Finite analysis of Nonstationary multi armed bandits

Sahil Chaudhary, Pratham Gupta

Indian Institute of Science

3 November 2025

Non-Stationary Distribution?

Definition (What exactly is non stationary?)

Non Stationary distribution is a distribution which is **not** stationary.

Non-Stationary Distribution?

Definition (What exactly is non stationary?)

Non Stationary distribution is a distribution which is **not** stationary.

Where does it really occur?

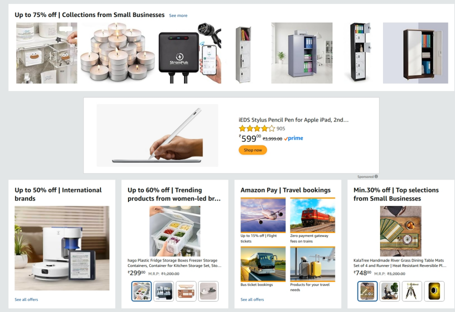


Figure: Recommendations from Amazon

Non-Stationary Distribution?

More Formally?

Definition (Non stationary distribution)

Let $X_1, X_2, \cdot$ be a sequence of observations. The distribution is said to be non stationary if,

$$X_i \sim \mathcal{D}_i, \quad \text{for some } i \neq j, \mathcal{D}_i \neq \mathcal{D}_j \tag{1}$$

Data Stream

Let $(x_1, x_2, \cdots, x_T)$ be the sequence of observations with $x_t \in [0, 1]$ with mean $\mu_t \in [0, 1]$. We consider following kinds of stream of data:

- **Static:** A stream of data is said to be static if $\mu_t = \mu$ for all $t \in [T]$ and some $\mu \in [0, 1]$.

Data Stream

Let $(x_1, x_2, \cdots, x_T)$ be the sequence of observations with $x_t \in [0, 1]$ with mean $\mu_t \in [0, 1]$. We consider following kinds of stream of data:

- **Static:** A stream of data is said to be static if $\mu_t = \mu$ for all $t \in [T]$ and some $\mu \in [0, 1]$.
- **Abruptly-Changing:** A stream is said to be abruptly changing if $\mu_t = \mu_{t+1}$ for some changepoints $\mathcal{T}_c \subset [T]$.

Data Stream

Let $(x_1, x_2, \cdots, x_T)$ be the sequence of observations with $x_t \in [0, 1]$ with mean $\mu_t \in [0, 1]$. We consider following kinds of stream of data:

- **Static:** A stream of data is said to be static if $\mu_t = \mu$ for all $t \in [T]$ and some $\mu \in [0, 1]$.
- **Abruptly-Changing:** A stream is said to be abruptly changing if $\mu_t = \mu_{t+1}$ for some changepoints $\mathcal{T}_c \subset [T]$.
- **Globally-Changing:** A stream is said to be globally changing if $|\mu_t - \mu_{t+1}| \leq b$ for all $t \in [T]$ and for some $b \in (0, 1)$.

kinds of Non Stationary

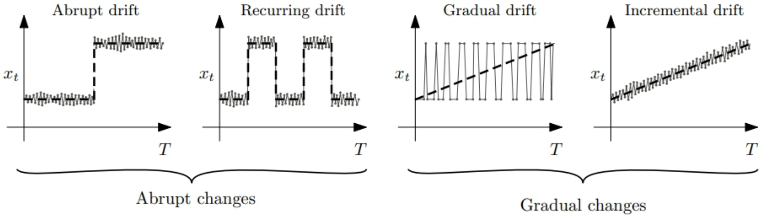
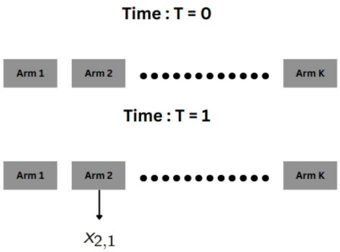
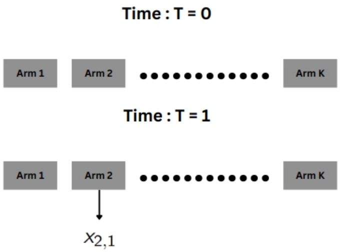


Figure: Kinds of Concept Drift.

