Assessment 5 - SIT 796

Survey on Exploration Methods in Deep Reinforcement Learning

1. Introduction

The survey proposes a single level classification of Exploration Methods in Deep Reinforcement Learning. The proposed segregation from [15] is modified on the fundamental working principle of the classified method with supporting examples for each method in the category to explain the method comprehensively.

2. Classification of Exploration Methods

The end goal of an agent in Reinforcement Learning is to learn based on its interaction with the environment. This action is generally taken utilizing the principle of exploration and exploitation. During exploitation the agent acts in a way which uses the learnings from prior interactions of the agent with the environment and maximizes the known reward. During exploration the agent tends to take actions which have not yet been taken and expect the reward.

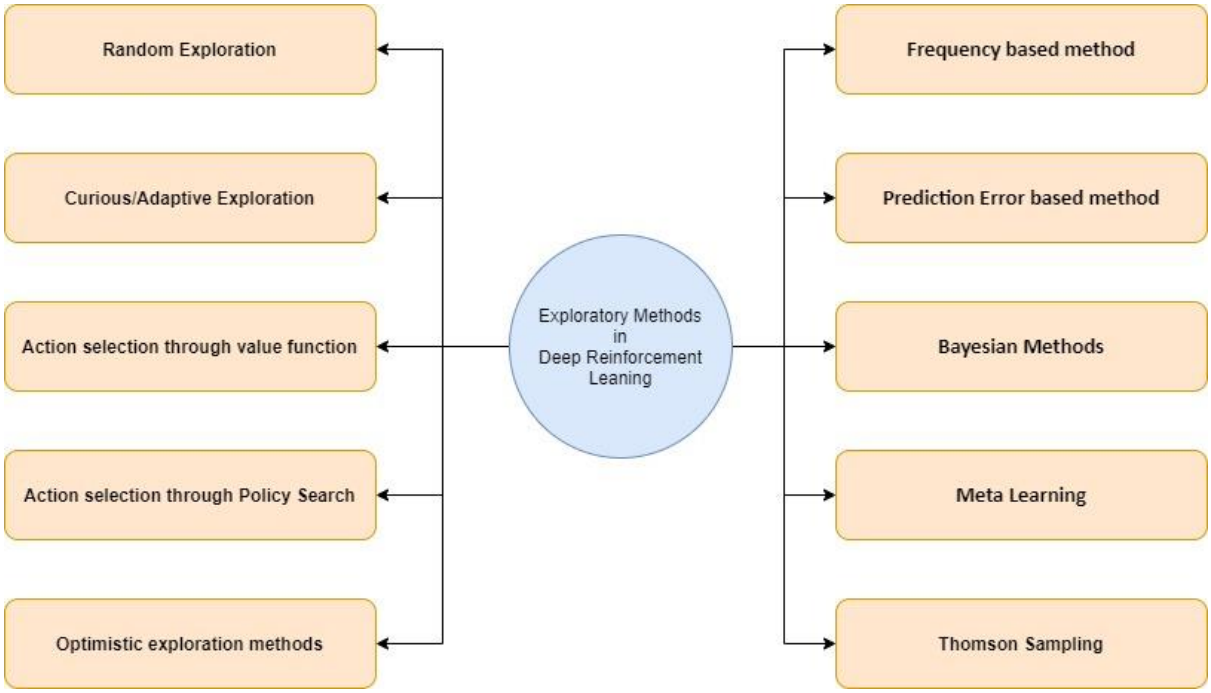

Fig 1: Classification of Exploratory methods used in reinforcement learning.

2.1 Random Exploration

Random exploration method is one the most basic exploration methods which does not act on any intuition to select any set of actions in response to the observed state of the environment. The driving factor behind this exploration is *randomness*. The agent chooses a random action irrespective of the information it receives from the environment, this leads to a suboptimal solution as the agent might move away from required goal.

An effective variation of random exploration was introduced in [1] as ε-greedy action selection. This has shown to be quite effective [2] in multiple RL scenarios. The parameter ε (denotes the probability to explore) in range [0,1], controls the balance between exploitation and exploration.

This is formally defined as:

$$a_t = \begin{cases} a_t' & \text{with probability } 1 - \varepsilon \\ \text{random action} & \text{with probability } \varepsilon \end{cases}$$

Where a_t' is the greedy action based on the greedy policy (exploitation). Random action denotes exploration.

