Stochastic Multi-Armed Bandits

Other Algorithms

[1]ECE
Technical University of Crete

March 10, 2023

So Far: Explore then Exploit Algorithm

Definition

Explore-then-Exploit:

- Explore first: Alternate each arm for a total of N times per arm.
- Exploit after: Pick arm with maximum mean reward (in first N rounds) - play until round T.

So Far: Explore then Exploit Algorithm

Definition

Explore-then-Exploit:

- Explore first: Alternate each arm for a total of N times per arm.
- Exploit after: Pick arm with maximum mean reward (in first N rounds) - play until round T.

Performance

- (despite simplicity) can achieve sublinear regret $O(T^{\frac{2}{3}} log(T))$
- Key parameter to tune: explore period $N = O(T^{\frac{2}{3}})$

So Far: Explore then Exploit Algorithm

Definition

Explore-then-Exploit:

- Explore first: Alternate each arm for a total of N times per arm.
- Exploit after: Pick arm with maximum mean reward (in first N rounds) - play until round T.

Performance

- (despite simplicity) can achieve sublinear regret $O(T^{\frac{2}{3}} log(T))$
- Key parameter to tune: explore period $N = O(T^{\frac{2}{3}})$

What's missing?

1. **Question 1: Can we make the performance "smoother" along the entire K rounds?** (poor performance in first rounds)
2. **Question 2: Can we do better than $O(T^{\frac{2}{3}} log(T))$?**

ϵ-Greedy Algorithm

Definition

ϵ-Greedy:

- Maintain estimate $\hat{\mu}_i(t) = \frac{\sum_{n=1}^{t} r_i^n \cdot X_{i,n}}{\sum_{n=1}^{t} X_{i,n}}$
- Reminder: r_i^n: reward of arm i at round n; $X_{i,n}$: 1 if arm i played at round n.
- At any round t,
  - with probability ϵ_t: pick a random arm.
  - with probability $1 - \epsilon_t$: pick arm $j : \hat{\mu}_j(t) \geq \hat{\mu}_i(t), \forall i$ (best arm so far)

ϵ-Greedy Algorithm

Definition

ϵ-Greedy:

- Maintain estimate $\hat{\mu}_i(t) = \frac{\sum_{n=1}^{t} r_i^n \cdot X_{i,n}}{\sum_{n=1}^{t} X_{i,n}}$
- Reminder: r_i^n: reward of arm i at round n; $X_{i,n}$: 1 if arm i played at round n.
- At any round t,
  - with probability ϵ_t: pick a random arm.
  - with probability $1 - \epsilon_t$: pick arm $j : \hat{\mu}_j(t) \geq \hat{\mu}_i(t), \forall i$ (best arm so far)

- Is this algorithm sublinear also?

ϵ-Greedy Algorithm

Definition

ϵ-Greedy:

- Maintain estimate $\hat{\mu}_i(t) = \frac{\sum_{n=1}^{t} r_i^n \cdot X_{i,n}}{\sum_{n=1}^{t} X_{i,n}}$
- Reminder: r_i^n: reward of arm i at round n; $X_{i,n}$: 1 if arm i played at round n.
- At any round t,
  - with probability ϵ_t: pick a random arm.
  - with probability $1 - \epsilon_t$: pick arm $j : \hat{\mu}_j(t) \geq \hat{\mu}_i(t), \forall i$ (best arm so far)

- Is this algorithm sublinear also?
- Is it better than "Explore-then-Exploit"?

ϵ-Greedy Algorithm

Definition

ϵ-Greedy:

- Maintain estimate $\hat{\mu}_i(t) = \frac{\sum_{n=1}^{t} r_i^n \cdot X_{i,n}}{\sum_{n=1}^{t} X_{i,n}}$
- Reminder: r_i^n: reward of arm i at round n; $X_{i,n}$: 1 if arm i played at round n.
- At any round t,
  - with probability ϵ_t: pick a random arm.
  - with probability $1 - \epsilon_t$: pick arm $j : \hat{\mu}_j(t) \geq \hat{\mu}_i(t), \forall i$ (best arm so far)

- Is this algorithm sublinear also?
- Is it better than "Explore-then-Exploit"?
- Only (key??) parameter to tune is ϵ_t...How?

ϵ-Greedy Algorithm

Definition

ϵ-Greedy:

- Maintain estimate $\hat{\mu}_i(t) = \frac{\sum_{n=1}^{t} r_i^n \cdot X_{i,n}}{\sum_{n=1}^{t} X_{i,n}}$
- Reminder: r_i^n: reward of arm i at round n; $X_{i,n}$: 1 if arm i played at round n.
- At any round t,
  - with probability ϵ_t: pick a random arm.
  - with probability $1 - \epsilon_t$: pick arm $j : \hat{\mu}_j(t) \geq \hat{\mu}_i(t), \forall i$ (best arm so far)

- Is this algorithm sublinear also?
- Is it better than "Explore-then-Exploit"?
- Only (key??) parameter to tune is ϵ_t...How?

