Thompson Sampling

Babak Miraftab

Introduction: Suppose that you have 1000 Euro and enter into a casino. There are two **armed bandits**, i.e. slot machine. For those people like me who have not been in a casino so far, there is an arm to pull and bandit takes your money. There are two cases either you win or lose, and your goal is to win as much money as possible. More precisely, each machine will pay a reward of 1 Euro when the outcome is success and 0 Euro when the outcome failure. Each machine pays out concerning a different probability distribution. Also, you do not know these distributions.

Figure 1: Slot machine

You probably come up with several questions. For instance, how many times shall I play for each machine or, more importantly, if I find a machine that turns out to pay out often, then shall I continue or change my machine to the next one so that the next one pays out better than the current one? More precisely, shall I continue with this slot machine to *exploit* or move to the next one to *explore* a better machine? In other words, there is a dilemma *exploit* or *explore*. Two important effects of choosing a machine are the following:

- Knowledge: you gain more data about that machine
- Collect rewards or penalties

Thompson sampling is a reinforcement Learning algorithm to tackle this dilemma with respecting to our original aim which is to maximize our expected reward. Actually Thompson Sampling tries to find balances in parallel finding the best machine(exploration), with exploitation, where you play with the best machine so far (it is known at any point in time).

Thompson Sampling: The Thompson Sampling uses a **Bayesian approach** with the assumptions of a **Beta** prior and **Binomial** likelihood. The Beta prior is a conjugate prior, and combining both will result in a Beta posterior distribution. Each machine has the Beta distribution of rewards and penalties. Because sampling from the Beta distribution will generate probabilities, the machine with the best success to failure ratio will usually have the largest probability, but there'll still be a chance for a lower ratio to generate the largest probability. Beta distribution belongs to a family of continuous probability distributions characterized by two values, denoted by α and β, i.e. $B(\alpha, \beta)$. We note that Beta distribution is always defined between 0 and 1. Let us fit our models from our machines with Beta distributions. In the first machine, if we want to randomly choose a value from the distribution, we would much likely get a value of about 0.7.

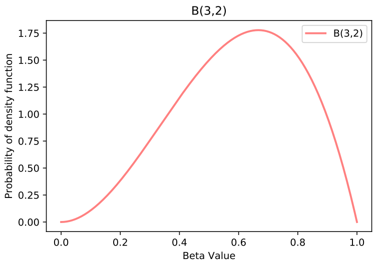

Figure 2: B(3,2)

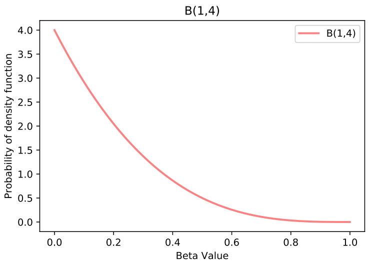

Figure 3: B(1,4)

Let us get back to Thompson Sampling. We benefit from the posterior probability distribution for sampling, and we update the posterior distribution using the observed reward information. For arms that are played fewer times, you get a wide uncertainty, and the sampling allows you to have a chance of picking them. By playing an arm more and more (increasing alpha or beta counts more), the Beta distribution becomes sharper and sharper, and it will eventually converge to the average value.

How does Thompson Sampling work?: Let us get back to our two machines. In the beginning, we do not have any information regarding these machines. So we start randomly choosing one of these two machines and pull the arm. Then we repeat it five more times. Assume that the first machine has 3 successes and 2 failures, and the second machine has only 1 success and 4 failures.

Machine 1	S	S	S	F	F
Machine 2	S	F	F	F	F

The above data can be considered as our environment. We associate each slot machine with the beta distribution $B(1,1)$ and then we take a sample from each machine. Assume that the sample of the first machine is the highest among all. Now we look at our environment. In the first round, we get success, and as rewards, we increment the value of α of the corresponding distribution. So we start the second round. This time suppose that the sample from the second machine is higher than the first one. So we choose the second machine. The environment says that we have a failure this time. So we increment the β value of the corresponding distribution. More generally we follow the following

$$\begin{cases} \alpha = \alpha + 1 & \textit{if we have a success} \\ \beta = \beta + 1 & \textit{if we have a failure} \end{cases} \tag{1}$$

Independently for each slot machine, we have a list of rewards. If we get success, then we add it to the associated reward list. We proceed to the next round. After all iterations, we are going to sum up all rewards. The machine with the highest rewards is the final winner.