2.2 Curious/Adaptive Exploration

[3] define *curiosity* as improving predictor of the reactions of a world model. In this method, an adaptive world model is used which instigates the controller to create scenarios where the expected response is different from the reality. This focusses on building model which encourages tough action sequences for the agent, this drives the agent towards exploration. This helps in *predicting the expectation of the sum of future error changes [3]*. While this is not in itself an exploration method, this strategy can be utilized as a submodule in the learning system which pushes the agent to make explorative choices as it will be intrinsically rewarded more for exploration.

This kind of exploration strategy is also known as reward free exploration strategy, where an intrinsic motivation is formed to encourage exploratory behavior of the agent. The focus of these strategies is to increase the frequency of such scenarios where the model performs poorly and thus encourage exploration.

2.3 Action selection through value function

Till now the approaches discussed do not have an intuition behind selecting action in response to the observed state of the environment. An intuitive approach is to assign transition probabilities to the actions of the agent. These probabilities are based on the value function which estimate the value for the current state: *s,* for time: *t.* This acts as a *reward* or feedback from the environment to the agent which guides the agent towards choosing the set of actions which lead to maximum rewards.

The best action selection is generally done using a greedy approach, [4] introduce a *Softmax* action selection method. In this, the current best action using greedy approach is assigned the highest probability and the rest of the actions are assigned probabilities with respect to their values. This leads the agent to choose actions towards maximizing the rewards.

2.4 Action selection through Policy Search

Like *Action selection through value function,* the agent is provided feedback from the environment to move towards maximum rewards. This approach differs from the previous one by the method of action selection by using a policy, instead of a value function, or in combination with a value function. When used in combination with a value function this makes a special case of *Actor-Critic* methods: the critic performs the estimation of value function, the actor updates the policy distribution as per the feedback of the critic.

These methods either change the action space (generally over single time step) or the parameter space (over entire episode). [5] demonstrate how intermediate strategies could be employed during two independent time steps when the action space is being changed. In contrast to these approaches parameter-space changes affect entire episode. One such episode-based policy search method which does parameter-space exploration was introduced in [6], where the effectiveness of the policy exploration is calculated by state-dependent exploration i.e., evaluating changes across multiple states.

2.5 Optimistic exploration methods

One more category for exploration method is combining reward with additional *bonus* [7] term. This makes the total reward to be the aggregation of following two terms:

$$\text{Reward}_{\text{total}}(s, a) = \text{Reward}(s, a) \oplus \text{Bonus}(s, a)$$

$\oplus$ Is the operator for aggregation of environment reward and bonus term.

Optimistic exploration methods utilize this *bonus* either in form of *Optimistic Initialization* [8] or *Upper Confidence Bounds* [UCBs] [9]. In optimistic initialization, unvisited state-action pairs are assumed to give the best outcome and in the latter method it is assumed that the pairs would yield outcome related to the maximum achievable reward. In this approach, a state is considered optimal when in has a high level of uncertainty and return potential. This approach is biased towards pushing the agent towards unknown state action pairs, as they have *bonus* term associated with them. This results in agent to behave in an explorative way, as unvisited pairs tend to lead the agent towards maximum rewards. These methods are generally of two forms: tabular or function approximation methods.

2.6 Frequency based method

Another method that utilizes the previously defined bonus is *Frequency Based Method.* This method stores in the memory the count for each visit done on all sets of state action pairs. This method can also be categorized into tabular and approximation methods. The methods in both the form introduce a calculation criterion based on which the frequency count of visited state action pair is quantified and aggregated with environment reward. This leads the agent towards exploring new state action pair due to the additional rewards associated with them.

[10] use such a technique to propose a *prioritized sweeping* approach that utilizes a predefined parameter as threshold to calculate whether a state action pair should be visited or not. This threshold parameter can be set manually thus offering control to finetune the behavior of the agent in response to the environment. A non-discounted return is used by the approach once the frequency count increases this configured threshold.

2.7 Prediction Error based method

This approach is also an example of manipulating total reward by utilizing *bonus* as defined in section 2.6. This approach also borrows some of its working from the curious/adaptive exploration. [3] and [11] have first proposed prediction error as an additive *bonus* term with the environment reward to push the agent towards difficult states. This approach differs from the previous two approaches as the bonus term here can be used as a standalone reward thus making this approach suitable where agent does not rely on the environment reward (as discussed in section 2.2).

The underlying assumption behind this approach is that the unvisited state-action pairs would be more difficult than visited pairs for the agent to solve, this implies that this approach does not require

storing unvisited state action pairs. This assumption pushes the agent towards difficult (unvisited) state action pairs thus instilling explorative behaviour in the agent.

