Comparing bandit algorithms in static and changing environments

Author: Cody Boon
Email: cody.m.boon@gmail.com
Supervisor: Julia Olkhovskaya

1. Background

Multi-armed bandit problems are problems where a solver has to pick from a set of **arms** (actions) repeatedly for a set number of times without knowing the (distribution of the) rewards of each arm [1].
Contexts are vectors **x** which are used in contextual environments by an arm's reward function, often in combination with some hidden weights vector **θ*** [2].
Optimal Policy Regret is the difference between an algorithms' achieved reward and that of a supposed optimal policy [1].

2. Research Question

What is the difference in regret performance between different bandit algorithms in stochastic, static contextual and non-static contextual environments.

3. Methodology

The following algorithms are being compared:

- **UCB** [1]
- **EXP3** [3]
- **LinUCB** [2]
- **CW-OFUL** [4]
- **SW-UCB** [5]

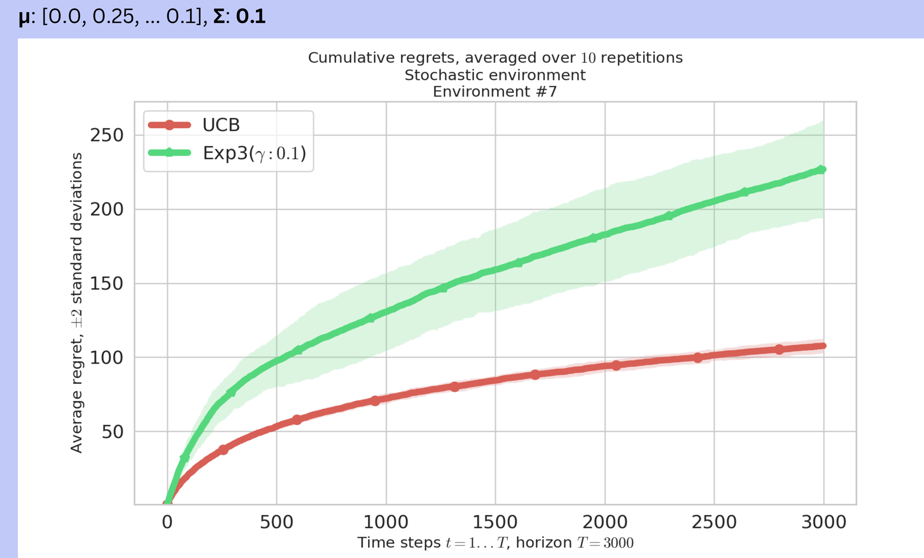
These algorithms are ran and compared in various environments:

- **Stochastic with static reward distributions**
- **Contextual with static θ***
- **Contextual with gradually changing θ***
- **Contextual with perturbed θ***

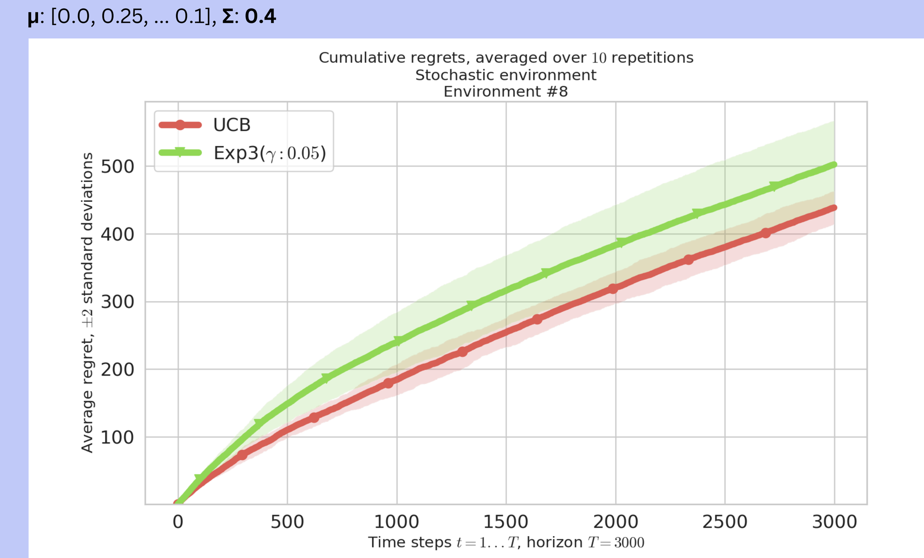
All three contextual environments have static noise and context distributions.
The contexts and rewards are pre-generated to make sure all algorithms are ran against the same exact randomly generated data.
LinUCB, CW-OFUL and SW-UCB have been excluded from the first test due to their incompatibility with the environment

