AIMS CDT: Online Learning and Multi-Armed Bandits

Mrinank Sharma

1 Introduction

In the Multi-Armed Bandit (MAB) problem, an agent (or algorithm) is required make decisions under uncertainty over a number of sequential steps in order to maximise a reward function, or equivalently, to minimise a loss function. This type of problem is found often in the real-world, including the ad selection problem, where an algorithm must determine which ad to display to a user, as well as in medical trials, where a doctor must decide which treatment to provide a patient.

Here, we focus on *stochastic bandits* for which the rewards provided by each arm are drawn according to some underlying distribution for each arm. We consider different algorithms to solve this problem, and empirically compare them their performance in terms of *regret*.

2 Preliminaries

2.1 Problem Definition

The algorithm chooses between K different actions (also refered to as arms) across T rounds. For the stochastic bandit problem, rewards provided by arm $a \in \{1,\dots,K\}$ are drawn according to some fixed (but unknown) distribution, $\mathcal{D}_a$ (if the distributions were known, the problem would be trivial). Denote the chosen action at round t as a_t. Then, the probability of reward at this timestep, r_t, is

$$\Pr(r_t|a_t = a) = \mathcal{D}_a(r_t). \tag{1}$$

The regret at round T is defined as:

$$R(T) \triangleq \mu^* \cdot T - \sum_{t=1}^{T} \mu(a_t) \tag{2}$$

where

$$\mu(a_t) = \mathbb{E}_{r\sim D_{a_t}}[r_t], \tag{3}$$

and $\mu^* = \max_a \mu(a)$. The maximiser of this will be denoted as a^*. The regret compares the performance of the algorithm with playing the best possible action (a^*) at each round.

Many of the algorithms that we will consider will require a notion of a *confidence radius*, representing a region around the empirical means which we believe that the true mean will lie in with high probability. Typically, the confidence radius is chosen to be of the form:

$$r(a) = \sqrt{\frac{2\log T}{n_a}} \tag{4}$$

where n_a is the number of times which arm a has been played. This choice is made so that:

$$\Pr[|\overline{r_a} - \mu_a| > r(a)] \leq \frac{2}{T^4} \tag{5}$$

where $\overline{r_a}$ is the sample mean of the rewards provided by arm a. Intuitvely, we require that the probability of the sample mean being far from the true mean (which we will refer to as the 'bad event') decays quickly enough, and this motivates this choice. Note that the above bound is typically derived using Hoeffding's inequality, which only applies for bounded random variables.

Here, we will consider the following algorithms: (1) Explore-first (2) Upper Confidence Bound (UCB) (3) ϵ-Greedy (4) Thompson Sampling. Note that this is an (approximate) Bayesian approach, which requires a prior over arm parameters to be defined. We will also focus on the cases where $\mathcal{D}_a$ are Bernoulli or Gaussian random variables. Please see Slivkins (2019) for an overview of these algorithms.

2.2 Confidence Radius for Gaussian RVs

The proof that the probability of a bad event decays with T^4 does not hold for Gaussian random variables, because these random variables have infinite support. However, we now prove a similar tail bound for these variables and thus derive a confidence radius, given that the probability of the bad event decays quickly enough.

Let $r_a \sim \mathcal{N}(\mu_a, \sigma^2)$, and suppose that $\{r_a^{(1)}, \ldots, r_a^{(n_a)}\}$ are samples from this distribution. We need to consider:

$$\Pr\Big[\Big|\underbrace{\frac{1}{n_a}\sum_{i=1}^{n_a} r_a^{(i)} - \mu_a}_{Z}\Big| > t\Big]. \tag{6}$$