ϵ-Greedy vs. Explore-then-Exploit

Question: What are the advantages of ϵ-Greedy (if any)?

ϵ-Greedy vs. Explore-then-Exploit

Question: What are the advantages of ϵ-Greedy (if any)?

- Answer 1: Initial rounds are not *just* exploration $\rightarrow$ begins to also exploit information immediately.

ϵ-Greedy vs. Explore-then-Exploit

Question: What are the advantages of ϵ-Greedy (if any)?

- Answer 1: Initial rounds are not *just* exploration $\rightarrow$ begins to also exploit information immediately.
- Answer 2: Never stops exploring $\rightarrow$ avoids getting stuck with wrong arm.

How to tune ϵ_t?

What about $\epsilon_t = \epsilon$ (constant)?

- Question: does this have sublinear regret?

How to tune ϵ_t?

What about $\epsilon_t = \epsilon$ (constant)?

- Question: does this have sublinear regret?
- Answer: No! Easy to prove (will be in homework).

How to tune ϵ_t?

What about $\epsilon_t = \epsilon$ (constant)?

- Question: does this have sublinear regret?
- Answer: No! Easy to prove (will be in homework).
- ϵ_t must decrease with t.

How to tune ϵ_t?

What about $\epsilon_t = \epsilon$ (constant)?

- Question: does this have sublinear regret?
- Answer: No! Easy to prove (will be in homework).
- ϵ_t must decrease with t.

What about $\epsilon_t = O(t^{-\frac{1}{3}})$?

How to tune ϵ_t?

What about $\epsilon_t = \epsilon$ (constant)?

- Question: does this have sublinear regret?
- Answer: No! Easy to prove (will be in homework).
- ϵ_t must decrease with t.

What about $\epsilon_t = O(t^{-\frac{1}{3}})$?

- Question: how many exploration rounds in total?

How to tune ϵ_t?

What about $\epsilon_t = \epsilon$ (constant)?

- Question: does this have sublinear regret?
- Answer: No! Easy to prove (will be in homework).
- ϵ_t must decrease with t.

What about $\epsilon_t = O(t^{-\frac{1}{3}})$?

- Question: how many exploration rounds in total?
- Answer: $O(T^{-\frac{2}{3}}) \cdots$ Why?

How to tune ϵ_t?

What about $\epsilon_t = \epsilon$ (constant)?

- Question: does this have sublinear regret?
- Answer: No! Easy to prove (will be in homework).
- ϵ_t must decrease with t.

What about $\epsilon_t = O(t^{-\frac{1}{3}})$?

- Question: how many exploration rounds in total?
- Answer: $O(T^{-\frac{2}{3}}) \cdots$ Why?
- Question: how many explorations in Explore-then-Exploit?

How to tune ϵ_t?

What about $\epsilon_t = \epsilon$ (constant)?

- Question: does this have sublinear regret?
- Answer: No! Easy to prove (will be in homework).
- ϵ_t must decrease with t.

What about $\epsilon_t = O(t^{-\frac{1}{3}})$?

- Question: how many exploration rounds in total?
- Answer: $O(T^{-\frac{2}{3}}) \cdots$ Why?
- Question: how many explorations in Explore-then-Exploit?
- Answer: Remember, $N = O(T^{\frac{2}{3}})$ for best sublinear regret.

ϵ-Greedy with sublinear ϵ_t

Theorem

With $\epsilon_t = O(t^{-1/3}(k \log t)^{1/3})$, ϵ-Greedy achieves $O(t^{2/3}(k \log t)^{1/3})$ regret, for *any* $t \leq T$

ϵ-Greedy with sublinear ϵ_t

Theorem

With $\epsilon_t = O(t^{-1/3}(k \log t)^{1/3})$, ϵ-Greedy achieves $O(t^{2/3}(k \log t)^{1/3})$ regret, for *any* $t \leq T$

Remarks

- Proof follows similar steps to Explore-then-Exploit.
- Q: Key difference with Explore-then-Exploit?

ϵ-Greedy with sublinear ϵ_t

Theorem

With $\epsilon_t = O(t^{-1/3}(k \log t)^{1/3})$, ϵ-Greedy achieves $O(t^{2/3}(k \log t)^{1/3})$ regret, for *any* $t \leq T$

Remarks

- Proof follows similar steps to Explore-then-Exploit.
- Q: Key difference with Explore-then-Exploit?
- A: The latter has sublinear regret only at the end (T). ϵ-Greedy has sublinear regret along the way as we wanted (see slide 4)!

