Adaptively Tracking the Best Bandit Arm with an Unknown Number of Distribution Changes

Peter Auer[1], Ronald Ortner[1] and Pratik Gajane[1,2]

[1]Montanuniversität Leoben

[2]chist-era project DELTA
Austrian Science Fund (FWF): I 3437

Switching Bandit Setting

Stochastic multi-armed bandit problem with changes

- A set of arms $\{1, \ldots, K\}$.
- Learner chooses arm a_t at steps $t = 1, 2, \ldots, T$.
- Learner receives random reward $r_t \in [0, 1]$ with (unknown) mean $\mathbb{E}[r_t] = \mu_t(a_t)$.
- The mean rewards $\mu_t(a)$ depend on time t.

Performance Measure

We define the **regret** in this setting as

$$\sum_{t=1}^{T} \left(\mu_t^* - r_t\right),$$

where $\mu_t^* := \max_a \mu_t(a)$ is the optimal mean reward at step t.

Performance Measure

We define the **regret** in this setting as

$$\sum_{t=1}^{T} \left(\mu_t^* - r_t\right),$$

where $\mu_t^* := \max_a \mu_t(a)$ is the optimal mean reward at step t.

The regret will depend on the number of changes L,

i.e., the number of times when $\mu_{t-1}(a) \neq \mu_t(a)$ for some a.

Previous Work

When the number of changes L is known:

- Upper bounds of $\tilde{O}(\sqrt{KLT})$ for algorithms which use number of changes L:
  - EXP3.S (Auer et al., SIAM J. Comput. 2002)
  - Garivier& Moulines, ALT 2011
  - Allesiardo et al, IJDSA 2017

Previous Work

When the number of changes L is known:

- Upper bounds of $\tilde{O}(\sqrt{KLT})$ for algorithms which use number of changes L:
  - EXP3.S (Auer et al., SIAM J. Comput. 2002)
  - Garivier& Moulines, ALT 2011
  - Allesiardo et al, IJDSA 2017

- Lower bound of $\Omega(\sqrt{KLT})$, which holds even when L is known.

Previous Work

When the number of changes L is known:

- Upper bounds of $\tilde{O}(\sqrt{KLT})$ for algorithms which use number of changes L:
  - EXP3.S (Auer et al., SIAM J. Comput. 2002)
  - Garivier& Moulines, ALT 2011
  - Allesiardo et al, IJDSA 2017
- Lower bound of $\Omega(\sqrt{KLT})$, which holds even when L is known.

For unknwown L:

- Optimal regret bounds for two arms (Auer et al., EWRL 2018)

Previous Work

When the number of changes L is known:

- Upper bounds of $\tilde{O}(\sqrt{KLT})$ for algorithms which use number of changes L:
  - EXP3.S (Auer et al., SIAM J. Comput. 2002)
  - Garivier& Moulines, ALT 2011
  - Allesiardo et al, IJDSA 2017
- Lower bound of $\Omega(\sqrt{KLT})$, which holds even when L is known.

For unknwown L:

- Optimal regret bounds for two arms (Auer et al., EWRL 2018)
- (Auer et al., EWRL 2018) was also the base for (Chen et al., 2019)

AdSwitch (for two arms)

AdSwitch for two arms (Sketch)

For episodes $l = 1, 2, \ldots$ do:

- **Estimation phase**:
  Select both arms are selected alternatingly,
  until better arm has been identified.

AdSwitch (for two arms)

AdSwitch for two arms (Sketch)

For episodes $l = 1, 2, \ldots$ do:

- **Estimation phase**:
  Select both arms are selected alternatingly,
  until better arm has been identified.
- **Exploitation and checking phase**:
  - Mostly exploit the empirical best arm.
  - Sometimes sample both arms to check for change.
    If a change is detected then start a new episode.

AdSwitch (for two arms)

AdSwitch for two arms

For episodes $l = 1, 2, \ldots$ do:

- **Estimation phase**:
  Sample both arms alternatingly until
  $\left|\hat{\mu}_1[t,s] - \hat{\mu}_2[t,s]\right| > \sqrt{\frac{C_1 \log T}{t-s}}$. Set $\hat{\Delta} := \hat{\mu}_1 - \hat{\mu}_2$.