4a. Data

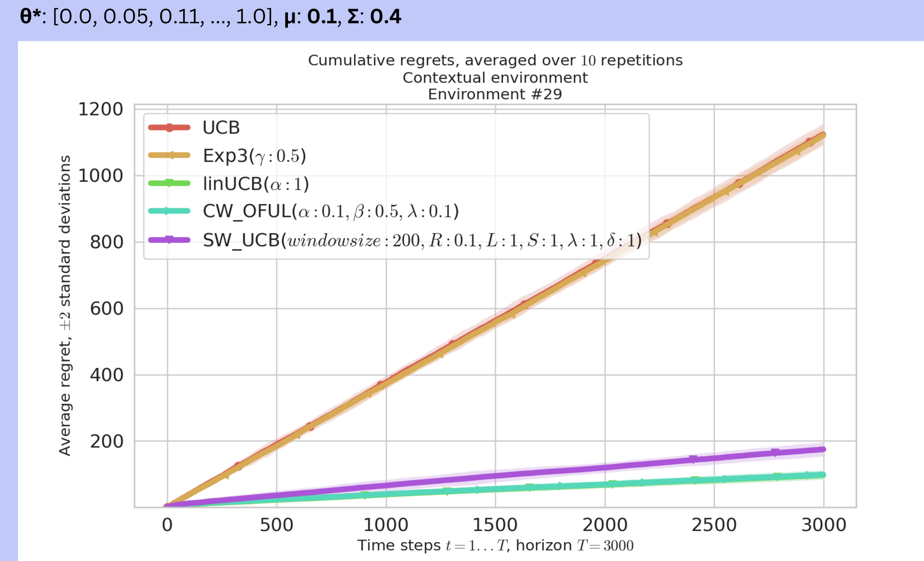
μ: [0.0, 0.25, ... 0.1], **Σ**: **0.1**



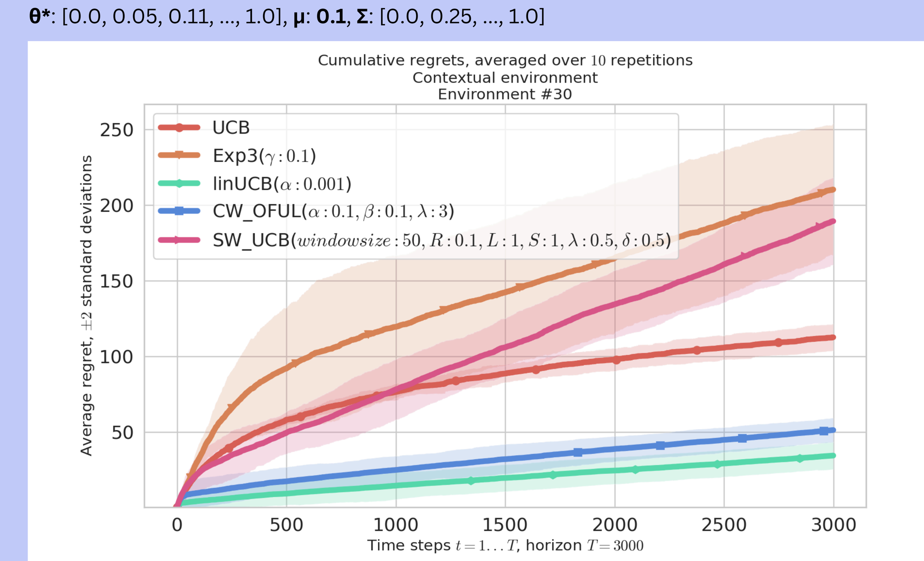
μ: [0.0, 0.25, ... 0.1], **Σ**: **0.4**



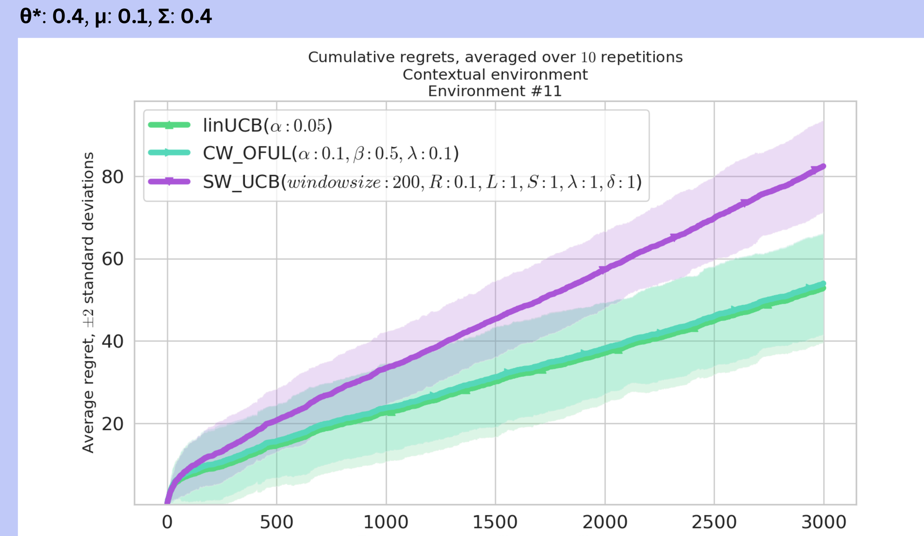