ϵ-Greedy with sublinear ϵ_t

Theorem

With $\epsilon_t = O(t^{-1/3}(k \log t)^{1/3})$, ϵ-Greedy achieves $O(t^{2/3}(k \log t)^{1/3})$ regret, for *any* $t \leq T$

Remarks

- Proof follows similar steps to Explore-then-Exploit.
- Q: Key difference with Explore-then-Exploit?
- A: The latter has sublinear regret only at the end (T). ϵ-Greedy has sublinear regret along the way as we wanted (see slide 4)!
- Q: What is another advantage?

ϵ-Greedy with sublinear ϵ_t

Theorem

With $\epsilon_t = O(t^{-1/3}(k \log t)^{1/3})$, ϵ-Greedy achieves $O(t^{2/3}(k \log t)^{1/3})$ regret, for *any* $t \leq T$

Remarks

- Proof follows similar steps to Explore-then-Exploit.
- Q: Key difference with Explore-then-Exploit?
- A: The latter has sublinear regret only at the end (T). ϵ-Greedy has sublinear regret along the way as we wanted (see slide 4)!
- Q: What is another advantage?
- A: tuning its key parameter ϵ_t does *not* require knowing the horizon!

ϵ-Greedy with sublinear ϵ_t

Theorem

With $\epsilon_t = O(t^{-1/3}(k \log t)^{1/3})$, ϵ-Greedy achieves $O(t^{2/3}(k \log t)^{1/3})$ regret, for *any* $t \leq T$

Remarks

- Proof follows similar steps to Explore-then-Exploit.
- Q: Key difference with Explore-then-Exploit?
- A: The latter has sublinear regret only at the end (T). ϵ-Greedy has sublinear regret along the way as we wanted (see slide 4)!
- Q: What is another advantage?
- A: tuning its key parameter ϵ_t does *not* require knowing the horizon!
- Other sublinear choises for ϵ_t also can work (might even do better).

Can we do even better?

- Q: What is the main disadvantage of *both* algorithms?

Can we do even better?

- Q: What is the main disadvantage of *both* algorithms?
- A: During exploration, *both* explore *any* arm (even ones that seem bad) with equal probability...

Can we do even better?

- Q: What is the main disadvantage of *both* algorithms?
- A: During exploration, *both* explore *any* arm (even ones that seem bad) with equal probability...
- Enter "Upper Confidence Bound (UCB)" algorithm.

Upper Confidence Bound (UCB) Algorithm

Intuition

Imagine we are at round t:

- Seems suboptimal to explore arms i, j with equal probability if $\hat{\mu}_i(t) \gg \hat{\mu}_j(t)$

Upper Confidence Bound (UCB) Algorithm

Intuition

Imagine we are at round t:

- Seems suboptimal to explore arms i, j with equal probability if $\hat{\mu}_i(t) \gg \hat{\mu}_j(t)$
- Goal 1: probability of exploring arm i at round t proportional to $\hat{\mu}_i(t)$

Upper Confidence Bound (UCB) Algorithm

Intuition

Imagine we are at round t:

- Seems suboptimal to explore arms i, j with equal probability if $\hat{\mu}_i(t) \gg \hat{\mu}_j(t)$
- Goal 1: probability of exploring arm i at round t proportional to $\hat{\mu}_i(t)$
- Goal 2: BUT...we should still explore an arm j if it has been explored very few times ($\rightarrow$ *ourestimate*$_j(t)$ is not to be trusted)

Upper Confidence Bound (UCB) Algorithm

Intuition

Imagine we are at round t:

- Seems suboptimal to explore arms i, j with equal probability if $\hat{\mu}_i(t) \gg \hat{\mu}_j(t)$
- Goal 1: probability of exploring arm i at round t proportional to $\hat{\mu}_i(t)$
- Goal 2: BUT...we should still explore an arm j if it has been explored very few times ($\rightarrow$ $ourestimate_j(t)$ is not to be trusted)
- Q: How to balance *both* goals?

Upper Confidence Bound (UCB) Algorithm

Intuition

Imagine we are at round t:

- Seems suboptimal to explore arms i, j with equal probability if $\hat{\mu}_i(t) \gg \hat{\mu}_j(t)$
- Goal 1: probability of exploring arm i at round t proportional to $\hat{\mu}_i(t)$
- Goal 2: BUT...we should still explore an arm j if it has been explored very few times ($\rightarrow$ $ourestimate_j(t)$ is not to be trusted)
- Q: How to balance *both* goals?
- A: “Optimism under uncertainty”.

