DDA4230 Reinforcement learning	Exploration in deep RL
Lecture 20	
Lecturer: Guiliang Liu	*Scribe: Baoxiang Wang*

1 Goal of this lecture

In this lecture we revisit exploration in reinforcement learning. We previously discussed multi-armed bandits whose main concern is the trade-off between exploration and exploitation. We provide some insights how such ideas in bandits can be generated to deep reinforcement learning.
Suggested reading: Chapter 17 of *Reinforcement learning: An introduction*; CS285 *Deep reinforcement learning* from UC Berkeley.

2 Hardness of exploration

Recall that the Atari game Montezuma's Revenge is played with several factors as

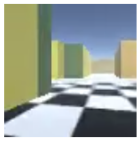
- Getting key → reward
- Opening door → reward
- Getting killed by skull → nothing (is it good? bad?)
- Finishing the game only weakly correlates with rewarding events
- We know what to do because we understand what these sprites mean!

This game is notoriously difficult for reinforcement learning and extensive efforts have been deployed to solve the problem.

Figure 1: Montezuma's Revenge can be difficult.

On the other hand of the story, emphasizing too much on exploration could result in suboptimal results as well. An argument is the so-called *noisy TV* experiment. Assume that in a maze an agent aims at finding the correct path to escape the maze. In the maze, there is a TV that presents noisy signals which is very large in randomness and is complete irrelevant to the task. An agent that focuses on exploration could be trapped into sitting in front of the TV and watching the TV for good, without solving the task. This happens when observing new states rewards the agent in an undiminished way, which could happen with many methods who rely on the intrinsic reward.

Agent in a maze with a noisy TV



Agent in a maze without a noisy TV

Figure 2: Noisy TV.

3 The exploration-exploitation dilemma

There are two potential definitions of the exploration problem

- How can an agent discover high-reward strategies that require a temporally extended sequence of complex behaviors that, individually, are not rewarding?
- How can an agent decide whether to attempt new behaviors (to discover ones with higher reward) or continue to do the best thing it knows so far?

And this corresponds to the balance between exploitation and exploration

- Exploitation: doing what you know will yield highest reward
- Exploration: doing things you haven't done before, in the hopes of getting even higher reward

3.1 Exploration as POMDPs

Recall that in bandits, the state space is $\mathcal{S} = \{1\}$, the action space is $\mathcal{A} = [m]$, and the reward function is $r(i) \sim \mathbb{P}(r \mid \theta_i)$ for some unknown parameter θ_i.

One way to deal with exploration is to introduce a state space Θ, where the state $\theta \in \Theta$ keeps track of the belief of such unknown parameters. In this way the problem of exploration can be seen as a combination of maintaining the belief of θ_i and the learning of decisions over the POMDP. Though, this will introduce unnecessarily high complexity to the problem.

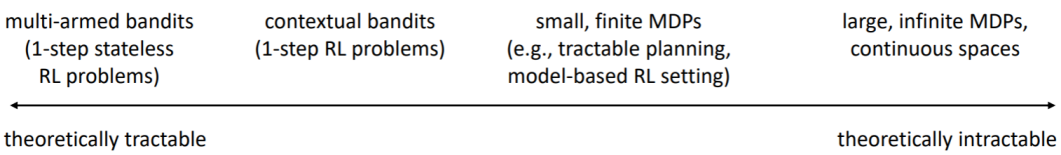


Figure 3: Tractability of exploration in a variety of settings.

3.2 Exploration strategies in bandits

Optimistic exploration Optimism is the fundamental underlying idea of UCB-based methods. While we maintain an empirical mean reward $\mu_{i,t}$ for each arm, instead of picking the arm with the largest empirical mean, we add a exploration bonus $\sigma_{i,t}$ as some sort of variance estimate, for example

$$\sigma_{i,t} = \sqrt{\frac{2 \log t}{N_{i,t}}}.$$

The optimism ideas is to keep trying an arm until it is sure that the arm is suboptimal (with probability at least $1 - \delta$, as an intuition by Hoeffding's inequality in UCB). This idea extends to other algorithms as well.

Posterior sampling Posterior sampling, known as Thompson sampling, is similar to the formulation of POMDPs. However instead of using the full distribution over the unobserved state, the algorithm simply samples one state from the belief space and executes the exploitation process using this state. In bandits, this equals to pretending the Thompson sampling to be the true mean reward and running the greedy algorithm. The algorithm dates back to 1933 but was grants rigorous theoretical guarantees only recently.

Information theory From the perspective of statistics, let the optimal action be a^* and let the optimal value be V^*. The goal of reinforcement learning is to determine either a^* or V^*, depending on if the approach is policy-based or value-based. Taking a^* as an example, at the start of the learning, a^* is a random variable with high entropy, i.e. we have limited information about a^*. With the learning process, the entropy $H(a^*)$ will decrease.

