Imperial College
London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Directed Exploration in Deep Reinforcement Learning

Author:
Christopher Chong

Supervisor:
Pedro Martinez Mediano

Submitted in partial fulfillment of the requirements for the MSc degree in Computing Science of Imperial College London

September 2018

Chapter 1

Background & Progress

1.1 Reinforcement Learning

Reinforcement learning is a type of learning where the learner, or agent, learns what actions to do in an environment to maximise a scalar reward signal [14]. The learner is expected to discover the actions via trial and error without any supervision.

There are four main sub-elements in a reinforcement learning system.

- *Policy*. A policy controls how an agent behaves at a certain state. It is a result of mapping from observed states of the environment to actions to be taken at that time.
- *Reward Signal*. A reward signal is a scalar value that captures the concept of goals in a system. The agent's only objective is to maximise the total reward signal given by the environment over time.
- *Value Function*. A value function predicts the total amount of reward expected to be obtained from a particular state over the long run. It takes into account of the states that will possibly follow after the current state, hence accounting for the rewards that those future states will provide.
- *Model*. A model represents the behavior of the environment, which allows the agent to predict the possible subsequent states and rewards upon executing a certain action. Reinforcement learning methods that utilise models for planning is called model-based methods, whereas those that do not are called model-free methods.

1.2 Markov Decision Process

Markov decision process (MDP) is used to formalise the sequential decision making process in the reinforcement learning problem described above [14]. At each time step *t*, a finite MDP environment contains a set of states $s_t \in \mathcal{S}$, actions $a_t \in \mathcal{A}$ and rewards $r_t \in \mathcal{R}$ [11]. A policy π maps the state-action pair $\mathcal{S} \times \mathcal{A}$ to the probability of the agent carrying out the action at that state.

$P(s'|s,a)$ represents the probability of the agent moving into state s' from state s after taking action a at time t. The goal, or the total reward over time, can then be formally defined as an expected discounted return G_t, where

$$G_t = \sum_{t=0}^{\infty} \gamma^t r_t. \tag{1.1}$$

γ is the discount rate, where $0 \leq \gamma \leq 1$. It indicates the present value of the future rewards, which controls how much emphasis is placed on long-term rewards, as compared to short-term rewards. The value function of a state s, denoted as $V_\pi(s)$, is essentially the expectation of G_t with respect to the state-transition probability $P = P(s'|s,a)$ under a policy π, given that the initial state s_0 is s.

$$V_\pi(s) = \mathop{\mathbb{E}}_{\pi,P}[G_t|s_0 = s]. \tag{1.2}$$

$V_\pi(s)$ is known as a state-value function. We can also define an action-value function $Q_\pi(s,a)$, or Q-value function, where

$$Q_\pi(s,a) = \mathop{\mathbb{E}}_{\pi,P}[G_t|s_0 = s, a_0 = a]. \tag{1.3}$$

An optimal policy π_* is a policy which achieves the largest expected return over the long run. The corresponding optimal Q-value function can be obtained by

$$Q_*(s,a) = max_\pi Q_\pi(s,a). \tag{1.4}$$

Following this equation, we can obtain the optimal policy π_* by selecting the action that yields the highest Q-value ($\pi_* = arg\ max_{a'} Q_*(s,a')$). Using bootstrapping technique, we can show that Equation (1.3) can be written as

$$Q_\pi(s,a) = \mathop{\mathbb{E}}_{\pi,P}[r_{t+1} + \gamma Q_\pi(s_{t+1}, a_{t+1})|s_0 = s, a_0 = a]. \tag{1.5}$$

Equation (1.5) is known as the Bellman expectation equation. Similarly, Equation (1.4) can be written as

$$Q_*(s,a) = \mathop{\mathbb{E}}_{P}[r_{t+1} + \gamma max_{a'} Q_*(s_{t+1}, a')|s_0 = s, a_0 = a]. \tag{1.6}$$

Equation (1.6) is known as the Bellman optimality equation.