Using the properties of Gaussian RVs, $Z \sim \mathcal{N}(0, \sigma^2/n_a)$. Then,

$$\begin{aligned}
\Pr[|Z| > t] &= 2\Pr[Z > t] && \text{(Symmetry)}\\
&= 2\Pr[\exp(\lambda Z) > \exp(\lambda t)] && \text{(For } \lambda > 0\text{)}\\
&\leq \frac{2\overbrace{\mathbb{E}[\exp(\lambda Z)]}^{\text{MFG}_Z(\lambda)}}{\exp(\lambda t)} && \text{(Markov's Inequality)}\\
&= 2\ \exp(\frac{\sigma^2\lambda^2}{2n_a} - \lambda t), && (7)
\end{aligned}$$

by subsituting the closed form expression for the Moment Generating Function (MGF) for Gaussian RVs. Then, minimising the upper bound over lambda yields:

$$\Pr[|Z| > t] \leq 2\exp(\frac{-n}{2\sigma^2}t^2). \tag{8}$$

Thus, setting

$$r_a(t) = \sqrt{\frac{8\sigma^2}{n}\log T}, \tag{9}$$

gives the probability of the bad event decaying with T^4, even though Gaussian RVs have infinite support. It is pleasing that this equation scales linearly with σ, as we expect that larger σ should give a larger confidence radius.

2.3 Thompson Sampling for Gaussian RVs

In order to implement Thompson Sampling, we need to be able to compute the posterior over arm parameters. We will use conjugate prior distributions, thus using a Beta and Gaussian prior for the Bernoulli and Gaussian arms respectively. The update equations for the Bernoulli arm posterior is found in Chapelle and Li (2011), so all that remains is to derive the update equations for Gaussian arms.

Consider arm a. The rewards from this arm have a Gaussian distribution i.e.,

$$r_a \sim \mathcal{N}(\mu_a, \sigma_a^2). \tag{10}$$

We will consider σ_a^2 to be a known parameter, whilst μ_a is unknown. We place a Gaussian prior on the mean:

$$p(\mu_a) = \mathcal{N}(\mu_a|\hat{\mu}, \hat{\sigma}^2). \tag{11}$$

The posterior will be updated in a sequential way (i.e. using *online learning*). Suppose that $\{r_a^{(1)}, \ldots, r_a^{(n_a)}\}$ are samples from this arm. Then, by Bayes' rule:

$$p(\mu_a|r_a^{(1)}, \ldots, r_a^{(n_a)}) \propto \underbrace{p(\mu|r_a^{(1)}, \ldots, r_a^{(n_a-1)})}_{\text{"prior"}}\underbrace{p(r_a^{(n_a)}|\mu_a)}_{\text{likelihood}}, \tag{12}$$

where the first term is effectively acting as a prior. Suppose that this effective prior is Gaussian with mean μ_p and variance σ_p^2. Then, the posterior is also Gaussian with the following form.

$$p(\mu_a|r_a^{(1)}, \ldots, r_a^{(n_a)}) = \mathcal{N}\Big(\mu\Big|\ \frac{\mu_p\sigma_a^2 + r_a^{(n_a)}\sigma_p^2}{\sigma_a^2 + \sigma_p^2}, \frac{\sigma_p^2\sigma_a^2}{\sigma_a^2 + \sigma_p^2}\Big) \tag{13}$$

The above equation provides a way to update the posterior on μ_a as additional observations of the reward arrive. Then, the normal Thompson sampling approach can be applied.

3 Simulations

3.1 Bernoulli Arms

We draw $K = 5$ Bernoulli arms, sampling the mean parameter according to Beta$(1, 1)$. This is then set to be the prior distribution on the mean of each Bernoulli distribution. We implement the explore-first algorithm setting $N = T^{2/3}(\log T)^{1/3}$, where N is the number of times each arm is explored before the exploit stage of the algorithm. This is the setting used to derive regret bounds in Slivkins (2019). Additionally, we use $\epsilon = t^{-1/3}(K \log t)^{1/3}$ for the ϵ-greedy algorithm, which is again the value used to derive regret bounds.

