CSC 588: Machine learning theory **Spring 2021**

Lecture 25: Explore-then-commit; UCB algorithm and analysis

Lecturer: Chicheng Zhang *Scribe: Jesse Friedbaum*

1 Stochastic Multi-Armed Bandits

Recall the product recommendation example set up:

For all $t = 1, 2, \dots, T$:

- There are environment draws $\ell_t \in [0,1]^k$ (which are note revealed to the learner). Let $\ell_t(i)$ represent the i^{th} element of ℓ_t.
- The learner selects action $a_t \in \{i\}_{i=1}^k$
- The learner suffers (and sees) loss $\ell_t(a_t)$

Assume: $\forall a$, $(\ell_t(a))_{t=1}^T$ are drawn i.i.d. from a distribution over $[0,1]$ with mean $\ell(a)$.
Our goal is to minimize the cost function

$$\sum_{t=1}^{T} \ell_t(a_t).$$

Example 1. Slot Machines: Suppose there are a row of k slot machines. If you use (pull the arm of) the i^{th} slot machine at time step t you loose $\ell_t(i)$ amount of money. The average money you lose playing the i^{th} slot machine is $\ell(i)$. You wish to play a machine at every time step while loosing the least amount of money.

Example 2. Restaurants: Suppose you have just moved to Tucson and find Tucson contains k restaurants. You eat out once a day and if you visit restaurant i on day j, you regret your decision $\ell_j(i)$ amount. You wish minimize the sum of the regret you experience for your decisions in dining out.

The challenge comes from balancing two goals:

1. **Explore:** Learn which machine is best (has the highest average pay out). Equivalently learn which restaurant you like the best.
2. **Exploit:** Play machines that are rewarding (have good payouts). Equivalently eat good food when you dine out.

An important performance metric will be the psuedo-regret:

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} \ell_t(a_t)\right] - T\ell(a^*)$$

where

$$a^* = \underset{a \in \{i\}_{i=1}^k}{\operatorname{argmin}}\, \ell(a).$$

This represents the difference between the expected loss using our choices and the expected loss from only making the best choice (choosing the best slot machine).

Because ℓ_t is drawn i.i.d. before a_t is chosen we may integrate out our randomness:

$$\mathbb{E}_{\ell_t \sim D}\left[\ell_t(a_t)\right] = \ell(a_t).$$

This allows us to rewrite our formula for pseudo-regret.

Definition 1. *The Psuedo-Regret for a stochastic multi-armed bandit problem is given by:*

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} \left(\ell(a_t) - \ell(a^*)\right)\right]$$

where

$$a^* = \underset{a \in \{i\}_{i=1}^k}{\text{argmin}}\, \ell(a).$$

2 ϵ - Greedy Algorithm

One algorithm to solve this problem is known as the ϵ - Greedy Algorithm. This algorithm consists of picking some percentage of the time ϵ, which we will spend exploring. The remaining $(1-\epsilon)$ amount of time is used to exploit the knowledge we have about the payouts of each arm.

We define some new quantities to help describe this algorithm. We begin with the number of times we have visited arm a at time t:

$$m_t(a) = \sum_{s=1}^{t} \mathbb{I}(a_s = a).$$

We now define the empirical average for arm a at time t:

$$\bar{\ell}_t(a) = \begin{cases} \frac{\sum_{s=1}^{t} \mathbb{I}(a_s = a)\ell_s(a_s)}{m_t(a)} & m_t(a) > 0 \\ 1 & m_t(a) = 0 \end{cases}$$

With these values we may define this algorithm as follows.

Algorithm 1 ϵ - Greedy

```
Initialize $\bar{\ell}_t$ to $\mathbf{0}$
for $t = 1, 2, \cdots, T$ do
  Draw $x \sim$ uniform $([0,1])$
  if $x \le \epsilon$ then
    $a_t \sim$ uniform $\left(\{i\}_{i=1}^k\right)$
  else
    $a_t = \text{argmin}_a\, \bar{\ell}_{t-1}(a)$
  end if
  Receive loss and update $\bar{\ell}_t$.
end for
```

With appropriate tuning of ϵ this algorithm can achieve

$$R_t \le \mathcal{O}\left(k^{\frac{1}{3}} T^{\frac{2}{3}}\right).$$

3 Explore-then-commit

Another algorithm for stochastic multi-armed bandits Explore-then-commit, described below.