AdSwitch (for two arms)

AdSwitch for two arms

For episodes $l = 1, 2, \ldots$ do:

- **Estimation phase**:
  Sample both arms alternatingly until
  $\left|\hat{\mu}_1[t,s] - \hat{\mu}_2[t,s]\right| > \sqrt{\frac{C_1 \log T}{t-s}}$. Set $\hat{\Delta} := \hat{\mu}_1 - \hat{\mu}_2$.
- **Exploitation and checking phase**:
  1. Let $d_i = 2^{-i}$ and $I = \max\{i : d_i \geq \hat{\Delta}\}$.
  2. Randomly choose i from $\{1, 2, \ldots, I\}$ with probabilities $d_i\sqrt{\frac{l+1}{T}}$.
  3. With remaining probability choose empirically best arm and repeat phase.
  4. If an i is chosen, sample both arms alternatingly for $2\left\lceil \frac{C_2 \log T}{d_i^2} \right\rceil$ steps to check for changes of size d_i:
     If $\hat{\mu}_1 - \hat{\mu}_2 \notin \left[\hat{\Delta} - \frac{d_i}{4}, \hat{\Delta} + \frac{d_i}{4}\right]$, then start a new episode.

Regret Bound for AdSwitch for two arms

W.h.p. the algorithm

- will identify the better arm in the exploration phase,
- will detect significant changes in the exploitation phase, while the overhead for additional sampling is not too large,
- will make no false detections of a change.

Regret Bound for AdSwitch for two arms

W.h.p. the algorithm

- will identify the better arm in the exploration phase,
- will detect significant changes in the exploitation phase, while the overhead for additional sampling is not too large,
- will make no false detections of a change.

Regret Bound for AdSwitch for two arms

W.h.p. the algorithm

- will identify the better arm in the exploration phase,
- will detect significant changes in the exploitation phase, while the overhead for additional sampling is not too large,
- will make no false detections of a change.

Theorem

The regret of AdSwitch in a switching bandit problem with two arms and L changes is at most

$$O\Big((\log T)\sqrt{(L+1)T}\Big).$$

The ADSWITCH Algorithm (Sketch)

For episodes ($\approx$ estimated changes) $\ell = 1, 2, \ldots$ do:

- Let the set *GOOD* contain all arms.

The ADSWITCH Algorithm (Sketch)

For episodes (≈ estimated changes) $\ell = 1, 2, \ldots$ do:

- Let the set *GOOD* contain all arms.
- Select all arms in *GOOD* alternatingly.

The ADSWITCH Algorithm (Sketch)

For episodes (≈ estimated changes) $\ell = 1, 2, \ldots$ do:

- Let the set *GOOD* contain all arms.
- Select all arms in *GOOD* alternatingly.
- Remove bad arms *a* from *GOOD*.

The ADSWITCH Algorithm (Sketch)

For episodes (≈ estimated changes) $\ell = 1, 2, \ldots$ do:

- Let the set *GOOD* contain all arms.
- Select all arms in *GOOD* alternatingly.
- Remove bad arms *a* from *GOOD*.
- Sometimes sample discarded arms not in *GOOD* (to be able to check for changes).

The ADSWITCH Algorithm (Sketch)

For episodes ($\approx$ estimated changes) $\ell = 1, 2, \ldots$ do:

- Let the set *GOOD* contain all arms.
- Select all arms in *GOOD* alternatingly.
- Remove bad arms *a* from *GOOD*.
- Sometimes sample discarded arms not in *GOOD* (to be able to check for changes).
- Check for changes (of all arms). If a change is detected, start a new episode.

The ADSWITCH Algorithm (Sketch)

For episodes ($\approx$ estimated changes) $\ell = 1, 2, \ldots$ do:

- Let the set *GOOD* contain all arms.
- Select all arms in *GOOD* alternatingly.
- Remove bad arms *a* from *GOOD*.
- *Sometimes sample discarded arms not in GOOD* (to be able to check for changes).
- Check for changes (of all arms). If a change is detected, start a new episode.

The ADSWITCH Algorithm (Sketch with more details)

For episodes (≈ estimated changes) $\ell = 1, 2, \ldots$ do:

- Let the set *GOOD* contain all arms.
- Select all arms in $GOOD \cup \mathcal{S}$ alternatingly.
- Remove bad arms *a* from *GOOD*.
  Keep in mind empirical gaps $\tilde{\Delta}(a)$.
- Sometimes sample discarded arms not in *GOOD*:
  - Define set $\mathcal{S}$ of arms $a \notin GOOD$ to be sampled.
  - At each step t, each $a \notin GOOD$, for $d_i \approx \tilde{\Delta}(a), 2\tilde{\Delta}(a), 4\tilde{\Delta}(a), \ldots$, with probability $d_i\sqrt{\ell/(KT)}$ add a to $\mathcal{S}$.
  - Keep a in $\mathcal{S}$ until it has been sampled $1/d_i^2$ times.
- Check for changes (of all arms).
  If a change is detected, start a new episode.