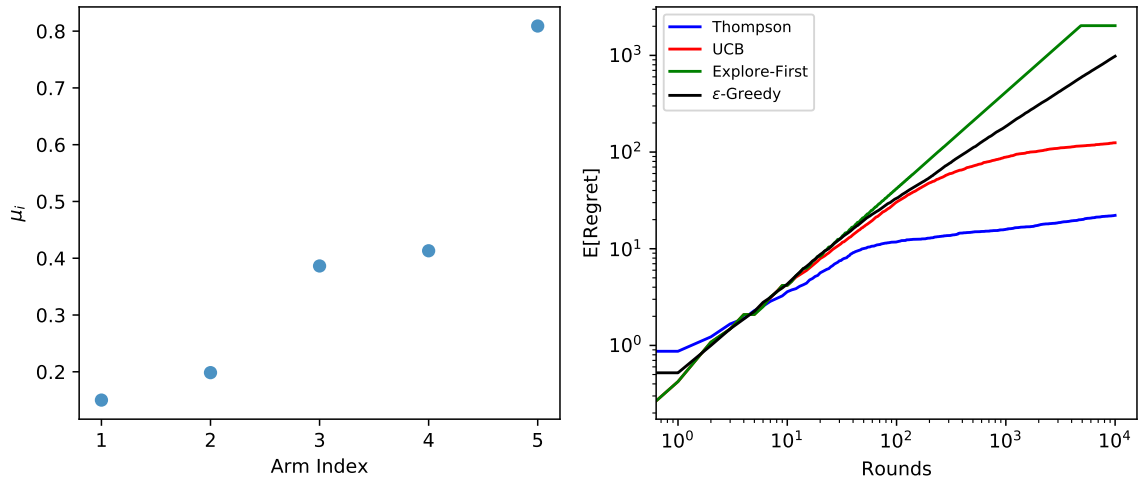

Figure 1: Simulation results using Bernoulli arms. Left: distribution of mean values of each arm. Right: expected cumulative regret for the different algorithms considered here. Shaded region shows the empiricial standard deviation producted across 5 random seeds.

3.2 Gaussian Arms

We draw $K = 10$ Gaussian arms, sampling the mean parameter according to $\mathcal{N}(0.5, 1)$. The variance of the each arm is set to be $\sigma^2 = 0.1^2$, and this is the same across all arms. We consider setting the prior distribution on the mean value of each arm to be the "correct" prior (i.e. the distribution according to which the mean was sampled) as well as $\mathcal{N}(1, 0.25^2)$ (labelled "A") and $\mathcal{N}(-1, 0.25^2)$ (labelled "B"). We use the same configurations for parameters of the explore-first algorithm and ϵ-Greedy algorithm as for the Bernoulli arms, but we use Eq. (9) as the confidence range for the UCB algorithm.