Algorithm 2 Explore-then-commit

```
for $t = 1, 2, \cdots, M$ do
    $a_t = ((t-1) \bmod k) + 1$
end for
calculate $\hat{a} = \operatorname{argmin}_a \bar{\ell}_M(a)$
for $t = M+1, M+2, \cdots, T$ do
    $a_t = \hat{a}$
end for
```

We will begin with some **rough** analysis of this algorithm. We first divide the algorithm into two parts: the first for loop is entirely dedicated to exploration and will be called the exploration step; the second for loop is entirely dedicated to exploitation and will be called the exploitation step. We bound the exploration step with the size of the loop: $\mathcal{O}(M)$. For the exploitation step, we recognize that we are simply making the empirically best choice and may therefore use the same tools from ERM analysis. Our error bound on our estimate of $\ell(a)$ is given by $\pm\sqrt{\frac{k}{M}}$. This implies

$$\ell(\hat{a}) \leq \ell(a^*) + \sqrt{\frac{k}{M}}$$

where, as usual, $a^* = \operatorname{argmin}_a \ell(a)$. Then for each step of the exploitation step we incur a maximum regret of $\sqrt{\frac{k}{M}}$ and the regret of the entire exploitation step may be bounded by $\mathcal{O}\left((T-M)\sqrt{\frac{k}{m}}\right)$ or loosing the bound slightly $\mathcal{O}\left(T\sqrt{\frac{k}{m}}\right)$. Adding our two rough bounds for the regret from the exploration step and the exploitation step, we arrive at the bound for total regret:

$$R_T \leq \mathcal{O}\left(M + T\sqrt{\frac{k}{M}}\right).$$

The M value that minimizes this bound is $M = k^{\frac{1}{3}}T^{\frac{2}{3}}$ which leads to a final regret bound of

$$R_T \leq \mathcal{O}\left(k^{\frac{1}{3}}T^{\frac{2}{3}}\right). \tag{1}$$

We will now try to find this bound more rigorously. To do this we will begin with an important lemma.

Lemma 2. *If $k \leq T$ there exists and event E known as the "clean event," which happens with probability*

$$P(E) \geq 1 - \frac{2}{T}$$

in which $\forall a$ and $\forall t$

$$\left|\hat{\ell}_t(a) - \ell_t(a)\right| \leq 2\sqrt{\frac{\ln(T)}{m_t(a)+1}}$$

Note that the +1 in the denominator of the right hand side of the above inequality is only there to deal with the $m_t(a) = 0$ case.

The proof of this lemma is a non-trivial application of Hoeffding's inequality and the union bound. For more information see Slivkin's Introduction to Multi-Armed Bandits. (Chicheng notes after lecture: See Section 1.3.1 of that book for a very careful treatment.)

We are now ready to prove (1).

Proof. The first step in finding our regret will be to split our regret into the regret if the clean event occurs and the regret if the clean event does not occur.

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} (\ell(a_t) - \ell(a^*))\right]$$

$$R_T = P(E^C) \cdot \mathbb{E}\left[\sum_{t=1}^{T} (\ell(a_t) - \ell(a^*))\, \mathbb{I}(E^C)\right] + P(E) \cdot \mathbb{E}\left[\sum_{t=1}^{T} (\ell(a_t) - \ell(a^*))\, \mathbb{I}(E)\right] \tag{2}$$

Recalling both $\ell(a_t) \in [0,1]$ and $\ell(a^*) \in [0,1]$ we find

$$\begin{aligned} P(E^C) \cdot \mathbb{E}\left[\sum_{t=1}^{T} (\ell(a_t) - \ell(a^*))\, \mathbb{I}(E^C)\right] &\leq P(E^C) \cdot \mathbb{E}\left[\sum_{t=1}^{T} (1)\right] \\ &\leq P(E^C) \cdot T \\ &\leq 2 \end{aligned} \tag{3}$$

where the last inequality comes for the application of lemma 2.

We no rewrite

$$\sum_{t=1}^{T} (\ell(a_t) - \ell(a^*)) = \sum_{t=1}^{M} (\ell(a_t) - \ell(a^*)) + \sum_{t=M+1}^{T} (\ell(a_t) - \ell(a^*)) \tag{4}$$

The first sum in the right hand side may be easily bounded

$$\sum_{t=1}^{M} (\ell(a_t) - \ell(a^*)) \leq M \tag{5}$$

The second sum is somewhat more complicated to bound, however applying lemma 2 yields

$$\begin{aligned} \sum_{t=M+1}^{T} (\ell(a_t) - \ell(a^*)) &\leq \sum_{t=M+1}^{T} \left(4\sqrt{\frac{\ln(T)}{\frac{M}{k}+1}}\right) \\ &= (T - M) \cdot \left(4\sqrt{\frac{\ln(T)}{\frac{M}{k}+1}}\right) \\ &\leq T \cdot \left(4\sqrt{\frac{\ln(T)}{\frac{M}{k}}}\right) \end{aligned} \tag{6}$$

Substituting (5) and (6) into (4) gives us

$$\sum_{t=1}^{T} (\ell(a_t) - \ell(a^*)) = M + 4T\sqrt{\frac{k\ln(T)}{M}}. \tag{7}$$

Substituting (3) and (7) into (2) leaves us with

$$R_T = 2 + M + 4T\sqrt{\frac{k\ln(T)}{M}}.$$

finally if we choose $M = k^{\frac{1}{3}}T^{\frac{2}{3}}$, we get

$$R_T \leq \mathcal{O}\left(k^{\frac{1}{3}}T^{\frac{2}{3}}\right)$$

□

4 Upper Confidence Bound Algorithm

We now demonstrate a better algorithm known as the Upper Confidence Bound (UCB) algorithm (Aner, et al. 2002). The idea behind this algorithm is to use confidence bounds to guide the exploration of our algorithm. The original paper was focused on reward maximization instead of loss minimization so we will modify the algorithm slightly to focus on loss minimization.