Back to Bandits



Back to Bandits



Essentially, $\exists$K arms, at each time $t \in [T]$, we observe a reward $x_{I(t),t}$ where mean of the rewards are $\mu_{I(t),t}$ and regret can be defined as,

$$\text{reg}(t) = \max_i \mu_{i,t} - \mathbb{E}[X_{I(t),t}] \tag{2}$$

$$\text{Reg}(T) = \sum_{t=1}^{T} \text{reg}(t) \tag{3}$$

How do we solve it?

Let's try same algorithms we already know?

We will experiment with 100 Arms with bernolli rewards as follows:

- Stationary: $\mu_i = \frac{100-(i-1)}{100}$ for $i \in [100]$
- Gradual: $\mu_{i,1} = \frac{100-(i-1)}{100}$ and $\mu_{i,t} = \frac{T-t+1}{T}\mu_{i,1} + \frac{t-1}{T}\mu_{i,T}$
- Abrupt: change points at $T/3$ and $2T/3$ where

  $\mu_{i,1} = \frac{100-(i-1)}{100}$
  $\mu_{i,T/3} = 1 - \mu_{i,1}$
  $\mu_{i,2T/3} = \mu_{i,1}$

Static Environment

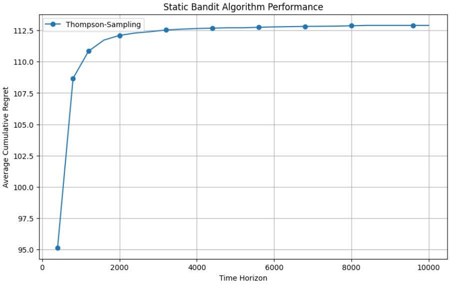
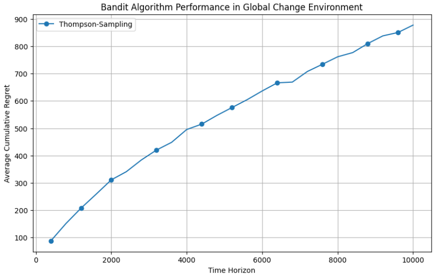
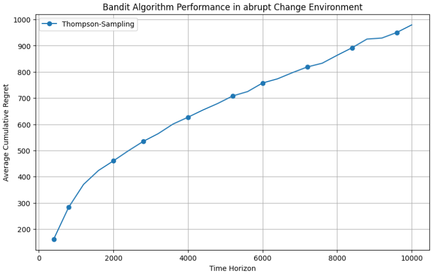


Figure: Thompson Sampling in static environment

Non Static Environment



(a) Gradual



(b) Abrupt

Figure: Thompson Sampling in Non-static environment

Ways to solve it?

- **Passive Methods:** You **do not** seek changes in the data stream while your algorithm runs.
- **Active Methods:** You seek changes in the data stream.

An Example

Algorithm Discounted_UCB

Input: Discount factor $\gamma \in (0, 1)$, exploration parameter ξ
Initialize: For $t = 1, \ldots, K$, play arm $I_t = t$.
for $t = K + 1, \ldots, T$ **do**
    For each arm $i \in \{1, \ldots, K\}$, compute:
        $N_t(\gamma, i) = \sum_{s=1}^{t} \gamma^{t-s} \mathbb{I}_{\{I_s = i\}}$
        $\overline{X}_t(\gamma, i) = \frac{1}{N_t(\gamma,i)} \sum_{s=1}^{t} \gamma^{t-s} X_s(i) \mathbb{I}_{\{I_s = i\}}$
        $n_t(\gamma) = \sum_{k=1}^{K} N_t(\gamma, k)$
        $c_t(\gamma, i) = 2\sqrt{\frac{\xi \log n_t(\gamma)}{N_t(\gamma,i)}}$
    Play arm $I_t = \arg\max_{i \in \{1,\ldots,K\}} \left(\overline{X}_t(\gamma, i) + c_t(\gamma, i)\right)$.
end for

Proposal

Passive Methods fails to perform as good as its vanilla variant. Thus, a good idea would be to detect change points.

ADWIN

- Maintain an adaptive window $W(t)$ of recent observations.
- When some split $W = W_1 \cup W_2$ shows significantly different means, discard the older part W_1 (shrink to W_2).

Algorithm ADWIN

Require: A univariate stream $S : (x_1, x_2, \ldots) \in [0, 1]$, confidence level $\delta \in (0, 1)$.