Condition for eviction from GOOD

An arm a is evicted from *GOOD* at time t, if

$$\max_{a' \in \mathrm{GOOD}_t} \hat{\mu}_{[s,t]}(a') - \hat{\mu}_{[s,t]}(a) > \sqrt{\frac{C_1 \log T}{n_{[s,t]}(a) - 1}},$$

start of the current episode $\leq s < t$ and $n_{[s,t]}(a) \geq 2$.

$$n_{[s,t]}(a) = \#\{s \leq \tau \leq t : a_\tau = a\}, \quad \hat{\mu}_{[s,t]}(a) = \frac{1}{n_{[s,t]}(a)} \sum_{\tau : s \leq \tau \leq t, a_\tau = a} r_t.$$

Condition for eviction from GOOD

An arm a is evicted from *GOOD* at time t, if

$$\max_{a' \in \mathrm{GOOD}_t} \hat{\mu}_{[s,t]}(a') - \hat{\mu}_{[s,t]}(a) > \sqrt{\frac{C_1 \log T}{n_{[s,t]}(a) - 1}},$$

start of the current episode $\leq s < t$ and $n_{[s,t]}(a) \geq 2$.

$$n_{[s,t]}(a) = \#\{s \leq \tau \leq t : a_\tau = a\}, \quad \hat{\mu}_{[s,t]}(a) = \frac{1}{n_{[s,t]}(a)} \sum_{\tau : s \leq \tau \leq t, a_\tau = a} r_t.$$

For a suitable constant C_1, this is a standard confidence bound on the mean rewards.

Condition for eviction from GOOD

An arm a is evicted from *GOOD* at time t, if

$$\max_{a' \in \mathrm{GOOD}_t} \hat{\mu}_{[s,t]}(a') - \hat{\mu}_{[s,t]}(a) > \sqrt{\frac{C_1 \log T}{n_{[s,t]}(a) - 1}},$$

start of the current episode $\leq s < t$ and $n_{[s,t]}(a) \geq 2$.

$$n_{[s,t]}(a) = \#\{s \leq \tau \leq t : a_\tau = a\}, \quad \hat{\mu}_{[s,t]}(a) = \frac{1}{n_{[s,t]}(a)} \sum_{\tau : s \leq \tau \leq t, a_\tau = a} r_t.$$

For a suitable constant C_1, this is a standard confidence bound on the mean rewards.

$$\tilde{\mu}_\ell(a) \leftarrow \hat{\mu}_{[s,t]}(a), \quad \tilde{\Delta}_\ell(a) \leftarrow \max_{a' \in \mathrm{GOOD}_t} \hat{\mu}_{[s,t]}(a') - \hat{\mu}_{[s,t]}(a).$$

Check for changes in an arm in GOOD

Declare a change for $a \in \mathrm{GOOD}$ at time t, if

$$\left|\hat{\mu}_{[s_1,s_2]}(a) - \hat{\mu}_{[s,t]}(a)\right| > \sqrt{\frac{2\log T}{n_{[s_1,s_2]}(a)}} + \sqrt{\frac{2\log T}{n_{[s,t]}(a)}},$$

for some $s_1 \leq s_2 < s \leq t$ within the current episode.

Another variation of the standard confidence bound on the mean rewards.

Check for changes in an arm not in GOOD

- Size of the change to be detected : $d_i = 2^{-i}$ where $d_i \geq \frac{\tilde{\Delta}(a)}{16}$.

Check for changes in an arm not in GOOD

- Size of the change to be detected : $d_i = 2^{-i}$ where $d_i \geq \frac{\tilde{\Delta}(a)}{16}$.
- Number of samples needed : $n = \lceil 2(\log T)/d_i^2 \rceil$.

Check for changes in an arm not in GOOD

- Size of the change to be detected : $d_i = 2^{-i}$ where $d_i \geq \frac{\tilde{\Delta}(a)}{16}$.
- Number of samples needed : $n = \lceil 2(\log T)/d_i^2 \rceil$.
- With probability $d_i \sqrt{\ell/(KT \log T)}$, add sampling obligation (d_i, n, s) at time s.