1.3 Q-Learning

Q-learning is an off-policy temporal difference learning algorithm for the Markovian environment described above. The update rule of the Q-value function is defined as below [18].

$$\Delta Q(s_t, a_t) = \alpha(r_{t+1} + \gamma max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)). \tag{1.7}$$

α is the step-size parameter, and the remaining terms that follow α indicate the error estimate for state s_t. The Q-value function here directly approximates the optimal

Q-value function independent of the policy used [14], which is the reason why it is an off-policy algorithm.

1.3.1 Deep Q-Network

In order to generalise to high-dimensional problems, a function approximator is needed to represent the Q-value function, where $Q(s, a; \theta) \approx Q_*(s, a)$. A deep Q-network (DQN) uses a deep convolutional neural network as the approximator with weights θ [10]. The agent is trained by minimising the following loss function at each iteration i.

$$L_i(\theta_i) = \mathop{\mathbb{E}}_{s,a,r,s' \sim D} \left[\left(r + \gamma max_{a'} Q_{target}(s', a'; \theta_{i-1}) - Q(s, a; \theta_i) \right)^2 \right] \tag{1.8}$$

This algorithm is model-free as it does not construct an estimate of the environment. Instead, it uses a mechanism known as experience replay, which saves the agent's experience (s_t, a_t, r_t, s_{t+1}) at each time step t into a data-set D. This is conducted over all the episodes of training. During learning, Q-learning updates are performed on samples of these experiences (s, a, r, s') drawn randomly from the data-set D. Note also that Q_{target} in Equation (1.8) is updated less frequently than the main network to stabilise the learning.

1.4 Exploration v.s. Exploitation

The exploration-exploitation dilemma is a significant problem in reinforcement learning [14]. An agent should exploit the actions that have given high rewards in the past. On the other hand, the agent has to have explored these actions in the first place in order to discover them. The balance between exploration and exploitation needs to taken care of in order to achieve an optimal solution in any reinforcement learning problem. All exploration techniques are generally classified into either undirected or directed exploration.

1.4.1 Undirected Exploration

An undirected exploration relies on randomness to explore, without deriving any information from the environment to guide exploration [16]. The ϵ-$greedy$ approach is a simple undirected exploration strategy. Most of the time, the agent behaves greedily by selecting the action with the highest Q-value to maximise immediate reward [14]. However, there is a small probability ϵ that the agent will implement one of the non-greedy actions (with equal probability). The main disadvantage of this method is due to its undirected nature, where the non-greedy actions are treated the same despite some of them are nearly greedy while others are not.

1.4.2 Directed Exploration

Directed exploration methods direct agents to choose actions to explore based on knowledges gathered from the learning process [16]. These actions allow agents to derive maximum information from the environment. There are a few principles that most directed exploration strategies are based on. Optimism in the face of uncertainty (OFU) and intrinsic motivation are some of these principles.

Optimism in the Face of Uncertainty

OFU is a heuristic that most provably-efficient directed exploration approaches rely upon [11]. Following this heuristic, agents are encouraged to take actions that have high uncertainty in the Q-value estimation [13]. These uncertain actions can be actions that were tried only for a small number of times, hence they are poorly understood. Although the reward received from these actions might be small, there is a good chance that these actions are actually the true optimal actions (especially if the estimated Q-values of these actions are close to that of the current optimal action). By exploring these uncertain actions, the agent can efficiently reduce their uncertainties, which in turn helps optimising the agent's performance.

Intrinsic Motivation

This heuristic is inspired by related researches in the field of developmental psychology of human beings [12]. When engaging in curiosity-driven and risk-taking activities, rational agents generate intrinsic goals. Upon accomplishing these goals, the agent gains intrinsic rewards. These rewards in turn instigate the agent to explore the environment more. In a reinforcement learning setting, the reward given to the agent is extrinsic. By incorporating intrinsic goals into the agent, exploratory values can be assigned to the agent's actions to promote exploration. The exploration in turn gives intrinsic rewards (in addition to any extrinsic reward) to the agent. One of the drawbacks of this heuristic is that it does not account for the reward structure of the system [8].