$W(1) = \{\}$
for $t = 1, 2, \ldots$ **do**
  $W(t+1) = W(t) \cup \{t\}$
  while $|\hat{\mu}_{W_1} - \hat{\mu}_{W_2}| > \epsilon_{cut}^{\delta}$ holds for some split $W(t+1) = W_1 \cup W_2$ **do**
    $W(t+1) = W_2$
  end while
end for

ADWIN

The cut-off value ϵ_{cut}^{δ} is defined as:

$$\epsilon_{cut}^{\delta} = \sqrt{\frac{1}{2|W_1|}\log\left(\frac{1}{\delta}\right)} + \sqrt{\frac{1}{2|W_2|}\log\left(\frac{1}{\delta}\right)}$$

Theorem (Uniform Hoeffding bound for the window set)

Let $p > 0$ be arbitrary.

$$\mathbb{P}\left(\forall W \in \mathcal{W}, |\mu_w - \hat{\mu}_w| \leq \sqrt{\frac{\log(T^{2+p})}{2|W|}}\right) \geq 1 - \frac{2}{T^p}$$

where $\mathcal{W}$ is the set of all possible window segment in time T and $\mu_W = \frac{1}{|W|}\sum_{t \in W} \mu_t$

ADWIN

In theorem 3, if we take $p = 1$ and $\delta = \frac{1}{T^3}$ then we have with probability at least $1 - 2/T$,

$$\forall W \in \mathcal{W}, |\mu_w - \hat{\mu}_w| \leq \sqrt{\frac{\log(T^3)}{2|W|}}$$

Let this be the good event $\mathcal{G}$.

ADWIN

In theorem 3, if we take $p = 1$ and $\delta = \frac{1}{T^3}$ then we have with probability at least $1 - 2/T$,

$$\forall W \in \mathcal{W}, |\mu_w - \hat{\mu}_w| \leq \sqrt{\frac{\log(T^3)}{2|W|}}$$

Let this be the good event $\mathcal{G}$.
Under Stationary environment

$$\begin{aligned} \mu_W = \mu \quad \forall W \in \mathcal{W} \\ \implies |\hat{\mu}_{W_1} - \hat{\mu}_{W_2}| = |\hat{\mu}_{W_1} - \mu - (\hat{\mu}_{W_2} - \mu)| \\ \implies \leq |\hat{\mu}_{W_1} - \mu| + |\hat{\mu}_{W_2} - \mu| \end{aligned}$$

So under good event $\mathcal{G}$, no cuts will be made.

Results

Run ADWIN algorithm with a confidence parameter $\delta = 1/T^3$. The expected total error, $\mathbb{E}[\mathrm{Err}(T)] = \mathbb{E}[\sum_{t=1}^{T} |\hat{\mu}_{W(t)} - \mu_t|]$, is bounded accordingly.

Environment	Assumptions	Error Bound
Static	Stationary stream	$\tilde{O}(\sqrt{T})$
Abrupt	M changepoints	$\tilde{O}(\sqrt{MT})$
Gradual	Change parameter $b = O(T^{-d})$ for $d \in (0, 3/2)$	$\tilde{O}(T^{1-d/3})$

Table: Summary of ADWIN Expected Total Error Bounds ($\delta = 1/T^3$).

ADS

Now that we have ADWIN, how can we use it in bandits?

ADS

Now that we have ADWIN, how can we use it in bandits?

- Have a base bandit algorithm (KL-UCB, Thompson Sampling etc.)

ADS

Now that we have ADWIN, how can we use it in bandits?

- Have a base bandit algorithm (KL-UCB, Thompson Sampling etc.)
- For each arm, maintain an instance of ADWIN to monitor the rewards obtained from that arm.

ADS

Now that we have ADWIN, how can we use it in bandits?

- Have a base bandit algorithm (KL-UCB, Thompson Sampling etc.)
- For each arm, maintain an instance of ADWIN to monitor the rewards obtained from that arm.
- Whenever ADWIN detects a change for an arm, update the statistic of the base bandit algorithm according to new ADWIN window.

ADS

Now that we have ADWIN, how can we use it in bandits?