Check for changes in an arm not in GOOD

- Size of the change to be detected : $d_i = 2^{-i}$ where $d_i \geq \frac{\tilde{\Delta}(a)}{16}$.
- Number of samples needed : $n = \lceil 2(\log T)/d_i^2 \rceil$.
- With probability $d_i\sqrt{\ell/(KT \log T)}$, add sampling obligation (d_i, n, s) at time s.
- Declare a change for $a \notin \mathrm{GOOD}$ at time t, if

$$\left|\hat{\mu}_{[s,t]}(a) - \tilde{\mu}_\ell(a)\right| > \tilde{\Delta}_\ell(a)/4 + \sqrt{\frac{2 \log T}{n_{[s,t]}(a)}}.$$

Regret Bound for ADSWITCH

W.h.p. the algorithm

- identifies bad arms,
- makes no false detections of a change,
- detects significant changes fast enough,
  while the overhead for additional sampling is not too large.

Regret Bound for ADSWITCH

W.h.p. the algorithm

- identifies bad arms,
- makes no false detections of a change,
- detects significant changes fast enough,
  while the overhead for additional sampling is not too large.

Theorem

The expected regret of AdSwitch in a switching bandit problem with K arms and L changes after T steps is at most

$$O\left(\sqrt{K(L+1)T(\log T)}\right).$$

Empirical average while no change

Lemma

If no change between time steps s and t, then w.h.p $\forall$ arms the empirical average is close to their true mean.

Empirical average while no change

Lemma

If no change between time steps s and t, then w.h.p $\forall$ arms the empirical average is close to their true mean.

- With probability $1 - 2K/T^2$, for all $1 \leq s \leq t \leq T$ with $L[s,t] = 0$, and all arms a,

$$\left|\hat{\mu}_{[s,t]}(a) - \mu_s(a)\right| < \sqrt{\frac{2\log T}{n_{[s,t]}(a)}}.$$

Empirical average while no change

Lemma

If no change between time steps s and t, then w.h.p $\forall$ arms the empirical average is close to their true mean.

- With probability $1 - 2K/T^2$, for all $1 \leq s \leq t \leq T$ with $L[s,t] = 0$, and all arms a,

$$\left|\hat{\mu}_{[s,t]}(a) - \mu_s(a)\right| < \sqrt{\frac{2 \log T}{n_{[s,t]}(a)}}.$$

- Since the error probability $2K/T^2$ causes only diminishing regret, we assume that all inequalities of the lemma are satisfied.

Counting the number of episodes

Lemma

The total number of episodes is bounded by the number of changes L.

Counting the number of episodes

Lemma

The total number of episodes is bounded by the number of changes L.

- For every episode l, the number of changes in l is at least 1.

Counting the number of episodes

Lemma

The total number of episodes is bounded by the number of changes L.

- For every episode l, the number of changes in l is at least 1.
- The algorithm starts a new episode only if there is a change in the current episode.

Distinguishing the sources of regret

$$\text{Regret at time } t = \text{regret wrt the best good arm}$$
$$+ \text{ regret of the best good arm wrt optimal arm}$$

best good arm = $\arg\max_{a \in \mathrm{GOOD}} \mu_a$

Distinguishing the sources of regret

$$\text{Regret at time } t = \text{regret wrt the best good arm} + \text{regret of the best good arm wrt optimal arm}$$

best good arm = $\arg\max_{a \in \mathrm{GOOD}} \mu_a$

- No such decomposition needed when optimal arm is in GOOD.

Distinguishing the sources of regret

$$\text{Regret at time } t = \text{regret wrt the best good arm} \\ + \text{regret of the best good arm wrt optimal arm}$$

best good arm = $\arg\max_{a \in \mathrm{GOOD}} \mu_a$

- No such decomposition needed when optimal arm is in GOOD.
- Otherwise two cases:
  - mean reward of optimal arm is close to the mean reward when it was evicted.
  - mean reward of optimal arm is far from the mean reward when it was evicted.

Distinguishing the sources of regret

- A good arm is selected.

Distinguishing the sources of regret

- A good arm is selected.
- A bad arm is selected, and its regret is not much larger than its eviction gap.

Distinguishing the sources of regret

- A good arm is selected.
- A bad arm is selected, and its regret is not much larger than its eviction gap.
- A bad arm is selected, its regret is large, and
  - its mean reward is far from the mean reward when it was evicted.
  - its mean reward is relatively close to the mean reward when it was evicted.

Concluding remarks

- First algorithm for switching bandits that achieves optimal regret bounds without knowing the number of changes in advance.
- Main technical contribution is the delicate testing schedule of the apparently inferior arms.
- Extending our approach to reinforcement learning in changing Markov decision processes?