Introduction of Stochastic Bandits

MA5249 Presentation I

Dick Jessen William A0200677H

NUS

November 2021

Outline

Problem Definition

Definition (Stochastic Bandits)

- The player (our algorithm) plays a game.
- The game has T rounds.
- Each round, there are K actions possible.
- Each action gives a reward depending on the action.
- The goal of the player is to reap as much reward as possible.

In this presentation, we assume the following.

- The algorithm only get to see the reward gained from the chosen action every round.
- The reward for each action a is independent and identically distributed (IID) from the distribution D_a.
- Rewards are bounded in the interval $[0, 1]$.

Some Notations

Definition (Mean Reward Vector)

The mean reward vector $\mu \in [0,1]^K$ is defined as $\mu(a) = \mathbb{E}[D_a]$.

Suppose that A denotes the set of all arms.

Definition (Best Mean Reward)

The best mean reward is defined as $\mu^* = \max_{a \in A} \mu(a)$.

Definition (Gap of an Arm, Optimal Arm)

The gap of any arm a is defined as $\Delta(a) = \mu^* - \mu(a)$. We call an arm optimal if the gap is 0.

Regret Value

Definition (Regret of an Algorithm)

For an algorithm, denote $R(T)$ as the regret of the algorithm at round T. In other words, $R(T) = \mu^* \cdot T - \sum_{t=1}^{T} \mu(a_t)$.

- This quantifies how much the algorithm is missing out from the best possible move (best arm).
- Because our algorithm is stochastic, we instead interested on the expected regret, $\mathbb{E}(R(T))$.
- Our goal is to minimize this quantity.

Explore-First Algorithm

Our idea is to spend some time trying all arms, then choose the best one and go with it for the rest of the round.

Algorithm Explore-First

1: Try each arm N times
2: $\hat{a} \leftarrow$ arm with highest average
3: Play $\hat{a}$ until end of round.

We will choose the suitable N in terms of T and K.

Regret Bound of Explore-First

Theorem

When we choose $N = (T/K)^{2/3} O(\log T)^{1/3}$, *we achieve expected regret* $\mathbb{E}[R(T)] \leq T^{2/3} O(K \log T)^{1/3}$.

Why Explore-First is not Enough

- There are several problems to our first algorithm.
- If there are a lot of arms with a large gap, we waste a lot of time checking these bad arms.
- Hence, we should spread our exploration phase (trying arms randomly) uniformly over time.
- We can formalize this by Epsilon-Greedy Algorithm.

Epsilon-Greedy Algorithm

Algorithm Epsilon-Greedy

for each round $t = 1, 2, \cdot$ **do**
    Toss a coin with success probability ϵ_t.
    if success **then**
        Explore: Choose an arm uniformly at random.
    else
        Exploit: Choose the arm with highest average so far.
    end if
end for

Here, we can explore even in the later rounds by setting the parameter ϵ_t.

Regret Bound for Epsilon-Greedy

Theorem

Epsilon-Greedy algorithm with $\epsilon_t = t^{-1/3} \cdot (K \log t)^{1/3}$ *has regret bound* $\mathbb{E}[R(t)] \leq t^{2/3} O(K \log t)^{1/3}$ *for each round* t.

Stronger Algorithms

In this section, we will examine two algorithms with much better bounds.

Definition (Some More Notations)

Fix a time t and arm a. Denote $n_t(a)$ as the number of rounds before t such that arm a is chosen, and $\overline{\mu}_t(a)$ be the average reward.

Now, we can define UCB and LCB for an arm a and time t.

Definition (UCB and LCB)

Define the UCB (upper confidence bound) and LCB (lower confidence bound) as $UCB_t(a) = \overline{\mu}_t(a) + r_t(a)$ and $LCB_t(a) = \overline{\mu}_t(a) - r_t(a)$.

Successive Elimination

Now, we write the idea of high-confidence elimination algorithm.

Algorithm Successive Elimination

1: All arms are set to active
2: **loop**
3: Play each arm once.
4: Deactivate all arms a such that if t is the current round, there exists another arm a' such that $UCB_t(a) < LCB_t(a')$ deactivation rule.
5: **end loop**

Intuitively, if we are sure that one arm is mostly outclassed by other arm, we will not use that arm anymore.