- Have a base bandit algorithm (KL-UCB, Thompson Sampling etc.)
- For each arm, maintain an instance of ADWIN to monitor the rewards obtained from that arm.
- Whenever ADWIN detects a change for an arm, update the statistic of the base bandit algorithm according to new ADWIN window.

This is the idea of ADS (Adaptive Shrinking Bandits) algorithm.

ADS

Algorithm ADS

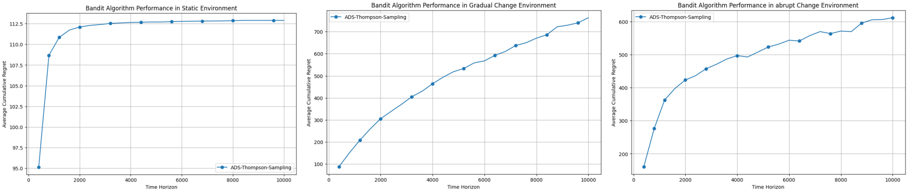
Require: Set of arms $[K]$, confidence level δ, base bandit algorithm.
1: Initialize the base bandit algorithm.
2: Initialize K instances of ADWIN, one for each arm.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: $\quad (I(t), X(t)) \leftarrow \textsc{Base-Bandit}(t)$
5: $\quad$ **if** $\exists$ a split $W = W_1 \cup W_2 : |\hat{\mu}_{W_1} - \hat{\mu}_{W_2}| \geq \epsilon_{cut}^{\delta}$ **then**
6: $\quad\quad$ Update the Window W(t+1) to W2 for base bandit algorithm.
7: $\quad$ **end if**
8: **end for**

ADS

How good is ADS ?

ADS

How good is ADS ?



(a) Static (b) Gradual (c) Abrupt

Figure: ADS-Thompson Sampling in various environment

Empirically, it works well!

ADS

What about theoretical guarantees?

ADS

What about theoretical guarantees? If we consider a stationary environment

ADS

What about theoretical guarantees? If we consider a stationary environment

- Good Set $\mathcal{G} \leftarrow$ no cuts will be made by ADWIN.

ADS

What about theoretical guarantees? If we consider a stationary environment

- Good Set $\mathcal{G} \leftarrow$ no cuts will be made by ADWIN.
- Base bandit algorithm will run on the whole data.

ADS

What about theoretical guarantees? If we consider a stationary environment

- Good Set $\mathcal{G} \leftarrow$ no cuts will be made by ADWIN.
- Base bandit algorithm will run on the whole data.
- Thus, Regret(ADS) = Regret(Base Bandit)

ADS

What about theoretical guarantees? If we consider a stationary environment

- Good Set $\mathcal{G} \leftarrow$ no cuts will be made by ADWIN.
- Base bandit algorithm will run on the whole data.
- Thus, Regret(ADS) $=$ Regret(Base Bandit)
- Since good event $\mathcal{G}$ holds with high probability, we can say that ADS performs almost as well as the base bandit algorithm in stationary environments.

ADS

What about non-stationary environments?

ADS

What about non-stationary environments?
This is hard to analyse.

Contact Us

But, if you think solving it is easy, **contact us** at

sahilc@iisc.ac.in, **prathamgupta@iisc.ac.in**

ADR

Adaptive Resetting (ADR) algorithm add following on top of ADS:

- When ADWIN detects change for any arm then the whole bandit algorithm is reset and starts afresh with new adjusted time horizon.

ADR

Adaptive Resetting (ADR) algorithm add following on top of ADS:

- When ADWIN detects change for any arm then the whole bandit algorithm is reset and starts afresh with new adjusted time horizon.
- We run algorithm a sequence of blocks of exponentially increasing number of sub blocks where each sub block is of length KN and number of sub blocks in block i is 2^{i-1}.

ADR

Adaptive Resetting (ADR) algorithm add following on top of ADS:

- When ADWIN detects change for any arm then the whole bandit algorithm is reset and starts afresh with new adjusted time horizon.
- We run algorithm a sequence of blocks of exponentially increasing number of sub blocks where each sub block is of length KN and number of sub blocks in block i is 2^{i-1}.
- Each block has a monitoring arm which is the arm that occurs most frequently in all but last sub block of that block.

