K Armed Bandit Notes

Colin Manko[1*]

October 6, 2020

Abstract

Quick overview of the neccessary formulas and concepts for K Armed Bandits. I'm reading Reinforcement Learning by Sutton and Barton.

Introduction

The K Armed Bandit can model a situation where an agent faces k actions, $a \in A$, each with a distribution of possible rewards, $r \in R$, and their corresponding probabilities of occurring, $p(r_t|a_t = a)$. The goal of this situation is to maximize the summation of rewards through time step T, $\sum_{t=1}^{T} R_t$. We do not transition to another state after selecting an action. Instead, we repeatedly face the same situation T times and we keep track of a_n, the number of times action a was selected. As $a_n \rightarrow \infty$, we can cacluate an accurate reward expectation, this assumes the probability ditributions over $r \in R|A$ are stationary, or non-changing. In truth, stationary refers more to having a non-changing $q(*)$, or the true expectation. Theoretically, changing the probabilities associated with the rewards for a given action could result in the same $q(*)$.

The reward expectation for an action is defined as follows:

$$q_*(a) \doteq \mathbb{E}[R_t|A_t = a]$$

Incremental Updates.

Q_n refers to the value estimate after n occurances of a.

$$Q_n = \frac{R_1 + R_2 + R_3 + \ldots + R_{n-1}}{n-1}$$

$$Q_{n+1} = \frac{1}{n}\sum_{i=1}^{n} R_i$$

[*] Corresponding author – colin@colinmanko.com

$$Q_{n+1} = \frac{1}{n}(R_n + \sum_{i=1}^{n-1} R_i)$$

$$Q_{n+1} = \frac{1}{n}(R_n + (n-1) \times \frac{1}{n-1}\sum_{i=1}^{n-1} R_i)$$

$$Q_{n+1} = \frac{1}{n}(R_n + (n-1)Q_n)$$

$$Q_{n+1} = Q_n + \frac{1}{n}(R_n - Q_n)$$

Constant Step Sizes.

Instead of the $\frac{1}{n}$, we can use a constant step size parameter named α. This means that the step size is non-decreasing. Whereas, the above is good for stationary problems (due to convergence properties), the constant step size can be good for non-stationary problems, although it is not gauranteed to converge.

$$\alpha \in [0, 1)$$

$$Q_{n+1} = Q_n + \alpha(R_n - Q_n)$$
$$Q_{n+1} = Q_n + \alpha R_n - \alpha Q_n$$
$$Q_{n+1} = (1-\alpha)Q_n + \alpha R_n$$
$$Q_{n+1} = \alpha R_n + (1-\alpha)Q_n$$
$$Q_{n+1} = \alpha R_n + (1-\alpha)(\alpha R_{n-1} + (1-\alpha)Q_{n-1})$$
$$Q_{n+1} = \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2 Q_{n-1})$$
$$Q_{n+1} = \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2(\alpha R_{n-2} + (1-\alpha)Q_{n-2})$$
$$Q_{n+1} = \alpha R_n + (1-\alpha)\alpha R_{n-1} + (1-\alpha)^2\alpha R_{n-2} + (1-\alpha)^3 Q_{n-2})$$
$$Q_{n+1} = (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

Note that this is exponentially decaying and it is a weighted average as the weights sum to 1.

$$(1-\alpha)^n + \sum_{i=1}^{n} \alpha(1-\alpha) = 1$$

Other Options for Stationary Problems.

You can find others like the $\frac{1}{n}$ that will converge. This is well studied in convergence studies. Ensure both are true:

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

That is to say that other weights for stationarity work.

A few interesting parameters and situations arise.

- You can choose the number of actions, k.
- For each $a \in A$, you can choose the number of rewards, R_n.
- For each $a \in A$ you can choose a distribution of $R_t|A_t = a$ and the associated probability distribution. For now, we assume all time steps have the same probability distribution for each $R_t|A_t = a$.
  - The probability distribution for each reward can be known or can be figured from data.
- These reward distributions, per each action, could be stationary or non-stationary.
- Can have different action-value estimation methods.
  - The sample-average method is the one in the formula listed above, which is good for stationary problems, but not non-stationary problems where you may want to not decrease the step-size parameter as much, as to give more weight to recent values.
- The number of time steps, and whether they are continuous or episodic (this is not as big of a deal in k armed bandit, but it is helpful to start thinking about this).
- Changing the amount of exploration over time.

Of course, having one reward per action would make finding the optimal action trivial. You can still use a reinforcement learning system to learn that, if for example you don't know $q_*(a)$.