Without loss of generality, let z be a latent variable we aim to determine, which can be a^*, V^*, or other latent variables such as goals for goal-conditioned policies. Then, with a new observation y, this entropy is reduced by

$$H(z) - H(z \mid y),$$

where y can be next states and reward signals. Then it is natural to formulate the information gain of z given y as $IG(z, y) = \mathbb{E}_y[H(z) - H(z \mid y)]$.

The information gain can be utilized to encourage exploration. For example in [RVR18], in bandits, it chooses arm i with the smallest $\widehat{\Delta}_i / IG(\theta_i \mid a_t = i)$. In reinforcement learning, $IG(\theta_i \mid a_t = i)$ can no longer be estimated unless there is a model for the reward r and the next state s'. We will discuss how this extension can be done later in this lecture.

4 Exploration in reinforcement learning

We see different ideas of exploration in bandits. These ideas can be extended to reinforcement learning, where a state transition is introduced.

1. Optimistic exploration
   - New state = good state;
   - Requires estimating state visitation frequencies or novelty;
   - Typically realized by means of exploration bonuses.
2. Posterior sampling
   - Learn distribution over Q-functions or policies;
   - Sample and act according to sample.
3. Information gain
   - Reason about information gain from visiting new states;
   - Reason about information gain from achieving new goals.

4.1 Optimistic exploration in RL

Pseudo-count Recall that the UCB algorithm chooses

$$a = \arg\max_i \hat{\mu}_{i,t} + \sqrt{\frac{2\log t}{N_{i,t}}}\,.$$

For RL, it is natural to extend the exploration bonus $\sqrt{\frac{2\log t}{N_{i,t}}}$ with some generalized count-based bonus term $B(N_{s,t})$, where $B(\cdot)$ is a monotonically decreasing function that determines the bonus. This term can be added to the reward which generalizes the mean estimation of bandit arms, as $r_b = r + B(N_{s,t})$. This can be done for any model-free algorithms in discrete state spaces.

Count-based methods raise a natural concern that exactly the same state is almost never observed for a second time. We then resort to the observation that some states are more similar than others.

One approach is to fit a generative model $\mathbb{P}_\theta(s)$, which describes the distribution of the states that have been visited so far. Once a new state s' is visited, $\mathbb{P}_\theta(s')$ will describe how dissimilar s' is compared to previous experiences. A lower $\mathbb{P}_\theta(s')$ will result in a larger exploration bonus. This term is known as the pseudo-count, which generates the frequency $N_{s,t}/N$ of discrete spaces, where N denotes the number of experiences. The update of $\mathbb{P}_\theta(s)$ is by incrementally estimating a new model and the count is calculated in analogous to the update of frequency by solving

$$\mathbb{P}_\theta(s) = \frac{N_{s,t}}{N}\,, \quad \mathbb{P}_{\theta'}(s) = \frac{N_{s,t}+1}{N+1}\,,$$

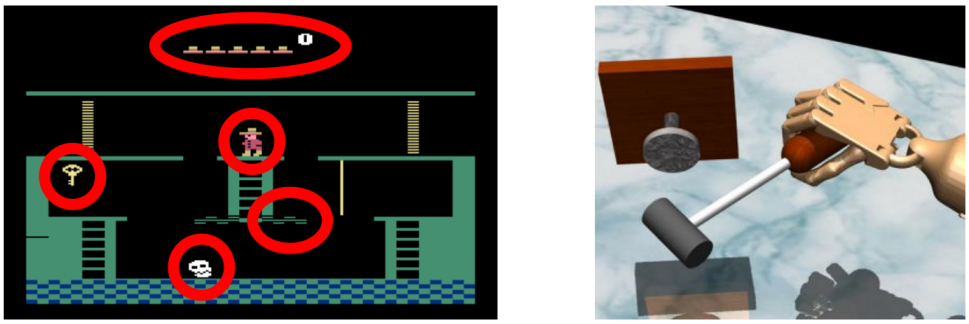


Figure 4: States are almost always unique in continuous tasks.

where $\mathbb{P}_{\theta'}(s)$ is the updated generative model after visiting s. Choices of the bonus term include $B = \sqrt{\frac{2\log t}{N_{s,t}}}$, $B = \sqrt{\frac{1}{N_{s,t}}}$, $B = \frac{1}{N_{s,t}}$, etc.

It is up to the choice of using different generative models $\mathbb{P}_\theta(s)$. This is simple, as $\mathbb{P}_\theta(s)$ is only used to measure the density of a newly-observed sample and we do not rely on the measure itself to generate samples. Concerns raised in generative adversarial networks do not need to be considered here. Simple neural networks will work.

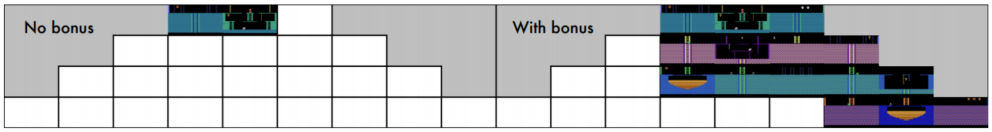


Figure 5: The efficacy of exploration by pseudo-count shown in [BSO+16].