ADR

Adaptive Resetting (ADR) algorithm add following on top of ADS:

- When ADWIN detects change for any arm then the whole bandit algorithm is reset and starts afresh with new adjusted time horizon.
- We run algorithm a sequence of blocks of exponentially increasing number of sub blocks where each sub block is of length KN and number of sub blocks in block i is 2^{i-1}.
- Each block has a monitoring arm which is the arm that occurs most frequently in all but last sub block of that block.
- The monitoring arm of current block is pulled deterministically in the last sub block of that block every K time steps.

ADR

Adaptive Resetting (ADR) algorithm add following on top of ADS:

- When ADWIN detects change for any arm then the whole bandit algorithm is reset and starts afresh with new adjusted time horizon.
- We run algorithm a sequence of blocks of exponentially increasing number of sub blocks where each sub block is of length KN and number of sub blocks in block i is 2^{i-1}.
- Each block has a monitoring arm which is the arm that occurs most frequently in all but last sub block of that block.
- The monitoring arm of current block is pulled deterministically in the last sub block of that block every K time steps.
- The monitoring arm of last block is pulled deterministically every K time steps in the current block.

ADR

Algorithm 5 ADR-bandit

Require: Set of arms $[K]$, confidence level δ, monitoring parameter $N \in \mathbb{N}$, base-bandit algorithm

Initialize base-bandit algorithm.
for $l = 1, 2, \ldots, \lceil \log_2(T/(KN) + 1) \rceil$ **do**
  for $t = (2^{l-1} - 1)KN + 1, \ldots, \min\{(2^l - 1)KN, T\}$ **do**
    if $l \geq 2$ and $t = 0 \pmod{K}$ **then**
      Pull $i^{(l-1)}$. ▷ Monitoring arm of the previous block.
    else if $l \geq 2$ and $t = 1 \pmod{K}$ and $t \geq (2^{l-1} - 2)KN + 1$ **then**
      Pull $i^{(l)}$. ▷ Monitoring arm of the current block.
    else
      $(I(t), X(t)) = \text{Base-bandit}(W(t))$ ▷ Do one step of the base-bandit algorithm
    end if
    if there exists $W(t+1) = W_1 \cup W_2$ such that $|\hat{\mu}_{i,W_1} - \hat{\mu}_{i,W_2}| \geq \epsilon_{\text{cut}}^{\delta}$ **then**
      Reset the entire algorithm with $T := T - t$.
    end if
    if $t = KN$ **then** ▷ The end of the first block $l = 1$.
      Set $i^{(1)} = \arg\max_{i \in [K]} N_i^{(1)}$ ▷ See Eq. (20) for $N_i^{(1)}$.
    else if $l \geq 2$ and $t = (2^{l-1} - 2)KN$ **then** ▷ Before the beginning of the last subblock of the block l.
      Set $i^{(l)} = \arg\max_{i \in [K]} N_i^{(l)}$. ▷ See Eq. (20) for $N_i^{(l)}$
    end if
  end for
end for

ADR

Let us parse what is happening here.

ADR

Let us parse what is happening here.

- Line 4-9 : Determine if we have to pull a monitoring arm or run base bandit algorithm.

ADR

Let us parse what is happening here.

- Line 4-9 : Determine if we have to pull a monitoring arm or run base bandit algorithm.
- Line 11-12 : ADWIN check with resetting of base bandit algorithm if change is detected.

ADR

Let us parse what is happening here.

- Line 4-9 : Determine if we have to pull a monitoring arm or run base bandit algorithm.
- Line 11-12 : ADWIN check with resetting of base bandit algorithm if change is detected.
- Line 14-17 : Evaluation of monitoring arms for the blocks.

ADR

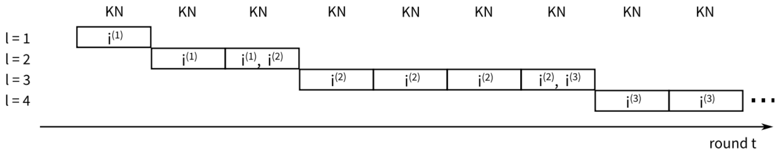


Figure: Illustration of sub blocks and monitoring arms in ADR

Here $i^{(l)}$ is monitoring arm of block l.

ADR Analysis

ADR Analysis

Static Case:

ADR Analysis

Static Case:

- We have already seen that under good event $\mathcal{G}$, no cuts will be made by ADWIN.

