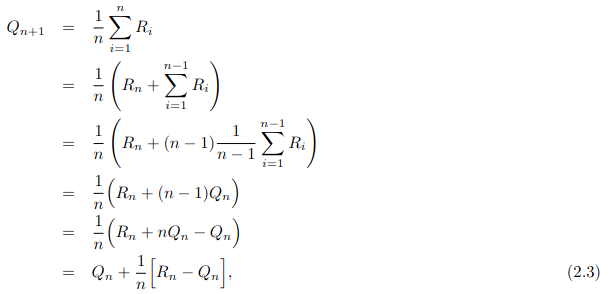
Taylor Bergeron

**Assignment #1**

1. What is the benefit of incremental update for action-value function, versus non-incremental?

The incremental update requires less memory, as it only needs memory for Qn and n. Qn is the current estimation of the reward after n time steps. The computation for each new reward, listed in textbook Equation 2.3, is not expensive, as it only requires three arithmatic operations.



1. Why balancing exploration/exploitation is important? Name a few ways to balance exploration/exploitation in multi-armed bandits.

Balancing exploration/exploitation is important because “exploitation is the right thing to do to maximize the expect reward on one step, but exploration may produce the greater total reward in the long run.” (p 26) They both have their advantages, so combining the two reaps the benefits of both. Because it is not possible to explore and exploit during the same action, there is a tradeoff that must occur.

There are many sophisticated ways to balance exploitation/exploitation, but the reading expands on action-value methods that distinctly choose between greedy or e-greedy actions. If an algorithm chooses only greedy actions, then the algorithm is only choosing actions with the highest estimated reward. There is no exploration, only exploitation. If e-greedy is implemented, a tradeoff occurs between exploration and exploitation. The bandit algorithm will choose the exploitative action with the highest estimated reward (greedy) with probability 1-e, and will choose an exploration random action with probability e.

The reading also delves into Upper Confidence Bound (UCB), which allows for the optimality of e-greedy actions to be considered. UCB considers how optimal an action using the Equation 2.10 in the texbook. If an action is chosen, the uncertainty term of the equation, the square root is reduced, as the action has been “tested.” All other actions’ square root term increases. In question 4 UCB is explained more.

1. In the equation

what is the target ?

The target is a desirable direction that the algorithm should move in, and in concrete terms, the target is the reward after doing an action at the nth time step. The reward is the actual reward for an action, not the estimated reward for the action.

1. What is the purpose of using Upper Confidence Bound (UCB)?

It aids in choosing an action that doesn’t fall into one of the distinct two categories of greedy and e-greedy actions. Greedy actions evaluate to the best at the current time step, but some actions may overall be better. The e-greedy action selection forces other non-greedy actions to be attempted. There is no preference for actions that are close to greedy or uncertain. UCB allows for the non-greedy actions to be chosen for their potential to be optimal. UCB considers “how close their estimates are to being maximal and the uncertainties in those estimates” (p 35).



Each action is evaluated for how high in value the reward estimates are and the uncertainty in the estimates. The main idea behind UCB is that the action selection square root term is a measure of the variance in the estimation of a certain action’s value. Each time an action is chosen, the uncertainty is reduced, as the action is “tested” again.

For the remaining actions that were not chosen, the uncertainty increases. This occurs as Nt(a), the number of times an action is chosen, does not increase. t, the number of time/timesteps since the program begin, has increased. Since t is in the numerator, and Nt(a) in the denominator, the uncertainty of action a increases if a is not chosen.

1. Why do you think in Gradient Bandit Algorithm, we defined a soft-max distribution to choose the actions from, rather than just choosing action?

Because the Gradient Bandit Algorithm is focused on choosing an action with the highest probability of being chosen. Soft-max gives us the probability distribution of the actions from which we can choose the action.

1. Read the article below and summarize what you learned in a paragraph: <https://www.spotx.tv/resources/blog/developer-blog/introduction-to-multi-armed-bandits-with-applications-in-digital-advertising/>

I learned about the interesting approach of multi-armed bandits (MAB) for advertising. I learned about the MAB approach getting stuck in a sub-optimal choice due to the simple tradeoff between exploration and exploration, due to the action-value algorithm.

To solve this, I learned about Thompson sampling, also known as Baysian bandits. The concept of Thompson sampling revolves around having the bandit maintain beliefs about where it thinks the true click-through-rate, CTR of each add is, rather than tracking the empirical CTR after each trial.

I learned about Regret and how it measures the “goodness” of a particular strategy. It is the simple concept of regretting missing out on something by not choosing the best option.