1.4.3 Low-Dimensional Problem

In low-dimensional tasks, where the action and state spaces are small and discrete, several methods are able to manage the exploration-exploitation trade-off very effectively [6].

One of these methods is Bayesian reinforcement learning [4]. In this method, the Q-values have an initial distribution, and they are updated after every time step. Thompson sampling is the technique used to select actions according to their posterior probability of being the greedy action. Given that, the rewards and the probabilities of every possible chain of events can be calculated (as the problem is low-dimensional). This allows the optimal balance between exploration and exploitation to be computed.

Another class of method is known as the PAC-MDP method, where one of its algorithms called R_{max} [3] has developed a built-in mechanism to address the exploration-exploitation dilemma. This algorithm is based on the OFU heuristic described above. The agent in this model-based approach acts based on the optimal policy derived from a fictitious model. In the real model, this optimal behavior is either translated into the optimal action or an exploratory action.

1.4.4 High-Dimensional Problem

Undirected Exploration

In high-dimensional tasks, the exploration-exploitation trade-off is still a long standing open problem. Methods that used undirected exploration strategy take exponential time to learn the optimal policy [7]. The DQN algorithm discussed above [10] is one of these methods. It does not perform well in sparse-reward environments, which require a very efficient exploration strategy. This is because a productive long sequence of actions is needed to get the long-delayed reward in such environments.

Directed Exploration

Various directed exploration strategies have been developed to tackle this infamous trade-off in high-dimensional problems. Some of these strategies are based on one heuristic (e.g. risk-seeking), whereas some are trying to connect both the notions of OFU and intrinsic motivation (e.g. pseudo-count) [11].

Count-Based

The pseudo-count based method is extended from the count-based method [13], which performs efficiently on low-dimensional problems. This strategy uses the visit count of a state-action pair to approximate the state's uncertainty, where state-action pairs with lower visit counts are given higher uncertainty. An exploration bonus (in addition to the normal reward signal) is given to each of the agent's visit to the uncertain states.

This approach, however, fails on high-dimensional problems due to the lack of generalisation of uncertainties. Two states with familiar features should have similar uncertainties despite one of them has not been visited before. Without such generalisation, most of the state spaces will be assigned with the same exploration bonus and they appear equally uncertain to the agent. This is because most of the states are not being visited in a high-dimensional problem [13].

Pseudo-Count Based

The pseudo-count based method solves the issue above by generalising the uncertainty across states [1]. This method utilises a qualitative exploration guidance from the intrinsic motivation heuristic – the change in prediction error, which is also known as *learning progress*. Pseudo-count is a new quantity invented to connect

learning progress with the original count-based method described above. However, there are a few drawbacks associated with this method. One of them is the unprincipled way of measuring similarity between states [13].

Risk-Seeking

This method aims to encourage risk-seeking behaviours in agents, which serve as the agent's exploration heuristic. The risk here is represented by the variance of the expected discounted return G_t in Equation (1.1). A Bellman-like relationship is first utilised to derive a loss function for this variance [15]. The loss function is then used by a neural network estimator to estimate the risk value. A new risk-seeking value is also introduced to represent agents of different degrees of risk-seeking tendency. OFU heuristic is then applied by directing the agent to select actions that maximise the risk-seeking value instead of the usual Q-value. One of the shortcomings of this method is that an agent tends to seek sub-optimal solutions in a stochastic environment.

1.4.5 Empirical Evaluation

Many directed exploration strategies (including the ones discussed above for high-dimensional settings) were evaluated on Atari 2600 games from the Arcade Learning Environment (ALE) [2]. These games are considered to be high-dimensional as they contain a visual state of $210 \times 160 \times 3$ array of RGB pixels.

