2. Multi-armed Bandits

Notes

- Evaluative and Instructive Feedback
  - RL uses evaluative feedback, where feedback is given on each individual actions, but not whether it was the best or worst action. Supervised learning uses instructive feedback, where the correct action is explicit indicated, and it is independent of the actions taken.
- K-armed Bandit Problem:
  - Refers to the problem of having to pick a decision out of K different decisions
  - The exact value of an arbitrary action a is given as $q_*(a) \doteq \mathbb{E}[R_t|A_t = a]$ (A_t is the action at time step t, R_t is the corresponding reward).
  - Often, only an estimate of the value can be obtained, which is denoted as $Q_t(a)$, and we want this to be as close to $q_*(a)$ as possible.
  - Exploitation vs Exploration:
    - Exploitation is often associated with greedy actions, where the action with the maximum known reward is picked every time
    - Exploration is often associated with non-greedy actions, where other actions with non-maximum reward is picked. This has the potential to produce greater total reward in the long run. In the short term, the reward is lower but in the long term, there is a high chance that the reward is higher.
    - Not possible to exploit and explore in a single action, thus there is a conflict
    - There are many methods that balances exploration and exploitation. However, most of these methods reply on strong assumptions which can become false in the long term.
- Action-value methods:

- Sample-averaging method:

$$Q_t(a) \doteq \frac{\text{sum of rewards when } a \text{ taken prior to } t}{\text{number of times } a \text{ taken prior to } t} = \frac{\sum_{i=1}^{t-1} R_i \cdot 1_{A_i=a}}{\sum_{t=1}^{t-1} 1_{A_i=a}}$$

  - Method that estimates the value function of an action. As the denominator goes to infinity, $Q_t(a)$ will tend to $q_*(a)$. If denominator is 0, assign it a default value, such as 0.
  - Going through some algebra, the value estimate for a single action can be implemented as $Q_{n+1} = Q_n + \frac{1}{n}(R_n - Q_n)$. This reduces the memory cost of the program. (page 31)
  - This form will occur frequently throughout the book, where it is an error in the estimate: *NewEstimtae ← OldEstimate + StepSize [Target - OldEstimate]*
- Action selection rule:

$$A_t \doteq arg\,max_a Q_t(a)$$

  - This is a greedy selection rule. An alternative is to use a near-greedy rule ϵ-greedy method, where there is a small probability of ϵ that a non-greedy action is selected at random.
  - Advantage of using this near-greedy rule is that as the number of steps increases, $Q_t(a)$ can be obtained for all actions, which in turn converges to their respective $q_*(a)$. This implies that the probability of selection the optimal action converges to more than $1 - \epsilon$, which is near certainty. However, this doesn't tell us much about the practical effectiveness.
- Exponential recency-weighted average: $Q_{n+1} = Q_n + \alpha(R_n - Q_n)$
  - For a non-stationary problem, where important to give more weight to recent rewards than those from the distant past. Here, α is constant.
- The step-size parameter $\alpha_n(a)$ can vary from step-to-step. For the action value to converge, the step size parameter must fulfil two conditions:

$$\sum_{n=1}^{\infty} \alpha_n(a) = \infty, \text{ and } \sum_{n=1}^{\infty} \alpha_n^2(a) < \infty$$

    - The first condition guarantees steps are large enough to overcome initial conditions/random fluctuations. The second condition guarantees steps small enough for convergence.
    - This convergence usually takes a long amount of time and/or needs considerable amount of tuning. This often only exists in theoretical work.
    - The constant step-size parameter does not satisfy the second condition. This is fine since this is what we want for a non-stationary problem.

- Initial values:
    - The methods above are dependent on the initial value of the action-value estimates, which means that they are biased by these initial values. The downside is that the initial values have to be picked manually, but the upside is that prior knowledge can be incorporated into the model.
    - The idea of *optimistic initial values* is the have positive initial values for all of the action-value estimates to be positive. This encourages initial exploration. This often results in low % optimal action initially, but should later outperform non-optimistic initial values. This idea would only work well with a stationary problem
- *Upper confidence bound* action selection:

$$A_t \doteq arg\,max_a \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right]$$

  where $c > 0$ and $N_t(a)$ is the number of times the action a has been selected up till t.