ADR Analysis

Static Case:

- We have already seen that under good event $\mathcal{G}$, no cuts will be made by ADWIN.
- So now we need to analyse the regret incurred due to monitoring arms.

ADR Analysis

Static Case:

- We have already seen that under good event $\mathcal{G}$, no cuts will be made by ADWIN.
- So now we need to analyse the regret incurred due to monitoring arms.
- Basic Idea as T grows the most frequently pulled arm will be the optimal arm so optimal arm will be monitoring arm in last block with high probability.

ADR Analysis : Static

So we know that under good event $\mathcal{G}$ no cuts will be made by ADWIN.

- $i^{(l)} \longleftarrow$ monitoring arm of block l.
- $\mathcal{T}_l^{\text{monitor}} \longleftarrow \{t : i^{(l)}\text{is pulled at time } t \text{ as l-th monitoring arm}\}$.

ADR Analysis : Static

So we know that under good event $\mathcal{G}$ no cuts will be made by ADWIN.

- $i^{(l)} \longleftarrow$ monitoring arm of block l.
- $\mathcal{T}_l^{\text{monitor}} \longleftarrow \{t : i^{(l)}\text{is pulled at time } t \text{ as l-th monitoring arm}\}$.

We want to bound the regret during T time steps.

$$\sum_t \text{reg}(t) = \sum_{t \in \mathcal{T}^{\text{base}}} \text{reg}(t) + \sum_l \sum_{t \in \mathcal{T}_l^{\text{monitor}}} \text{reg}(t)$$

ADR Analysis : Static

So we know that under good event $\mathcal{G}$ no cuts will be made by ADWIN.

- $i^{(l)} \longleftarrow$ monitoring arm of block l.
- $\mathcal{T}_l^{\text{monitor}} \longleftarrow \{t : i^{(l)}\text{is pulled at time } t \text{ as l-th monitoring arm}\}$.

We want to bound the regret during T time steps.

$$\sum_t \text{reg}(t) = \sum_{t \in \mathcal{T}^{\text{base}}} \text{reg}(t) + \sum_l \sum_{t \in \mathcal{T}_l^{\text{monitor}}} \text{reg}(t)$$

$$= \sum_l \left(\sum_{t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l} \text{reg}(t) + \sum_{t \in \mathcal{T}_l^{\text{monitor}}} \text{reg}(t) \right)$$

ADR Analysis : Static

So we know that under good event $\mathcal{G}$ no cuts will be made by ADWIN.

- $i^{(l)} \longleftarrow$ monitoring arm of block l.
- $\mathcal{T}_l^{\text{monitor}} \longleftarrow \{t : i^{(l)} \text{is pulled at time } t \text{ as l-th monitoring arm}\}$.

We want to bound the regret during T time steps.

$$\sum_t \text{reg}(t) = \sum_{t \in \mathcal{T}^{\text{base}}} \text{reg}(t) + \sum_l \sum_{t \in \mathcal{T}_l^{\text{monitor}}} \text{reg}(t)$$
$$= \sum_l \left(\sum_{t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l} \text{reg}(t) + \sum_{t \in \mathcal{T}_l^{\text{monitor}}} \text{reg}(t) \right)$$

Can we bound $\left(\sum_{t \in \mathcal{T}_l^{\text{monitor}}} \text{reg}(t) \right)$ in terms of $\sum_{t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l} \text{reg}(t)$?

ADR Analysis : Static

Suppose

$$\frac{\sum_t 1[t \in \mathcal{T}_l^{\text{monitor}}]}{\sum_t 1[t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l]} 1[I(t) = i^{(l)}] \leq \mathbf{C}$$

ADR Analysis : Static

Suppose

$$\frac{\sum_t 1[t \in \mathcal{T}_l^{\text{monitor}}]}{\sum_t 1[t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l]} 1[I(t) = i^{(l)}] \leq \mathbf{C}$$

Then,

$$\sum_{t \leq S} \text{reg(t)} = \sum_{l} \left(\sum_{t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l} \text{reg}(t) + \sum_{t \in \mathcal{T}_l^{\text{monitor}}} \text{reg}(t) \right)$$

ADR Analysis : Static

Suppose

$$\frac{\sum_t 1[t \in \mathcal{T}_l^{\text{monitor}}]}{\sum_t 1[t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l]} 1[I(t) = i^{(l)}] \leq \mathbf{C}$$