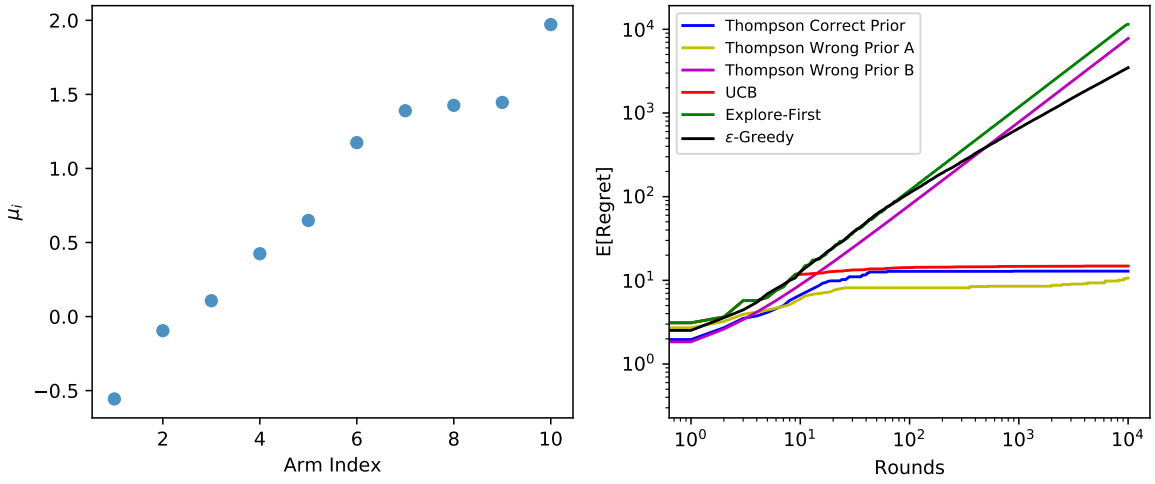

Figure 2: Simulation results using Gaussiani arms. Left: distribution of mean values of each arm. Right: expected cumulative regret for the different algorithms considered here. Shaded region shows the empiricial standard deviation produced across 5 random seeds.

Please see https://github.com/MrinankSharma/AIMSBandits for full code listings.

4 Discussion

Inspecting Fig. 1, the performance of the considered algorithms is similar for the first 10 rounds. After this point, Thompson sampling seems to perform best, following by UCB, ϵ-Greedy and the explore-first algorithm. It is worth noting that the additional regret after the exploration round of the explore-first algorithm is zero i.e., the algorithm does eventually find the best arm. Theoretically, UCB achieves a regret upper bound of $\mathcal{O}(\sqrt{Kt\log T})$, corresponding to a slope of $1/2$ on the log-log plot whilst ϵ-Greedy achieves a bound of $t^{2/3}\mathcal{O}(K\log t)^{1/3}$, which should correspond to a slope of $2/3$ (provided the log term is small enough) (Slivkins, 2019). The bound for the explore-first algorithm does not hold for all t. The UCB bound is not tight since as t becomes large, eventually the right arm will be picked. The slopes do look approximately linear between rounds 1 and 100, but the slopes for UCB and ϵ-Greedy are similar with value approximately 0.8. The reason for this is unknown.

Inspecting Fig. 2, corresponding to the Gaussian arms, UCB and Thompson Sampling (for the “correct" prior and prior A) perform significantly better than the other algorithms considered. It is clear that the prior used has a significant effect on the performance of Thompson sampling, but it is suprising that the best performance (in terms of regret) is given by prior *A*, which overestimates the true prior mean and underestimates the standard deviation. It is not entirely clear why this occurs, but it is worth noting that using the true prior gives a smaller empirical standard deviation compared to both misspecified priors. UCB using the modified confidence region also performs very close to Thompson sampling and this is also far easier to implement. Using prior B results in incredibly poor performance, in fact, worse than all algorithms other than the explore-first algorithm for a larger number of rounds, which is concerning. This has implications in practice; perhaps if there is no true prior information, one is best served by using an non Bayesian approach rather than applying a weak uninformative prior, as using a poor prior can actually have a large adverse affect upon Thompson sampling.

5 Conclusions

By bounding the tail probability for Gaussian random variables, we derive an alternative formula for the confidence radius of a Gaussian arm, and then derive Thompson sampling for these arms. We run simulations on both Bernoulli and Gaussian arms, comparing Thompson sampling with other standard algorithms. The posteriors for Thompson sampling are updated using *online learning*. In the simulations considered, Thompson sampling, provided that the prior on arm parameters is specified correctly, outperforms other algorithms in terms of expected cumulative regret. Of the other algorithms, UCB also performs very well, and in cases whether compute power is incredibly limited, UCB may be a better option than Thompson sampling. It is worth nothing that the prior used for Thompson sampling has a large effect on the performance of the algorithm, and in cases where there is truly no prior information, a more simple algorithm, such as UCB, is more appropriate as choosing a poor prior results in incredibly poor performance. It is suprising that UCB can perform as well as a Bayesian approach.

References

Aleksandrs Slivkins. Introduction to multi-armed bandits. *CoRR*, abs/1904.07272, 2019. URL http://arxiv.org/abs/1904.07272.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. pages 2249–2257, 2011. URL http://papers.nips.cc/paper/4321-an-empirical-evaluation-of-thompson-sampling.pdf.