A few of these games are classified as hard sparse-reward games [1]. This means that agents in these games could not use a $\epsilon\text{-}greedy$ approach to achieve a high-scoring policy. One of these games is Montezuma's Revenge. It is known as perhaps the hardest game available in ALE due to its highly sparse reward system and hostile environment. Table 1.1 shows the average score of this game achieved by agents using the pseudo-count and risk-seeking approach respectively [1]. Note that both of these approaches used different versions of the original DQN algorithm [10].

Exploration Strategy	Average Score (by 100 million frames)
Pseudo-Count	3439
Risk-Seeking	556

Table 1.1

1.5 Project Direction

This project aims to build on top of one of the latest state-of-the-art directed exploration strategies used in high-dimensional reinforcement learning settings. We aim to develop novel extensions to address the drawbacks that one of those strategies faces. Pseudo-count and risk-seeking approaches discussed above are two options

that this project might choose from. DQN or an extended version of DQN [5] will be used as the agent to experiment with the extended directed exploration strategy. Empirical evaluations of this method will be conducted in a simulated environment, either in games (e.g. ALE [2]) or robotics simulations (e.g. MuJoCo [17]).

1.6 Legal & Ethical Considerations

	Yes	No
Section 1: HUMAN EMBRYOS/FOETUSES		
Does your project involve Human Embryonic Stem Cells?		✓
Does your project involve the use of human embryos?		✓
Does your project involve the use of human foetal tissues / cells?		✓
Section 2: HUMANS		
Does your project involve human participants?		✓
Section 3: HUMAN CELLS / TISSUES		
Does your project involve human cells or tissues? (Other than from Human Embryos/Foetuses i.e. Section 1)?		✓
Section 4: PROTECTION OF PERSONAL DATA		
Does your project involve personal data collection and/or processing?		✓
Does it involve the collection and/or processing of sensitive personal data (e.g. health, sexual lifestyle, ethnicity, political opinion, religious or philosophical conviction)?		✓
Does it involve processing of genetic information?		✓
Does it involve tracking or observation of participants? It should be noted that this issue is not limited to surveillance or localization data. It also applies to Wan data such as IP address, MACs, cookies etc.		✓
Does your project involve further processing of previously collected personal data (secondary use)? For example Does your project involve merging existing data sets?		✓
Section 5: ANIMALS		
Does your project involve animals?		✓
Section 6: DEVELOPING COUNTRIES		
Does your project involve developing countries?		✓
If your project involves low and/or lower-middle income countries, are any benefit-sharing actions planned?		✓
Could the situation in the country put the individuals taking part in the project at risk?		✓

Section 7: ENVIRONMENTAL PROTECTION AND SAFETY		
Does your project involve the use of elements that may cause harm to the environment, animals or plants?		✓
Does your project deal with endangered fauna and/or flora /protected areas?		✓
Does your project involve the use of elements that may cause harm to humans, including project staff?		✓
Does your project involve other harmful materials or equipment, e.g. high-powered laser systems?		✓
Section 8: DUAL USE		
Does your project have the potential for military applications?		✓
Does your project have an exclusive civilian application focus?		✓
Will your project use or produce goods or information that will require export licenses in accordance with legislation on dual use items?		✓
Does your project affect current standards in military ethics e.g., global ban on weapons of mass destruction, issues of proportionality, discrimination of combatants and accountability in drone and autonomous robotics developments, incendiary or laser weapons?		✓
Section 9: MISUSE		
Does your project have the potential for malevolent/criminal/terrorist abuse?		✓
Does your project involve information on/or the use of biological-, chemical-, nuclear/radiological-security sensitive materials and explosives, and means of their delivery?		✓
Does your project involve the development of technologies or the creation of information that could have severe negative impacts on human rights standards (e.g. privacy, stigmatization, discrimination), if misapplied?		✓
Does your project have the potential for terrorist or criminal abuse e.g. infrastructural vulnerability studies, cybersecurity related project?		✓
Section 10: LEGAL ISSUES		
Will your project use or produce software for which there are copyright licensing implications?		✓
Will your project use or produce goods or information for which there are data protection, or other legal implications?		✓
Section 11: OTHER ETHICS ISSUES		
Are there any other ethics issues that should be taken into consideration?		✓