    - The additional term is a measure of the uncertainty/variance of the action-value estimate. By adding this term, this gives a upper bound

to the estimate of the action-value $q_*(a)$.

- Understanding the term: the larger $N_t(a)$ is, the lower the uncertainty of the action-value since more occurrences reduces uncertainty in general. A larger t, with no increase in $N_t(a)$, increases the uncertainty since other actions are being picked.
- Disadvantages: not good at dealing with non-stationary problems, as well as large state spaces.

- Gradient bandit algorithm:
  - Instead of selection rules directly based on action values, we can introduce a numerical preference denoted by $H_t(a)$. The probability of selecting an action, based off $H_t(a)$, is given as a soft-max (Boltzmann) distribution:

$$\Pr\{A_t = a\} \doteq \frac{e^{H_t}(a)}{\sum_{b=1}^{k} e^{H_t(b)}} \doteq \pi_t(a).$$

    There is a natural learning algorithm for soft-max action preferences based on stochastic gradient ascent. At each step, the algorithm is given by:

$$H_{t+1}(a) \doteq H_t(A_t) + \alpha(R_t - \bar{R}_t)(1 - \pi_t(A_t)),$$
$$H_{t+1}(a) \doteq H_t(a) - \alpha(R_t - \bar{R}_t)\pi_t(a) \text{ for all } a \neq A_t$$

    where $\bar{R}_t$ is the average of the rewards up to time t, and can be thought of as tbe reward baseline.
  - If the current reward is higher than the baseline, the current action will increase in probability, whilst the other actions decrease in probability, and vice versa for when the current reward is lower than the baseline.
  - Mathematical explanation given on page 38.

- Associative search / Contextual bandits:
  - Up till now, everything has been non-associative tasks, where there is no association between different actions and situations.

    - A general reinforcement learning task involves more than one situation, and the goal is to learn a policy: a mapping from situations to optimal actions for particular situations.
    - An associative search task is a bridge between non-associative tasks and general RL tasks. An associative search task involves trial-and-error learning to search for optimal actions, and also have association of these optimal actions to situations. The thing that distinguishes an associative search task and a general RL task is that in a full RL problem, a specific action has the ability to affect the rewards in future situations, whilst an associative search task only deals with immediate rewards.
- Additional comments:
    - A parameter study can be created to find the best performing parameter for each algorithm. This often involves a plot of the performance against parameter value.
    - A *Gittins index* is another action value that can help balance exploration and exploitation.This is a Bayesian method, and it requires complete knowledge of the prior distribution of possible problems, which is generally not available.
    - *Thomson sampling* is the idea of selecting the best action based on its posterior probability of being the best action.
    - In a Bayesian approach, the information state of the problem can be calculated. For example, the possible actions and rewards for a horizon of 1000 steps can be calculated, and the best possible chain of events can be selected. However, this tree of possibility increases exponentially, making this method unfeasible.

Exercises

2.1 The ϵ here is quite high at 0.5, which means that the probability of selecting the greedy action is also 0.5. The selection rule here disregards the value of the action, since there is equal probability in selecting either of the two actions.

2.2 Given the action-reward sequence (A_i, R_i) of (1, -1), (2, 1), (2, -2), (2, 2), (3, 0) for a ϵ-greedy method. Exploration occurs on the 4th step, given that the

sample-averaging method gives a $Q_4(2) = -0.5$, whilst the rest of the value estimates, other than 1, give 0, which is more than -0.5. Exploration also occurs on the 5th step since $Q_5(2) = 0.5$, which is the highest value estimate so far.

2.3 Running the agent for 10,000 timesteps, the average reward for the $\epsilon = 0.01$ surpasses the one for $\epsilon = 0.1$. However, the percentage of optimal action does not.

