "the systematic checking and identification of errors in spelling, punctuation, grammar and sentence construction, formatting and layout in the test"; see
`https://www.leeds.ac.uk/secretariat/documents/proof_reading_policy.pdf`

Contents

Chapter 1

Introduction and Background Research

1.1 Introduction

Reinforcement Learning (RL) [1] is based on the concept of learning through experience; through trial-and-error. An agent learns how to behave in an environment by interacting with it and receiving reinforcement (both positive and negative) through a numerical signal called the reward. RL is akin to human and animal learning, and comes from the field of Psychology through the studies of operant conditioning [2], which have shown that human and animal behaviour can be shaped through positive and negative reinforcement.

Experience can only be gained by "trying new things", however it is infeasible to "try everything", especially in physical tasks of practical interest. Thus, there is a need for "exploration"; this is when the agent "tries new things" and learns the consequences of its actions. Exploration is a widely studied topic in RL, as it is necessary for learning. Although exploration methods based on optimism, such as UCRL [3] and Fitted R-Max [4], intrinsic motivation, such as the use of Competence Maps [5], and Bayesian Methods, such as Bayesian Q-Learning [6], random exploration is ubiquitous in practice. Randomness precludes efficiency, as sub-optimal actions may be continually evaluated even if they have been realised to be sub-optimal. Randomness also prohibits explainability, and does not imply intelligence.

Automated Planning (or just Planning) uses embedded knowledge of the environment, in the form of a model, to determine the optimal sequence of successive actions to fulfil a goal [7, 8]. However, planning relies on the model to accurately represent the environment; which cannot be guaranteed, due to approximations and abstractions, and therefore planning can be quite fragile. A clear distinction between Planning and RL is that Planning relies on previously obtained knowledge, whereas RL relies on obtaining knowledge through experience.

Although Planning and RL take different approaches to decision making, they may be combined, which is known as model-based RL - which has been show to be very successful in recent years ([9], [10]). Model-based RL can be explicit, where an agent plans over a learned model, or implicit where planning and learning is more tightly coupled [11], throughout this work we focus on the latter.

Forms of model-based RL have been developed that overcome the inherent inaccuracies of models and reduce the exploration required for learning by using the planner to constrain and inform exploration, such as DARLING [12]; which is a big inspiration for this work.

Within this work we explore the development of a framework that synergises planning and learning in order to drive and constrain exploration by making intelligent hypotheses about the environment, informed by the inherently inaccurate, but still useful, model, previous experience and environmental observations, rather than through randomness. The ultimate goal of this work is to mitigate the effect of the inherent inaccuracies in the model on the quality of learned behaviour; resulting in agents that can learn beyond the inaccuracies of the model, through intelligent exploration.

1.2 Reinforcement Learning

Reinforcement Learning (RL) does not fall into either of the traditional machine learning paradigms (supervised and unsupervised learning) - it is a machine learning paradigm of its own. A RL problem comes in the form of a sequential-decision making problem, and is formalised through a Markov Decision Process (MDP). Within an RL problem a goal-directed decision-making agent learns how to behave in an environment, which may be stochastic. The agent learns by interacting with the environment through actions and observing the affects through its new state and a numerical reward signal. The goal of the agent is to learn how to map states to actions in order to maximise the cumulative long-term reward signal [1]. The behaviour that the agent learns is known as the **policy**. The **reward function** indicates the immediate value of state-action pair, the goal of the agent is to maximise the cumulative returns from this. The **value function** indicates the expected cumulative reward the agent can receive starting from a given state.

Within this work, we split RL into two categories: model-free and model-based. Model-free RL is the traditional instantiation of RL - the agent learns to act in an environment, with no knowledge of its dynamics. We briefly mentioned model-based RL within the introduction, but didn't explain what it actually is. Model-based RL is where the agent learns to act in an environment, and has some understanding of the dynamics of the environment in the form of a model. By the nature of models, the model is inaccurate, more often than not.

1.2.1 Markov Decision Processes

The Markov Property states that the future is conditionally independent of the past given the present. An RL problem that satisfies the Markov Property is known as a Markov Decision Problem, and can be modelled by a Markov Decision Process (MDP). Where an agent is able to fully observe its state, the problem can be modelled as an MDP. Conversely, where an agent can only partially observe its state, the problem can be modelled by a Partially Observable Markov Decision Process (POMDP). Consider an agent interacting with a stochastic gridworld, the agent can easily observe its state, whereas a robot navigating a maze may not be able to observe its exact state, due to uncertainty in sensors, joint readings, etc. In fact, the real-world is a POMDP. Within this work, as a simplification, we assume that the agent is fully able to observe its state. Furthermore that the environment can be discretised (rather than being modelled in a continuous nature); hence we consider **finite** MDPs. A finite MDP is a 4-tuple: MDP $= (S, A, T, R)$ where:

- S is a finite set of states.
- A is a finite set of actions.
- $T : S \times A \times S \to [0, 1]$ is the transition function, which determines the probability of transitioning from a state $s \in S$ to $s' \in S$ with an action $a \in A$.
- $R : S \times A \times S \to \mathbb{R}$ is the reward function, which determines the reward signal, $r \in \mathbb{R}$ received by the agent from transitioning from a state $s \in S$ $s' \in S$ with the action $a \in A$. This reward is extrinsic to the agent; it comes from the environment.