Counting via errors As we have just discussed, the generative model need to be able to output the densities, but does not necessarily need to produce good samples. Then a simpler approach is to use the (s, a) pair to estimate some function $f(s, a)$ and then treat the estimation error as the novelty of this new (s, a) pair. As a consequence, the exploration bonus is $\|\hat{f}(s, a) - f(s, a)\|$ which is added to the reward.

It is very natural to choose $f(s, a)$ to be the state transition model which is to predict the next state s'. This target $f(\cdot)$ has an alternative interpretation of information gain. It is also possible to simply predict a function $f(\cdot)$ that is randomly determined in the beginning of the learning.

4.2 Posterior sampling

Recall that Thompson sampling in bandits samples a bandit instance θ each time according to the posterior estimation based on the current history. This extends to reinforcement learning via sampling a Q-function from a distribution of Q-functions at each step. The reason we can do this is that Q-learning is off-policy and consequently the experience collected at each round can be used in the future no matter which Q-function is used to generate the experiences.

The space of distributions over a function is large and intractable in general. We therefore seek such representations of distributions over Q-functions. One approach is to simply train multiple Q-functions using the same set of data, by sampling with replacement a subset of which in the training of each of the Q-function. These Q-functions can share some low-level features in the neural network to reduce the computational cost [OBPR16]. Another approach is to use distributional Q-learning, where the Q-function estimates a distribution over the action value given a state-action pair instead of estimating its mean [DRBM18, BDM17]. Posterior sampling can then sample one action value of that distribution.

An important observation is to use randomized value estimations instead of random policies in posterior sampling. As exploring with random actions (e.g., epsilon-greedy) can incur oscillation back and forth and might not go to a coherent or interesting state. Exploring with random Q-functions can commit to a randomized but internally consistent strategy for an entire episode.

4.3 Information gain

We first need to reason about which variable we should consider for information gain. Despite that information gain is not tractable in general in reinforcement learning, different variables will have different implications. $IG(r)$ seems uninformative. $IG(\mathbb{P}(s))$ on the generative model makes some sense, but can be very hard to compute. $IG(\mathbb{P}(s' \mid s, a))$ is again a natural choice, though through only a heuristic that information about the world model might be helpful for the learning process.

For the information gain of generative models, an approximation is to use the predictive gain $\log \mathbb{P}_\theta(s) - \log \mathbb{P}_{\theta'}(s)$. This connects with the count-based model described before [BSO+16]. For $IG(\mathbb{P}(s' \mid s, a))$, it can be approximated by the KL-divergence $d_{\mathrm{KL}}(\mathbb{P}(s' \mid \theta) \parallel \mathbb{P}(s' \mid \theta'))$ between the before-update and the after-update prediction of the world model. This connects with counting methods via errors where d_{KL} can be replaced by other measures as well.

More generally, $IG(\mathbb{P}(s' \mid s, a))$ can be written into $d_{\mathrm{KL}}(\mathbb{P}(\theta \mid \tau, x) \parallel \mathbb{P}(\theta \mid \tau))$, where τ is the trajectory up to the moment and $x = (s, a, r, s')$ is the new experience collected at this step. This term is still intractable, but by decomposing $\mathbb{P}(\theta \mid \tau)$ one can use variational inference to construct a surrogate of the objective. Optimizing this surrogate up to some further approximations and tricks constitute variational information maximizing exploration, or VIME in short [HCD+16].

References

[ACBF02] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[BDM17] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pages 449–458. PMLR, 2017.

[BESK18] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2018.

[BSO+16] Marc G Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, pages 1479–1487, 2016.

[CL11] Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. *Advances in Neural Information Processing Systems*, 24:2249–2257, 2011.

[DRBM18] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[FCRL17] Justin Fu, John D Co-Reyes, and Sergey Levine. Ex2: Exploration with exemplar models for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2574–2584, 2017.

[HCD+16] 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 1117–1125, 2016.

[KN09] J Zico Kolter and Andrew Y Ng. Near-Bayesian exploration in polynomial time. In *International Conference on Machine Learning*, pages 513–520, 2009.

[OBOM17] Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International Conference on machine Learning*, pages 2721–2730, 2017.

[OBPR16] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, pages 4033–4041, 2016.

[RVR18] Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. *Operations Research*, 66(1):230–252, 2018.

[SL08] Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

[TFM+19] Adrien Ali Taïga, William Fedus, Marlos C Machado, Aaron Courville, and Marc G Bellemare. Benchmarking bonus-based exploration methods on the Arcade learning environment. *arXiv preprint arXiv:1908.02388*, 2019.

[THF+17] Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. # Exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, pages 1–18, 2017.

Acknowledgement

This lecture notes partially use material from *Reinforcement learning: An introduction* and CS285 *Deep reinforcement learning* from UC Berkeley.