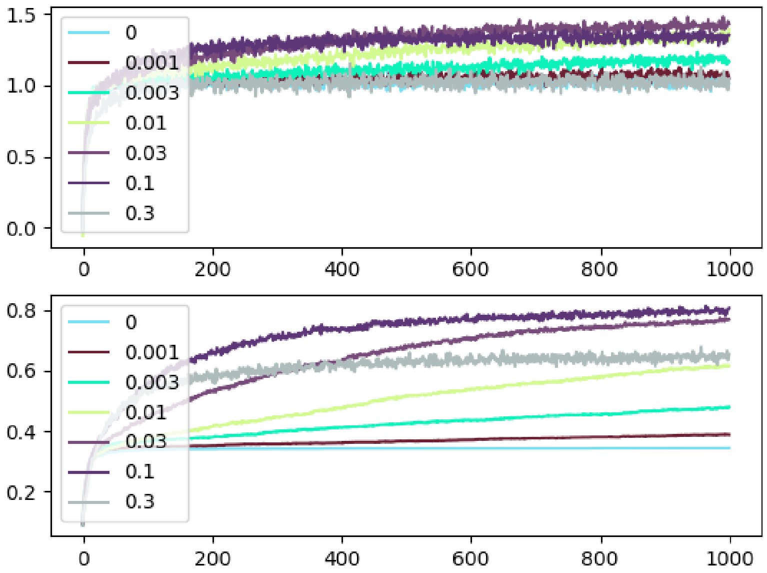
K Armed Test Bed.

To give some figures to the exploration of these concepts, I have implemented my own K armed bandit test bed. In all, there are 1000 iterations 2000 times. I may change the distributions of rewards and whether those distributions are stationary or non-stationary. I'll specifiy for each graph discussed.

Exploration.

In order to learn this, your reinforcement system cannot always choose the optimal action. Meaning, the system also has to *explore* in order to find the optimal actions. Balancing *exploring* and *exploiting* (taking the *greedy* action) is a delicate process. One needs to be careful that the method of doing so does not violate any conditions surrounding stationarity, etc. Also, given the dynamics of the systems (as detailed in the bullet points above), there could be an optimal configuration of *exploration* and *greedy* behavior.

Here are a few simulations over epsilon greedy versus just greedy:



The above k armed bandit had 10 actions with rewards randomly generated from a normal distribution with variance 1 and mean 0. Once found, the reward distributions were given by a normal distribution with variance 1 and mean of the given reward. We find that there is a trade off between the number of iterations and the amount of exploring. Over a long enough timeline it seems as though a small value for exploration would suffice. It is possible to change the amount of exploration over time. It is also possible to optimize uncertainty by using an upperbound of uncertainty, to prioritize actions that have high uncertainty (or really, prioritized by chance of being the optimal value). Of course, this depends on how often and for how long the dynamics change and for how many time steps you expect to train, etc.

A Few Notes.

A note about variance. If variance was 0, you would want a greedy method. A variance of zero means that the rewards are deterministic and simply choosing the highest will result in the optimal value add.

A note about stationarity, if the dynamics were non-stationary, you would want more exploration to pick up on the changing dynamics.

I should also note that when you are exploring, you should choose randomly to break ties with an argmax.

Optimistic Initial Values

Essentially, starting with optimistic initial values means that instead of starting with $Q(a)$ as 0, you start with higher values. This basically means that the system will be encouraged to explore. So far all methods discussed are dependent on initial action-value methods. This also is, but it is well suited in partcular for stationary problems, as over time exploration will occur less.

For example, with the epsilon greedy methods and using a step size of $\frac{1}{n}$ as you would with a stationary problem with the sample average method, you would be biased until all actions are chosen once. Meaning, the highest action-value would have been randomly selected had all action-values been estimated at zero. This also doesn't take into account uncertainty, which we will talk about shortly.

With a constant stepsize (for non-stationary problems) actually there will always be a bias, although it decreases overtime. That is because there is no convergence to an average, the most recent values are always more strongly weighted. So, itital values essentially become parameters you can tweak.

But, they do encourage exploration.

Upper Confidence Bound

This is to capture uncertainty as well. The idea is if an action has not been slected often, there is much uncertainty to whether or not that value actually is the highest value, so this will pick based on the upper bound. The bounds decrease as a_n goes up, meaning that the estimate $Q(a)$ is more certain.

$$A_t = argmax(Q_t(a) + c\sqrt{\frac{ln(t)}{N_t(a)}})$$

Without modification, it is difficult to do this past bandit problems. This is also not great with large state spaces.

This does seem more built for stationary problems, it is possible that the system will become more certain for non-stationary problems, and then the dynamics change.

I can imagine a system whereby you monitor new values agaist old values to assign a probability that we are following a new distribution. In that case, maybe it makes sense to only keep a certain amount in the Q estimate. I guess though that is solved by constantly weighting the newest value more favorably.