# Evaluating Performance of Bandit Algorithms in Non-stationary Environments


Author: Weicheng Hu

Supervisor: Julia Olkhovskaia, Department of Sequential Decision Making, EEMCS


## Background

**Bandit problems:** a decision-making scenario where an agent must choose from multiple options over time to maximize cumulative rewards.

**Contextual bandit problems:** Before the player makes a choice, the player could observe the arm vectors to make more informed choices. However, there exists some hidden vectors to affect the final reward.

**Challenges:** Most of the literatures assume the environment is stationary. However, in real-world scenarios it might not be the case, on the opposite, the hidden vector could change unexpectedly[1].

**Question**: Which bandit algorithm adapts well and performs better in non-stationary environment, out of a selection of algorithms?

**Performance** is measured by Cumulative Regret, which is the difference between the rewards of the best possible policy and the policy used by the agent[2].

## Methodologies

Methodology and Background

- Used Python 3.9 and its popular libraries: Pandas for data preprocessing, Matplotlib for data visualization.
- SMPyBandits framework was used and modified to support contextual environments.

Formal Problem Description

- The experiment involves a linear contextual bandit setup with multiple rounds.

Algorithms Evaluated

- UCB: Upper confidence bounds for balancing exploration and exploitation.
- EXP3: Designed for adversarial settings, adapts to observed rewards probabilistically.
- LinUCB: Models reward as a linear function of context vector, uses upper confidence bounds.
- LinEXP3: Combines EXP3 and LinUCB, using Bayesian inference.

## Experimental Setup and Results

Environment: Artificial data simulating non-stationary environments (trigonometric and logarithmic reward vectors).

Algorithm Setup:

- UCB1: No additional parameters, fixed formula.
- EXP3: $\gamma = 0.15$.
- LinUCB: $\alpha = 0.3$.
- LinEXP3: $\eta = 0.5$, $\gamma = 0.2$, $M = 1500$ and $\beta = 0.5$.

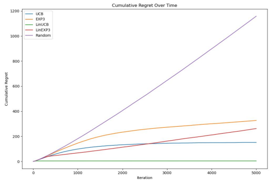
Environment Setup:

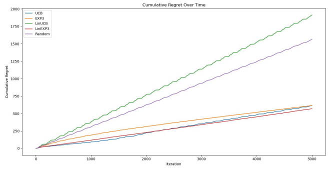
1. Trigonometric hidden vectors to simulate periodic changes
2. Logarithmic environments to simulate positive slowly changing environment

Ten 3d distributions with different characteristics, 5000 iterations with 20 realizations, performance evaluated by average cumulative regret.

Results:

| Algorithm | Avg. Computation Time |
|---|---|
| UCB1 | 0.02 |
| LinUCB | 0.08 |
| EXP3 | 0.02 |
| LinEXP3 | 0.87 |
| Random Policy | 0.01 |




## Conclusion

UCB1: Good as a baseline agent, suitable for stable environments. However it does not use contextual information which makes it to converge slower.
EXP3: Good adaptation in both stationary and non-stationary environment, allows false tolerance if the environment is unknown; though it might not converge.
LinUCB: Best in stable contextual environments, uses contextual information, but performs terrible if the optimal arm is changing.
LinEXP3: Best in non-stationary contextual environments, uses contextual information, performance is also good in stationary environments, but high computational cost.

Answer to the research question: There is no universal optimal algorithm unfortunately, the best algorithm still depends on the nature of the environments and the requirement of the player.

## Discussion and Future Works

Discussions:

1. Ture the values for the environments
2. Use more arbitrary changing functions, not only limited to changes in time, but some other params as well.
3. Introduce covariance matrices for context vectors to simulate more realistic learning processes.

Future works:

1. Incorporate a broader range of algorithms like Contextual Thompson Sampling.
2. Validate algorithms using real-world datasets.

## References


[1] G. Neu and J. Olkhovskaya, “Efficient and robust algorithms for adversarial linear contextual bandits,” in Conference on Learning Theory. PMLR, 2020, pp. 3049–3068
[2] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” Machine learning, vol. 47, no. 2-3, pp. 235–256, 2002.