Upper Confidence Bound (UCB) Algorithm

Definition: Upper Confidence Bound Algorithm

- Maintain estimate $\hat{\mu}_i(t) = \frac{\sum_{n=1}^{t} r_i^n \cdot X_{i,n}}{\sum_{n=1}^{t} X_{i,n}}$
- Maintain estimate $Q_i(t) = \sum_{n=1}^{t} X_{i,n}$

Upper Confidence Bound (UCB) Algorithm

Definition: Upper Confidence Bound Algorithm

- Maintain estimate $\hat{\mu}_i(t) = \frac{\sum_{n=1}^{t} r_i^n \cdot X_{i,n}}{\sum_{n=1}^{t} X_{i,n}}$
- Maintain estimate $Q_i(t) = \sum_{n=1}^{t} X_{i,n}$
- At any round t:
  - update $ucb_i(t) = \hat{\mu}_i(t) + \sqrt{\frac{\ln T}{Q_i(t)}}$ ("upper confidence bound")
  - play arm j: $ucb_j(t) \geq ucb_i(t), \ \forall i$

Upper Confidence Bound (UCB) Algorithm

Definition: Upper Confidence Bound Algorithm

- Maintain estimate $\hat{\mu}_i(t) = \frac{\sum_{n=1}^{t} r_i^n \cdot X_{i,n}}{\sum_{n=1}^{t} X_{i,n}}$
- Maintain estimate $Q_i(t) = \sum_{n=1}^{t} X_{i,n}$
- At any round t:
  - update $ucb_i(t) = \hat{\mu}_i(t) + \sqrt{\frac{\ln T}{Q_i(t)}}$ ("upper confidence bound")
  - play arm j: $ucb_j(t) \geq ucb_i(t), \ \forall i$

- Arm i is played either because it has a high predicted reward ($\hat{\mu}_i(t)$ is large) or because it hasn't been explored enough ($Q_i(t)$ is small).

Upper Confidence Bound (UCB) Algorithm

Definition: Upper Confidence Bound Algorithm

- Maintain estimate $\hat{\mu}_i(t) = \frac{\sum_{n=1}^{t} r_i^n \cdot X_{i,n}}{\sum_{n=1}^{t} X_{i,n}}$
- Maintain estimate $Q_i(t) = \sum_{n=1}^{t} X_{i,n}$
- At any round t:
  - update $ucb_i(t) = \hat{\mu}_i(t) + \sqrt{\frac{\ln T}{Q_i(t)}}$ ("upper confidence bound")
  - play arm j: $ucb_j(t) \geq ucb_i(t), \quad \forall i$

- Arm i is played either because it has a high predicted reward ($\hat{\mu}_i(t)$ is large) or because it hasn't been explored enough ($Q_i(t)$ is small).
- We will show a simple(r) proof suggesting that UCB achieves better regret $O(\sqrt{T})$ than previous schemes

Upper Confidence Bound (UCB) Algorithm

Definition: Upper Confidence Bound Algorithm

- Maintain estimate $\hat{\mu}_i(t) = \frac{\sum_{n=1}^{t} r_i^n \cdot X_{i,n}}{\sum_{n=1}^{t} X_{i,n}}$
- Maintain estimate $Q_i(t) = \sum_{n=1}^{t} X_{i,n}$
- At any round t:
  - update $ucb_i(t) = \hat{\mu}_i(t) + \sqrt{\frac{\ln T}{Q_i(t)}}$ ("upper confidence bound")
  - play arm j: $ucb_j(t) \geq ucb_i(t), \quad \forall i$

- Arm i is played either because it has a high predicted reward ($\hat{\mu}_i(t)$ is large) or because it hasn't been explored enough ($Q_i(t)$ is small).
- We will show a simple(r) proof suggesting that UCB achieves better regret $O(\sqrt{T})$ than previous schemes
- In fact, UCB achieves order optimal regret $O()$ (see Bandit book)

Upper Confidence Bound (UCB) Algorithm

Definition: Upper Confidence Bound Algorithm

- Maintain estimate $\hat{\mu}_i(t) = \frac{\sum_{n=1}^{t} r_i^n \cdot X_{i,n}}{\sum_{n=1}^{t} X_{i,n}}$
- Maintain estimate $Q_i(t) = \sum_{n=1}^{t} X_{i,n}$
- At any round t:
  - update $ucb_i(t) = \hat{\mu}_i(t) + \sqrt{\frac{\ln T}{Q_i(t)}}$ ("upper confidence bound")
  - play arm j: $ucb_j(t) \geq ucb_i(t), \quad \forall i$

- Arm i is played either because it has a high predicted reward ($\hat{\mu}_i(t)$ is large) or because it hasn't been explored enough ($Q_i(t)$ is small).
- We will show a simple(r) proof suggesting that UCB achieves better regret $O(\sqrt{T})$ than previous schemes
- In fact, UCB achieves order optimal regret $O()$ (see Bandit book)
- Remark: "order optimal" means that no other algorithm exist that gets better regret in stochastic MAB problems.