- γ is the discount factor.

The transition function, T is a key indicator about the nature of the environment. If $\forall s, s' \in S, \forall a \in A$, $T(s, a,' s) \in \{0, 1\}$, then the environment is deterministic, otherwise it is probabilistic. A deterministic environment is one which there is no variance in the outcomes of action in a given state; taking the same action in the same state always produces the same outcome. Whereas a probabilistic (or non-deterministic) environment has uncertainty associated with transitions.

The Bellman Equation determines the expected reward for being in a state $s \in S$ and following a fixed policy π:

$$V^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} P(s'|s, \pi(s))V^\pi(s')$$

Where V^π is the value function of the policy, π, $0 \leq \gamma \leq 1$ is the discount factor. The Bellman Optimality equation determines the reward for taking the action giving the highest expected return.

$$V^{\pi*}(s) = \text{argmax}_a R(s, a) + \gamma \sum_{s'} P(s'|s, a)V^{\pi*}(s')$$

Where $V^{\pi*}$ is the value function of the optimal policy, $\pi*$, $0 \leq \gamma \leq 1$ is the discount factor.

1.2.2 Dynamic Programming

- Given a perfect model of the environment, embedded in a MDP, an optimal policy can be computed using Dynamic Programming. However, this assumption of a perfect model is flawed.

1.2.2.1 Policy Iteration

1.2.2.2 Value Iteration

1.2.3 Temporal Difference Learning

The (temporal) credit assignment problem [13] is the problem of determining which actions led to an outcome, and assigning credit among them; it's often the case that a sequence of actions led to an outcome, rather than a single action. In the context of RL, temporal credit assignment is important because in order to maximise the cumulative long-term reward, the agent needs to know which actions will realise such outcome. Temporal Difference (TD) [14, 15] learning uses this concept; learning is driven by the error/difference between temporally successive predictions, so learning occurs whenever there is a change in prediction over time. It's a method for learning to predict; learning a prediction from another later learned prediction. TD Learning algorithms are model-free. TD Learning algorithms can be on-policy, where the value of the policy being currently carried out by the agent is learnt, or off-policy, where the value of the optimal policy is learnt independently of the agent's actions [16].

1.2.3.1 Q-Learning

Q-Learning [17, 18] is an off-policy Temporal Difference Learning method. Q-Learning is defined by:

$$Q(s_t, A_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

1.2.3.2 SARSA: On-Policy TD-Learning

SARSA [19, 20] is an on-policy Temporal Difference Learning method. SARSA is defined by:

$$Q(S_t, A_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, A_t)]$$

1.3 Planning

- Planning involves reasoning on a model of an environment in order to produce a sequence of actions that will achieve a goal.[7].

1.3.1 A* Search

- A* Search [21] is not necessarily a Planning algorithm, rather it is a search algorithm. However, a simple Planning agent with the purpose of navigation in a discretised state space can use A* for planning.
- Heuristic
- Admissable

1.4 Exploration in Reinforcement Learning

Thrun [22] distinguished exploration methods into two categories: directed and undirected. A more recent work distinguished between reward-free and reward-based exploration [23].

1.4.1 Random

The most common use of randomness in exploration is through ϵ-greedy [17, 20], which aims to balance the exploration and exploitation through an ϵ factor, such that the agent exploits its learned knowledge with probability ϵ, and explores randomly with probability $1 - \epsilon$. A common thing is to decay ϵ temporally, so the agent explores a lot early on, and then exploits more after it has learnt for a while. Whilst this does provide a balance between the two extremes of pure exploration and pure exploitation, it results in continually evaluating sub-optimal actions long after they have been realised to be sub-optimal.

ϵz-greedy [24]

Softmax [25]

Random walk [26, 27]

1.4.2 Optimism

1.4.3 Intrinsically Motivated

1.4.4 Deliberate

1.5 Related Work (Planning and Learning)

- DARLING
- Dyna
- AlphaGo?

Chapter 2

Methods

<Everything that comes under the 'Methods' criterion in the mark scheme should be described in one, or possibly more than one, chapter(s).>

2.1 Meta Actions

2.1.1 Deterministic Rewards and Transitions

2.1.2 Deterministic Rewards and Stochastic Transitions

2.1.3 Stochastic Rewards and Deterministic Transitions

2.1.4 Stochastic Rewards and Transitions

Chapter 3

Results

<Results, evaluation (including user evaluation) *etc.* should be described in one or more chapters. See the 'Results and Discussion' criterion in the mark scheme for the sorts of material that may be included here.>

3.1 A section

3.1.1 A sub-section

3.2 Another section

Chapter 4

Discussion

<Everything that comes under the ‘Results and Discussion’ criterion in the mark scheme that has not been addressed in an earlier chapter should be included in this final chapter. The following section headings are suggestions only.>

4.1 Conclusions

4.2 Ideas for future work

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning - an introduction.* Adaptive computation and machine learning, MIT Press, 1998.