1.7 Progress & Plan

1.7.1 Progress

Most of the efforts up-to-date have been spent on reading and understanding the fundamental theories of reinforcement learning. *Reinforcement Learning: An Introduction* by Richard Sutton and Andrew Barto [14] is the main resource used to study the basic theories. Before delving into studying different exploration strategies, various general RL algorithms used in high-dimensional settings are currently being studied, such as DQN and A3C [9].

Two exploration heuristics are currently being focused on, which are OFU and intrinsic motivation, as discussed above. The risk-seeking method, which is proposed by Nat Dilokthanakul (one of the supervisors for this project), is given extra attention to as it could potentially be the idea that this project aims to extend from.

1.7.2 Plan

As this project is more theory-focused, studying and understanding the fundamental theories in the beginning phase of the project is crucial for the subsequent development of extension ideas in the later stage. Meanwhile, practical implementations of theories via programming are also emphasised from the beginning to ensure the development of practical experiences from the start.

Table 1.3 shows the short-term plan (mainly in June) to build up sufficient theoretical knowledges and practical experiences to tackle the research question, whereas Table 1.4 shows the subsequent strategy to achieve the ultimate objective of the project. Note the use of *early* and *late* in the tables; the former indicates the first half of the month, whereas the latter indicates the second half of the month.

Task	Period
Study fundamental reinforcement learning theories	early June
Implement Q-learning in low-dimensional settings	early June
Study directed exploration strategies of high-dimensional settings	late June
Study and implement simple convolutional neural networks	late June
Implement deep Q-networks in simple simulated environments	late June

Table 1.3

Task	Period
Experiment with and finalise the simulated environment	early July
Develop the extension of the finalised exploration strategy	early July
Construct a suitable extension of DQN for the extended idea	late July
Evaluate the strategy in the chosen simulated environment	late July
Identify weaknesses in the extended idea and make improvements	early August
Compile results and write the first draft of the final report	early August
Finalise the results and edit drafts of the final report	late August

Table 1.4

Bibliography

[1] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1471–1479, 2016. pages 5, 6

[2] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013. pages 6, 7

[3] Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3(Oct):213–231, 2002. pages 5

[4] Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, Aviv Tamar, et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015. pages 4

[5] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. *arXiv preprint arXiv:1710.02298*, 2017. pages 7

[6] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pages 1109–1117, 2016. pages 4

[7] Sham Machandranath Kakade et al. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003. pages 5

[8] Jan Leike. Exploration potential. *CoRR*, abs/1609.04994, 2016. pages 4

[9] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016. pages 9

[10] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015. pages 3, 5, 6

[11] Brendan O'Donoghue, Ian Osband, Remi Munos, and Volodymyr Mnih. The uncertainty bellman equation and exploration. *arXiv preprint arXiv:1709.05380*, 2017. pages 1, 4, 5

[12] Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009. pages 4

[13] Suraj Narayanan Sasikumar. Exploration in feature space for reinforcement learning. *arXiv preprint arXiv:1710.02210*, 2017. pages 4, 5, 6

[14] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998. pages 1, 3, 9

[15] Aviv Tamar, Dotan Di Castro, and Shie Mannor. Learning the variance of the reward-to-go. *Journal of Machine Learning Research*, 17(13):1–36, 2016. pages 6

[16] Sebastian B Thrun. Efficient exploration in reinforcement learning. 1992. pages 3, 4

[17] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5026–5033. IEEE, 2012. pages 7

[18] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992. pages 2