θ*: [0.0, 0.05, 0.11, ..., 1.0], **μ**: **0.1**, **Σ**: **0.4**



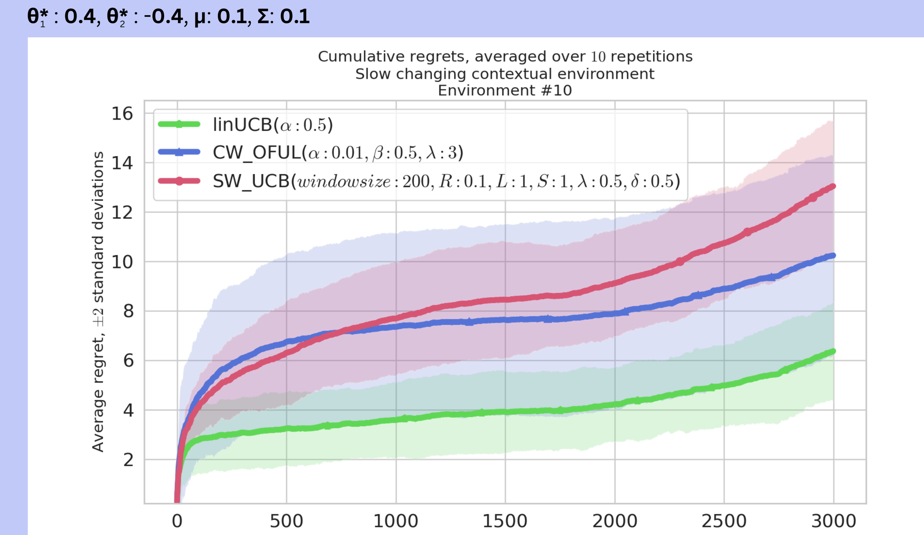
θ*: [0.0, 0.05, 0.11, ..., 1.0], **μ**: **0.1**, **Σ**: [0.0, 0.25, ..., 1.0]



