EE675A - Introduction to Reinforcement Learning

Lecture 4: UCB Algorithm & Thompson Sampling

18/01/2023

Lecturer: Subrahmanya Swamy Peruru *Scribe: Priya Gole | Sachin Bhadang*

1 Recap

In the last class, we discussed the Successive Elimination Algorithm and analysed instance-independent and instance-dependent regret.

The Successive Elimination algorithm proceeds as follows: The player maintains a set of active arms S. At every round, the player first samples from the rewards of every arm in the active set and computes its LCB and UCB. Then, $\forall a \in A$ the player then eliminates an arm a if $\exists$ arm a' such that $UCB_t(a) \leq LCB_t(a')$.

The instance-dependent bound for the algorithm is of the order $\mathcal{O}\left(\frac{klogT}{\Delta}\right)$.

And the instance-independent bound for the algorithm is of the order $(\sqrt{kT\ logT})$.

2 Upper-Confidence-Bound Algorithm

The mechanics of this algorithm is simple. We choose the arm with the maximum Upper Confidence bound for our next action.
Define $n_t(a)$ to be the number of times arm a has been played up to time t. Define r_t to be the reward we observe at time t.
And let $a(t)$ be the arm chosen at time t. Then, the estimated reward of the arm a at time t is:

$$\overline{\mu_t}(a) = \frac{\sum\limits_{i:a(i)=a} r_i}{n_t(a)}$$

UCB value of arm a at time t is given as:

$$UCB_t(a) = \overline{\mu_t}(a) + \epsilon_t(a)$$

Where,

$$\epsilon_t(a) = \sqrt{\frac{2logT}{n_t(a)}}$$

It's the idea of "optimism under uncertainty" that drives the algorithm. We optimistically choose the arm with the maximum UCB, keeping in mind the best-case scenario of the actual mean of the arm coinciding with UCB. That is how we exploit the data that we have in our hand.

The $\epsilon_t(a)$ is the uncertainty or variance from our estimated mean and it is the value that controls the exploration. If an arm hasn't been tried very often, then $n_t(a)$ will be small, consequently the uncertainty term will be large, making this arm more likely to be selected.

Algorithm 1 UCB Algorithm

```
Require: $k$ arms
Ensure: number of rounds $T \geq k$
  1. Play each arm once.
  2. For $t > k$, play arm $a(t) = \arg\max_{a \in A} UCB_t(a)$. Update $UCB_t(a)$.
```

2.1 Regret Analysis

Since, we chose arm a at time t,

$$UCB_t(a) \geq UCB_t(a^*) \tag{1}$$

where a^* is the best arm. (Note that a is the best arm according to the algorithm at step t, whereas a^* is the actual best arm that gives us the maximum reward. With t large enough, a coincides with a^*)

Using (1), we get,

$$\begin{aligned} \overline{\mu_t}(a) + \epsilon_t(a) &\geq \overline{\mu_t}(a^*) + \epsilon_t(a^*) \\ \epsilon_t(a) - \epsilon_t(a^*) &\geq \overline{\mu_t}(a^*) - \overline{\mu_t}(a) \end{aligned}$$

Therefore, $\Delta(a)$ is bounded by,

$$\begin{aligned} \Delta(a) &= \mu(a^*) - \mu(a) \\ &\leq UCB_t(a^*) - LCB_t(a) \\ &= (\overline{\mu_t}(a^*) + \epsilon_t(a^*)) - (\overline{\mu_t}(a) - \epsilon_t(a)) \\ &= \overline{\mu_t}(a^*) - \overline{\mu_t}(a) + \epsilon_t(a^*) + \epsilon_t(a) \\ &\leq (\epsilon_t(a) - \epsilon_t(a^*)) + (\epsilon_t(a^*) + \epsilon_t(a)) \\ &= 2\epsilon_t(a) \end{aligned}$$

So, we have,

$$
\begin{aligned}
\Delta(a) &\leq 2\epsilon_t(a) \\
\Delta(a) &\leq 2\sqrt{\frac{2logT}{n_t(a)}} \\
\Delta(a) &\leq \mathcal{O}\left(\sqrt{\frac{logT}{n_t(a)}}\right)
\end{aligned}
$$

2.2 Instance dependent bound

We have,

$$\Delta(a) \leq \mathcal{O}\left(\sqrt{\frac{logT}{n_t(a)}}\right)$$

Rewriting this we get,

$$n_t(a) \leq \mathcal{O}\left(\frac{logT}{(\Delta(a))^2}\right)$$

Therefore, expected total regret contributed by arm a is:

$$
\begin{aligned}
R(T;a) &= \Delta(a)n_t(a) \\
&\leq \Delta(a) \times \mathcal{O}\left(\frac{logT}{(\Delta(a))^2}\right) \\
&\leq \mathcal{O}\left(\frac{logT}{\Delta(a)}\right)
\end{aligned}
$$

Hence, Total expected regret

$$
\begin{aligned}
R(t) &= \sum_{a\in A} R(T;a) \\
&\leq \sum_{a\in A} \mathcal{O}\left(\frac{logT}{\Delta(a)}\right) \\
&= \mathcal{O}(logT) \sum_{a\in A} \left(\frac{1}{\Delta(a)}\right) \\
&\leq \mathcal{O}\left(\frac{k\ logT}{\Delta}\right)
\end{aligned}
$$

where $\Delta = min_{a\in A}\ \Delta(a)$.