To define this algorithm we first define our lower confidence bounds

$$LCB_t(a) = \bar{\ell}_{t-1}(a) - b_t(a)$$

where

$$b_t(a) = 2\sqrt{\frac{\ln(T)}{m_{t-1}(a)+1}}$$

is derived from lemma 2. Our algorithm may then be written

Algorithm 3 UCB algorithm

```
for $t = 1, 2, \cdots, T$ do
    $a_t = \text{argmin}_a LCB_t(a)$
    receive loss and update the LCB's
end for
```

To motivate this algorithm consider figure 3. The algorithm is biased towards actions with a low empirical loss ($\bar{\ell}_t(a)$). This can be seen in the fact that if we are using the bounds drawn in blue, action 1 would be chosen. However, if we have a very low degree of certainty about the loss on an arm, we will choose that instead of an arm with a better emperical loss. For example, we are using the bounds drawn in black, action 2 would be taken even though it has a higher empirical loss.

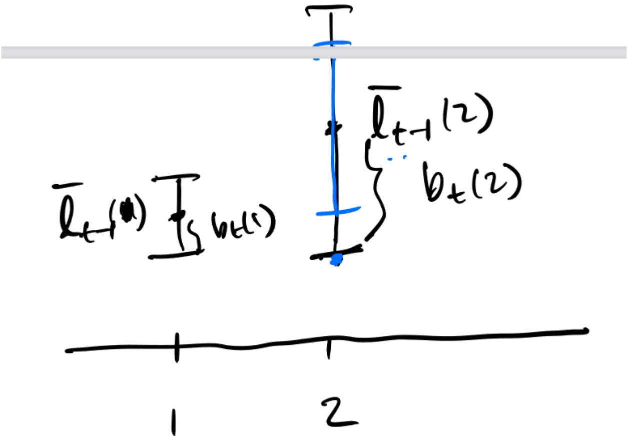

Figure 1: Example of how the UCB algorithm chooses actions

Before we analyze the performance of this algorithm, we need one more definition:

Definition 3. *The suboptimality gap of a choice "a" is given by*

$$\Delta(a) = \ell(a) - \ell(a^*).$$

Note that if the suboptimality gap of a is very large, then a is easily eliminated as poor choice of action. On the other hand, if the suboptimality gap of a is very small, then a is hard to eliminate as a suboptimal choice, but a also yields very little regret when chosen. Essentially it is an "easy" case if $\Delta(a)$ is very large or very small.

The performance of LCB is given by the following theorem.

Theorem 4. *The LCB algorithm satisfies*

$$R_T \leq \sum_{a:\Delta(a)>0} \frac{16\ln(T)}{\Delta(a)} + 3k \tag{8}$$

$$R_T \leq \mathcal{O}\left(\sqrt{TK}\right) \tag{9}$$

Equation (8) is known as the gap dependent bound and (9) is known as the gap independent bound. The gap dependent bound is good for cases where the suboptimality gaps are large, but if all the suboptimality gaps are small (e.g. $\Delta(a) = \frac{1}{T}$, $\forall a \neq a^*$), then the gap dependent bound gives very large regret despite this being an "easy" case. In these cases the gap independent bound is more useful.

There is also a matching lower bound theorem.

Theorem 5. *There exist a constant $c > 0$, such that for any algorithm $\mathcal{A}$, there exist a stochastic Multi-Armed Bandit environment, such that $\mathcal{A}$ satisfies $R_T \geq c \cdot \sqrt{Tk}$.*

In order to prove theorem 4, we will need a couple of lemmas.

Lemma 6. *On event E (the clean event):*

$$LCB_t(a) \leq \ell(a) \tag{10}$$

$$LCB_t(a) \geq \ell(a) - b_t(a) \tag{11}$$

Inequality (10) is known as honesty and (11) is known as tightness. The proof is straight forward, and we provide a brief outline here.