Then,

$$\begin{aligned}
\sum_{t \leq S} \text{reg(t)} &= \sum_{l} \left(\sum_{t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l} \text{reg}(t) + \sum_{t \in \mathcal{T}_l^{\text{monitor}}} \text{reg}(t) \right) \\
&= \sum_{l} \left(\sum_{t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l} \Delta_{I(t)} + \sum_{t \in \mathcal{T}_l^{\text{monitor}}} \Delta_{i^{(l)}} \right)
\end{aligned}$$

ADR Analysis : Static

Suppose

$$\frac{\sum_t 1[t \in \mathcal{T}_l^{\text{monitor}}]}{\sum_t 1[t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l]} 1[I(t) = i^{(l)}] \leq \mathbf{C}$$

Then,

$$\begin{aligned}
\sum_{t \leq S} \text{reg(t)} &= \sum_l \left(\sum_{t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l} \text{reg}(t) + \sum_{t \in \mathcal{T}_l^{\text{monitor}}} \text{reg}(t) \right) \\
&= \sum_l \left(\sum_{t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l} \Delta_{I(t)} + \sum_{t \in \mathcal{T}_l^{\text{monitor}}} \Delta_{i^{(l)}} \right) \\
&\leq \sum_l \left(\sum_{t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l} \Delta_{I(t)} + \mathbf{C} \sum_{t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l} \Delta_{i^{(l)}} 1[I(t) = i^{(l)}] \right)
\end{aligned}$$

ADR Analysis : Static

$$\sum_{t \leq S} \mathrm{reg(t)} \leq (\mathbf{C}+1) \sum_{I} \sum_{t \in \mathcal{T}^{\mathrm{base}} \cap \mathcal{T}_I} \Delta_{I(t)}$$

ADR Analysis : Static

$$\sum_{t \leq S} \text{reg(t)} \leq (\mathbf{C}+1) \sum_{l} \sum_{t \in \mathcal{T}^{\text{base}} \cap \mathcal{T}_l} \Delta_{I(t)}$$
$$\leq (\mathbf{C}') \sum_{t \in \mathcal{T}^{\text{base}}} \Delta_{I(t)}$$

Since base bandit algorithm has logarithmic Regret,

$$\mathbb{E}[\text{Reg}(T)] \leq \mathbf{C}^{st} \sum_{i \neq 1} \frac{\log T}{\Delta_i} + O(1)$$

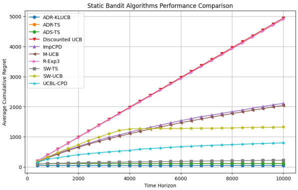
ADR Analysis: Abrupt

May be after class?

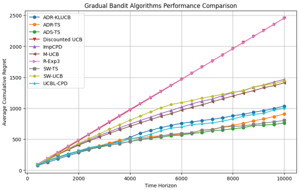
Environment	Regret Bound
Abrupt	$\mathbb{E}[\mathrm{Reg}(T)] = \tilde{O}(\sqrt{MKT})$
Gradual	$\mathbb{E}[\mathrm{Reg}(T)] = \tilde{O}(\sqrt{K}T^{1-d/3})$

Table: Summary of ADR-bandit Regret Bounds

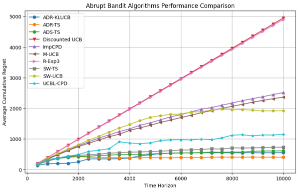
Experiments



(a) Static



(b) Gradual



(c) Abrupt

Figure: Regret using various bandit algorithms in non stationary environments

Please feel free to experiment using our code at
PrathamGupta423/Nonstationary-Multi-Armed-Bandits

References I

[1] João Gama et al. "A survey on concept drift adaptation". In: *ACM Comput. Surv.* 46.4 (Mar. 2014). ISSN: 0360-0300. DOI: 10.1145/2523813. URL: https://doi.org/10.1145/2523813.

[2] Junpei Komiyama, Edouard Fouché, and Junya Honda. "Finite-time Analysis of Globally Nonstationary Multi-Armed Bandits". In: *Journal of Machine Learning Research* 25.112 (2024), pp. 1–56. URL: http://jmlr.org/papers/v25/21-0916.html.

Thank You!