[2] B. F. Skinner, *The behavior of organisms : and experimental analysis / by B. F. Skinner.* Appleton-Century-Crofts New York, 1938.

[3] P. Auer and R. Ortner, "Logarithmic online regret bounds for undiscounted reinforcement learning," in *Advances in Neural Information Processing Systems* (B. Schölkopf, J. Platt, and T. Hoffman, eds.), vol. 19, MIT Press, 2006.

[4] N. K. Jong and P. Stone, "Model-based exploration in continuous state spaces," in *The Seventh Symposium on Abstraction, Reformulation, and Approximation*, July 2007.

[5] S. B. Thrun and K. Möller, "Active exploration in dynamic environments," in *Advances in Neural Information Processing Systems* (J. Moody, S. Hanson, and R. Lippmann, eds.), vol. 4, Morgan-Kaufmann, 1991.

[6] R. Dearden, N. Friedman, and S. Russell, "Bayesian q-learning," in *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI '98, (USA), p. 761–768, American Association for Artificial Intelligence, 1998.

[7] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (2nd Edition).* Prentice Hall, December 2002.

[8] S. M. LaValle, *Planning Algorithms.* Cambridge, U.K.: Cambridge University Press, 2006. Available at http://planning.cs.uiuc.edu/.

[9] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, pp. 354–, Oct. 2017.

[10] S. Levine and V. Koltun, "Guided policy search," in *Proceedings of the 30th International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 1–9, PMLR, 17–19 Jun 2013.

[11] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker, "Model-based reinforcement learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 16, no. 1, pp. 1–118, 2023.

[12] M. Leonetti, L. Iocchi, and P. Stone, "A synthesis of automated planning and reinforcement learning for efficient, robust decision-making," *Artificial Intelligence*, vol. 241, pp. 103 – 130, September 2016.

[13] M. Minsky, "Steps toward artificial intelligence," *Proc. IRE*, Jan. 1961.

[14] R. S. Sutton, *Temporal Credit Assignment in Reinforcement Learning.* PhD thesis, 1984. AAI8410337.

[15] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[16] D. Poole and A. Mackworth, *Artificial Intelligence: Foundations of Computational Agents.* Cambridge, UK: Cambridge University Press, 2 ed., 2017.

[17] C. J. C. H. Watkins, *Learning from Delayed Rewards.* PhD thesis, King's College, Cambridge, UK, May 1989.

[18] C. J. C. H. Watkins and P. Dayan, "Technical note q-learning.," *Mach. Learn.*, vol. 8, pp. 279–292, 1992.

[19] G. A. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems," Tech. Rep. TR 166, Cambridge University Engineering Department, Cambridge, England, 1994.

[20] R. S. Sutton, "Generalization in reinforcement learning: Successful examples using sparse coarse coding.," in *NIPS* (D. S. Touretzky, M. Mozer, and M. E. Hasselmo, eds.), pp. 1038–1044, MIT Press, 1995.

[21] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[22] S. Thrun, "Efficient exploration in reinforcement learning.," Tech. Rep. CMU-CS-92-102, Carnegie Mellon University, Pittsburgh, PA, January 1992.

[23] S. Amin, M. Gomrokchi, H. Satija, H. van Hoof, and D. Precup, "A survey of exploration methods in reinforcement learning," *CoRR*, vol. abs/2109.00157, 2021.

[24] W. Dabney, G. Ostrovski, and A. Barreto, "Temporally-extended ε-greedy exploration," in *International Conference on Learning Representations*, 2021.

[25] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing* (F. F. Soulié and J. Hérault, eds.), (Berlin, Heidelberg), pp. 227–236, Springer Berlin Heidelberg, 1990.

[26] C. Anderson, "Learning and problem solving with multilayer connectionist systems," 11 2001.

[27] M. C. Mozer and J. Bachrach, "Discovering the structure of a reactive environment by exploration," in *Advances in Neural Information Processing Systems* (D. Touretzky, ed.), vol. 2, Morgan-Kaufmann, 1989.

Appendix A

Self-appraisal

<This appendix should contain everything covered by the 'self-appraisal' criterion in the mark scheme. Although there is no length limit for this section, 2—4 pages will normally be sufficient. The format of this section is not prescribed, but you may like to organise your discussion into the following sections and subsections.>

A.1 Critical self-evaluation

A.2 Personal reflection and lessons learned

A.3 Legal, social, ethical and professional issues

<Refer to each of these issues in turn. If one or more is not relevant to your project, you should still explain *why* you think it was not relevant.>

A.3.1 Legal issues

A.3.2 Social issues

A.3.3 Ethical issues

A.3.4 Professional issues

Appendix B

External Material

<This appendix should provide a brief record of materials used in the solution that are not the student's own work. Such materials might be pieces of codes made available from a research group/company or from the internet, datasets prepared by external users or any preliminary materials/drafts/notes provided by a supervisor. It should be clear what was used as ready-made components and what was developed as part of the project. This appendix should be included even if no external materials were used, in which case a statement to that effect is all that is required.>