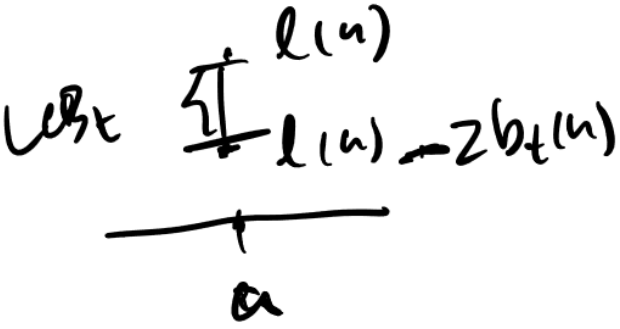

Figure 2: Visual representation of lemma 6

Proof. In event E

$$\begin{aligned} LCB_t(a) &= \bar{\ell}_{t-1}(a) - b_t(a) \\ &\leq (\ell(a) + b_t(a)) - b_t(a) \\ &= \bar{\ell}(a) \end{aligned}$$

and

$$\begin{aligned}
LCB_t(a) &= \bar{\ell}_{t-1}(a) - b_t(a) \\
&\geq (\ell(a) - b_t(a)) - b_t(a) \\
&= \ell(a) - 2b_t(a)
\end{aligned}$$

□

Lemma 7. *For all a, $\mathbb{E}\left[m_T(a)\right] \leq \frac{16\ln(T)}{\Delta(a)^2} + 3$*

This lemma is useful because it allows us to bound the number of pulls on an arm that is highly suboptimal. We now prove lemma 7.

Proof. Define

$$N(a) = \left\lceil \frac{16\ln(T)}{\Delta(a)^2} \right\rceil \tag{12}$$

We prove that on event E, $m_T(a) \leq N(a)$. Suppose by way of contradiction that $m_T(a) \geq N(a) + 1$. Then there must exist some value t such that $a_t = a$ and $m_{t-1}(a) = N(a)$. This implies that

$$2b_t(a) < \Delta(a). \tag{13}$$

The proof of equation (13) is left as an exercise to the reader. Now consider

$$\begin{aligned}
LCB_t(a) &\geq \ell(a) - 2b_t(A) \\
&> \ell(a) - \Delta(a) \\
&= \ell(a^*) \\
&\geq LCB_t(a^*). \qquad (14)
\end{aligned}$$

That suggests that $LCB_t(a)$ is not the minimum among all arms at round t, contradicting the fact that $a_t = a$. This proves (12).

We now consider the expectation on $m_T(a)$.

$$\begin{aligned}
\mathbb{E}\left[m_T(a)\right] &= \mathbb{E}\left[m_T(a)\mathbb{I}(E)\right] + \mathbb{E}\left[m_T(a)\mathbb{I}(E^C)\right] \\
&\leq \mathbb{E}\left[m_T(a)\mathbb{I}(E)\right] + T \cdot P(E^C) \\
&\leq \mathbb{E}\left[m_T(a)\mathbb{I}(E)\right] + 2 \qquad (15) \\
&\leq \left\lceil \frac{16\ln(T)}{\Delta(a)^2} \right\rceil + 2 \qquad (16) \\
&\leq \frac{16\ln(T)}{\Delta(a)^2} + 3
\end{aligned}$$

We justify (15) with lemma 2 and (16) with (12).

□

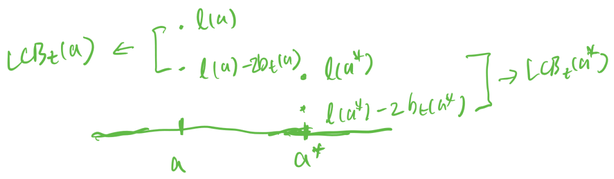

Figure 3: Visual representation of the contradiction argument from equation (14)

We are now ready to prove the gap dependent bound (9).

Proof.

$$\begin{aligned}
R_T &= \mathbb{E}\left[\sum_{t=1}^{T} \ell(a_t) - \ell(a^*)\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \Delta(a_t)\right] \\
&= \mathbb{E}\left[\sum_{t=a}^{k} \sum_{t:a_t=a} \Delta(a)\right] \\
&= \mathbb{E}\left[\sum_{t=a}^{k} m_T(a) \cdot \Delta(a)\right] \\
&= \sum_{t=a}^{k} \Delta(a)\mathbb{E}\left[m_T(a)\right] \qquad (17) \\
&= \sum_{a:\Delta(a)>0} \Delta(a)\mathbb{E}\left[m_T(a)\right] \\
&\leq \sum_{a:\Delta(a)>0} \Delta(a) \cdot \left(\frac{16\ln(T)}{\Delta(a)^2} + 3\right) \\
&\leq \sum_{a:\Delta(a)>0} \frac{16\ln(T)}{\Delta(a)} + \sum_{a:\Delta(a)>0} 3 \\
&= \sum_{a:\Delta(a)>0} \frac{16\ln(T)}{\Delta(a)} + 3k
\end{aligned}$$

Note (17) comes from lemma 7. □