2.3 Instance independent bound

Let us take some constant ϵ. Divide the arms into two groups using this ϵ as follows:

Case 1: Arms with $\Delta(a) < \epsilon$:
Here the expected regret of arm a is

$$\begin{aligned} R(T;a) &= \Delta(a)n_t(a) \\ &\leq \epsilon n_t(a) \end{aligned}$$

Case 2: Arms with $\Delta(a) \geq \epsilon$:
The expected regret of arm a is

$$\begin{aligned} R(T;a) &= \mathcal{O}\left(\frac{logT}{\Delta(a)}\right) \\ &\leq \mathcal{O}\left(\frac{logT}{\epsilon}\right) \end{aligned}$$

The total regret is the sum of the regret of each group. Hence, the total expected regret is

$$\begin{aligned} R(t) &= \sum_{a\in A} R(T;a) \\ &= \sum_{a:\Delta(a)<\epsilon_t(a)} R(T;a) + \sum_{a:\Delta(a)\geq\epsilon_t(a)} R(T;a) \\ &\leq \epsilon\, T + \mathcal{O}\left(\frac{k\ logT}{\epsilon}\right) \end{aligned}$$

The above inequality is true for any ϵ. So to minimize the RHS, we differentiate wrt ϵ. We get the optimal ϵ as

$$\epsilon^* = \left(\sqrt{\frac{k}{T}\ logT}\right)$$

That gives, the total expected regret bound:

$$R(t) \leq (\sqrt{kT\ logT}$$

3 Bayesian Learning

Till now, to get an idea of what arm is "good" we used point estimates (mean) and confidence intervals (mean ± uncertainty). But to get a better picture, we can look at the probability distribution of the rewards of an arm.

The data that is available to us is through actually playing the arms and recording the rewards. We would like to estimate the probability distribution through this observed data (i.e., we want $P(\mu(a)) = \theta$).
Let us take an example.
We have two arms X and Y. Say, we play arm X with the probability $P(X > Y)$.
1. Let's assume a uniform prior distribution (*unif[0,1]*).
2. Say we play the arms thrice and we get the rewards 1, 1, 0.
3. We want to estimate the Posterior distribution i.e. $P(\mu = \theta \mid \textit{data})$
We know, by Baye's theorem,

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)}$$

Therefore, we have

$$\begin{aligned} P(\mu = \theta \mid data) &= \frac{P(data \mid \mu = \theta)P(\mu = \theta)}{P(data)} \\ &= \frac{P(data \mid \mu = \theta)P(\mu = \theta)}{\sum P(data \mid \mu = \theta)P(\mu = \theta)} \\ &= \frac{\binom{3}{2}\theta^2(1-\theta)}{\int_\theta \binom{3}{2}\theta^2(1-\theta)\,d\theta} \\ &= \frac{\theta^2(1-\theta)}{\int_0^1 \theta^2(1-\theta)\,d\theta} \\ &\propto \theta^2(1-\theta) \\ &= B(3,2) \end{aligned}$$

The probability distribution is said to be beta with parameters α and β if it is of the form

$$B(\alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{c}$$

So, we had the prior distribution which was uniform distribution that can be seen as beta distribution $B(1, 1)$. The posterior distribution that we obtained was also a beta distribution $B(3, 2)$. Such pairs where prior and posterior distribution are of the same form, most likely with different parameters, are called conjugate pairs.

Generalizing the previous example.
1. We have 2 arms with a beta prior probability distribution and Bernoulli reward.
2. The data after n coin tosses is S_n successes and F_n failures.
3. The posterior distribution that we would get is $B(S_n + 1, F_n + 1)$
So, we have the estimated probability distribution.
Let us try to formulate an algorithm exploiting the data that we have. We play arm *a* with probability $P(X > Y)$, and arm *b* otherwise.

Algorithm 2 Algorithm

```
Require: 2 arms (a and b) with beta prior distribution:
    $X = B(\alpha_t(a), \beta_t(a))$
    $Y = B(\alpha_t(b), \beta_t(b))$
    At round $t$
    1. Sample $\theta_t(a)$ from $B(\alpha_t(a), \beta_t(a))$ and
    Sample $\theta_t(b)$ from $B(\alpha_t(b), \beta_t(b))$
    2. If $\theta_t(a) > \theta_t(b)$ play arm a. Else play arm b.
```

4 Thompson Sampling

Assume that we have a Bernoulli Multi Arm Bandit instance.
That is, the reward that we get when arm a is pulled corresponds to some Bernoulli distribution.
Also, for every arm a, the prior distribution (at $t = 0$) is a beta distribution $B(\alpha_0, \beta_0)$.

The algorithm for Bernoulli Multi Arm Bandit is as follows:

Algorithm 3 Bernoulli Thompson Sampling

```
Require: $\forall a \in A$, the prior distribution of arm $a$ at $t = 0$, $B(\alpha_0(a), \beta_0(a))$.
    At round $t$
    1. For each arm $a$: Sample $\theta_t(a)$ from $B(\alpha_{t-1}(b), \beta_{t-1}(b))$
    2. Play arm $a(t) = \arg\max_{a \in A} \theta_t(a)$
    3. Update the posterior of the arm $a(t)$ based on the reward you got in that round.
```

For more details on these topics, refer to [1] and [2].

References

[1] A. Slivkins. Introduction to multi-armed bandits, 2019.

[2] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.