2.8 Bayesian Methods

In Bayesian Methods, a posterior is defined as a function mapping the interaction of the agent with the environment. This posterior is used to approximate optimal decisions for a given Markov Decision Process. A survey containing approaches specific to Bayesian methods is covered comprehensively in [12]. One such approach is demonstrated in [13] where approximation of policies and value functions is done by solving linear functions constituting state components combinations. A policy update rule based on Monte-Carlo method is used to update the policy.

The underlying principle beneath the posterior used in Bayesian methods is that as the agent's observation increases the update in belief reflects the new information from increased observation set.

2.9 Meta Learning

Certain exploration techniques emerge from learning a bias from a set of tasks that enable increase in learning efficiency from unseen tasks like previous encountered tasks [14]. This theory forms the underlying working principle of meta-reinforcement learning. In simpler words, a meta reinforcement learning model has the capability to adopt to unseen environments on which it is not trained. The model utilizes the learning experience of a previous model to respond to new environment, this is possible as a general meta learning model has two components: A learner which learns to behave in simulation environment and an optimizer which updates the model in response to the unseen environment. This capability intrinsically generates exploratory nature in the model.

A major distinguishing factor of meta reinforcement learning, and traditional reinforcement learning is that, in meta-RL a distribution of MDPs over same state-action space is considered whereas in traditional RL a single MDP is used to model the problem.

2.10 Thomson Sampling

Thompson Sampling [15] is an algorithm where actions are processed sequentially to balance exploitation and exploration. Recall that in greedy approach the best value action is chosen, this might sometimes yield a suboptimal action plan. Thomson Sampling, also known as Probability Matching, addresses this by building a probability model from the obtained rewards by interactions of agent with the environment. This model is then used to sample the actions, which pushes the agent towards maximum reward and optimal solution. After an action is chosen the algorithm updates the probability distribution with observed action-space pair. This probability update step encourages exploration behavior in the agent.

Thomson Sampling makes it possible to estimate the mean reward of the actions. This is complemented by another value which represents the confidence score of the estimate. These properties allow the method to choose optimal action and result in near optimal total reward.

3. Conclusion

This survey presents a classification of exploratory methods used in deep RL. The paper introduces a basic exploration technique: Random Exploration and build its way up to complex exploratory methods used. The exploratory nature of each presented method is also highlighted.

4. References

[1] R. S. Sutton, "Generalization in reinforcement learning: Successful examples using sparse coarse coding," *Advances in neural information processing systems*, vol. 8, 1995.

[2] R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 1054–1054, 1998.

[3] J. Schmidhuber, "Curious model-building control systems," in *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*, 1991.

[4] J. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," *Advances in neural information processing systems*, vol. 2, 1989.

[5] P. Wawrzyński, "Control policy with autocorrelated noise in reinforcement learning for robotics," *Int. J. Mach. Learn. Comput.*, vol. 5, no. 2, pp. 91–95, 2015.

[6] T. Rückstieß, F. Sehnke, T. Schaul, D. Wierstra, Y. Sun, and J. Schmidhuber, "Exploring parameter space in reinforcement learning," *Paladyn*, vol. 1, no. 1, 2010.

[7] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Trans. Evol. Comput.*, vol. 11, no. 2, pp. 265–286, 2007.

[8] E. Even-Dar and Y. Mansour, "Convergence of optimistic and incremental Q-learning," in *Advances in neural information processing systems*, 2001.

[9] A. L. Strehl and M. L. Littman, "An analysis of model-based Interval Estimation for Markov Decision Processes," *J. Comput. Syst. Sci.*, vol. 74, no. 8, pp. 1309–1331, 2008.

[10] A. W. Moore and C. G. Atkeson, "Prioritized sweeping: Reinforcement learning with less data and less time," *Mach. Learn.*, vol. 13, no. 1, pp. 103–130, 1993.

[11] J. Schmidhuber, "A possibility for implementing curiosity and boredom in model-building neural controllers," in *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, 1991, pp. 222–227.

[12] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar, "Bayesian reinforcement learning: A survey," *arXiv [cs.AI]*, 2016.

[13] M. O. Duff, "Design for an optimal probe," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 131–138.

[14] R. Vilalta and Y. Drissi, "A perspective view and survey of meta-learning," *Artificial intelligence review*, vol. 18, no. 2, pp. 77–95, 2002.

[15] S. Amin, M. Gomrokchi, H. Satija, H. van Hoof, and D. Precup, "A survey of exploration methods in reinforcement learning," *arXiv [cs.LG]*, 2021.