2.4 Given that α_n is not constant → $Q_{n+1} = Q_n + \alpha_n(R_n - Q_n)$

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha_n(R_n - Q_n) \\
&= \alpha_n R_n + (1 - \alpha_n)Q_n \\
&= \alpha_n R_n + (1 - \alpha_n)(\alpha_{n-1}R_{n-1} + (1 - \alpha_{n-1})Q_{n-1}) \\
&= \alpha_n R_n + (1 - \alpha_n)\alpha_{n-1}R_{n-1} + (1 - \alpha_n)(1 - \alpha_{n-1})Q_{n-1} \\
&= \left[\prod_{i=1}^{n}(1 - \alpha_i)\right]Q_1 + \sum_{i=1}^{n}\left[\prod_{j=i}^{n}(1 - \alpha_{j+1})\right]\alpha_i R_i
\end{aligned}
$$

2.5 With a non-stationary problem, and with $\epsilon = 0.1$, the constant step-size parameter outperforms the sample averages method by a significant amount. After 10,000 steps, the constant step-size parameter is able to reach an average reward of 1.3, whilst the sample average method was able to reach an average reward of only 1.0. In terms of % optimal action, 80% and 50% was obtained respectively.

2.6 With optimistic initial values, the action-value estimates are larger than what the actual values actually are. This means that making any action, regardless of how optimal the action is, would result in the action-value estimate decreasing. This also means that the agent is extremely exploratory in the initial stages. This can explain the large oscillations - with non-optimal actions not touched and the optimal action being chosen, the action-value estimate of the non-optimal actions will be higher than the optimal action. This leads to spikes and troughs in the % optimal action as the agent has to deal with non-optimal actions with overly optimistic action-value estimates.

2.7 Method to avoid bias by initial values of constant step-size while retaining their advantage on non-stationary problems: Use step size of $\beta_n \doteq \alpha/\bar{o}_n$, where $\bar{o} \doteq \bar{o}_{n-1} + \alpha(1 - \bar{o}_{n-1})$ for $n \geq 0$, with $\bar{o}_0 \doteq 0$.

$$
\begin{aligned}
Q_{n+1} &= Q_n + \frac{\alpha}{\bar{o}_n}(R_n - Q_n) \\
&= \left[\prod_{i=1}^{n}(1-\frac{\alpha}{\bar{o}_i})\right]Q_1 + \sum_{i=1}^{n}\left[\prod_{j=i+1}^{n}(1-\frac{\alpha}{\bar{o}_j})\right]\frac{\alpha}{\bar{o}_i}R_i \\
&= 0 + \sum_{i=1}^{n}\left[\prod_{j=i+1}^{n}(1-\frac{\alpha}{\bar{o}_j})\right]\frac{\alpha}{\bar{o}_i}R_i, \text{ since } \bar{o}_1 = \alpha \\
&= \sum_{i=1}^{n}\left[\prod_{j=i+1}^{n}\frac{\bar{o}_{j-1}(1-\alpha)}{\bar{o}_j}\right]\frac{\alpha}{\bar{o}_i}R_i \\
&= \sum_{i=1}^{n}\frac{\alpha}{\bar{o}_n}(1-\alpha)^{n-i}R_i.
\end{aligned}
$$

This is identical to the exponential recency-weighted average, with a different step-size parameter.

2.8 In a 10-armed bandit problem, the UCB selection rule will select each of the arms exactly once, since it needs to go around to remove the infinite UCB values caused by $N_t(a) = 0$ for each arm. After the first 10 steps, all of the UCB values are finite, but still have high uncertainties. At the 11th step, the agent would then select the best performing arm out of the 10 arms, which results in a spike of he reward. However, by selecting the best performing arm, the uncertainty term decreases for that arm, which likely causes the UCB value to decrease. The UCB selection rule would mean that other arms with higher uncertainties, and probably lower rewards, will be selected. This can explain the decrease in reward after the spike at the 11th step.

2.9 Case of two actions:

$$
\Pr\{A_t = a\} \doteq \frac{e^{H_t(a)}}{e^{H_t(a)} + e^{H_t(b)}} = \frac{1}{1 + e^{-(H_t(a) - H_t(b))}}.
$$

The last term is a logistic function.

2.10 In the case that we are unable to tell which case we face at any step, selecting solely action 1 or 2 gives an expected reward of 50. Thus, one can just stick with either of the actions.

In the case where the case is observable, one would select action 2 in case A and action 1 in case B, with expected reward of 55. By using a suitable action-value algorithm (eg. sample averaging, since it is stationary) for each case, one will be able to learn to maximise the reward.

2.11 Varying ϵ for the sample average and fixed step methods for a non-stationary problem. For both methods, there is a peak at $\epsilon = 2^{-6}$. We can also see that the fixed step gives a better average reward for all ϵ values.