Regret Bound for Successive Elimination

Theorem

Successive Elimination algorithm achieves regret $\mathbb{E}[R(t)] = O(\sqrt{Kt \log T})$ *for all rounds* $t \leq T$.

We also have the alternative bound here.

Theorem (Alternative Bound)

Successive Elimination algorithm achieves regret

$$\mathbb{E}[R(t)] = O(\log T) \left[\sum_{\text{arms } a \text{ with } \mu(a)<\mu(a^*)} \frac{1}{\mu(a^*) - \mu(a)} \right].$$

UCB1 Algorithm

- The next algorithm have a idea of optimism under uncertainty.
- Assume that each arm is as good as it can possibly be given the observations so far, and choose the best arm based on that.

Algorithm UCB1

1: Try each arm once
2: **for** each round $t = 1, 2, \cdots T$ **do**
3: Pick some arm that maximizes $UCB_t(a)$.
4: **end for**

It is known that UCB1 obtains similar bounds with Successive Elimination.

Incorporating Past Information

- In some cases, some information about the mean reward vector μ may be known to the algorithm before.
- We can encode this information by a constraint on μ or a bayesian prior.
- Such model can allow nontrivial regret bounds that are independent of K, and even works for$K = \infty$.

Constrained μ

- The canonical modelling embeds arms into $\mathbb{R}^d$, for positive integer $d \in \mathbb{N}$.
- Then, μ is a function on a subset of $\mathbb{R}^d$ that maps arms into their mean rewards.
- The constraint is that μ is on some family of well-behaved functions.
- Some example include:
  - Linear functions: $\mu(a) = w \cdot a$ for a fixed but unknown $w \in \mathbb{R}^d$.
  - Concave functions: The set of arms is a convex subset and μ'' exists and negative.
  - Lipschitz functions: $|\mu(a) = \mu(a')| \leq L \cdot ||a - a'||_2$ for all arms a, a' and some constant L.

Bayesian Bandits

- Here, μ is drawn independently from a distribution $\mathbb{P}$, the Bayesian prior.
- We are interested in Bayesian regret: regret in expectation over $\mathbb{P}$.
- This is a special case of Bayesian approach, with many uses in statistics and machine learning.
- $\mathbb{P}$ implicitly defines the family of feasible μ, and specifies whether and to which extent some μ in the family are more likely than the others.
- However, sampling assumption may be very idealized and $\mathbb{P}$ may not fully known to the algorithm.

Literature Review

Here is a quick summary of variants on the problem as well as the progress in it.

- Optimality: Our bounds for Successive Elimination is near optimal. Audibert and Bubeck manages to shave the log T factor to get $O(\sqrt{KT})$. Garvier and Cappe managed to get the lower bound of $\frac{1}{2 \ln 2}$ in 2011 in the alternate form of Successive Elimination.
- Regret for all rounds: Suppose that T is unknown. There are some workarounds.
  - Use UCB1 with $r_t(a) = \sqrt{\frac{2 \log t}{n_t(a)}}$. (Auer 2002).
  - Use a doubling trick to guess T.
- Bandits with predictions: Another goal of a bandit problem is to output a prediction a_t^* after each round t. Good algorithms for total regret, such as UCB1 and Successive Elimination, also works well in this case. However, improvements are possible (Mannoe et. al. 2004, Bubeck 2011).

- Available arms: There are some cases when some arms may not be available. Kleinberg (2008) allows arms to be inactive for some rounds. More extremely, Chakrabarti (2008) examines the case when arms are permanently disabled, with randomized schedule known by the algorithm. Here, UCB1 works well for both cases.
- Partial feedback: The algorithm receive partial feedback. Yue (2009) explores the problem where the only info is which rewards are higher. This has applications in web search optimizations. However, we can determine which states is better by using interleaving (Hoffmann 2016).
- Relaxed benchmark: We ignore the best $\epsilon > 0$ arms. We want to find regret relative to the best remaining arm, which is $\epsilon-$regret. We see that $\frac{1}{\epsilon}$ is an effective number of arms. A result by Kleinberg (2006) achieves a $\epsilon-$regret of $\frac{T^{2/3}\,\text{polylog}\,T}{\epsilon}$ using the doubling trick.