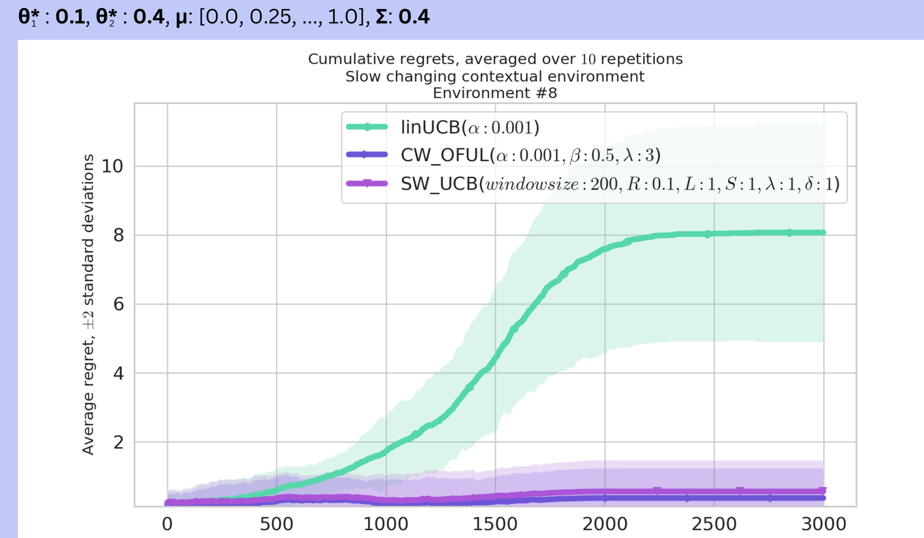
θ*: **0.4**, **μ**: **0.1**, **Σ**: **0.4**



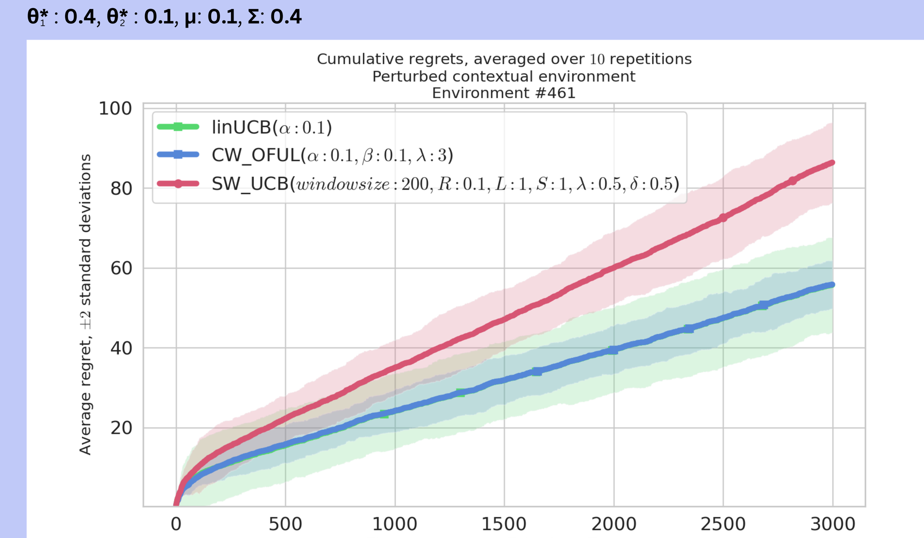
θ^*_1 : 0.4, **θ^*_2 : -0.4**, **μ**: **0.1**, **Σ**: **0.1**



θ^*_1 : 0.1, **θ^*_2 : 0.4**, **μ**: [0.0, 0.25, ..., 1.0], **Σ**: **0.4**



θ^*_1 : 0.4, **θ^*_2 : 0.1**, **μ**: **0.1**, **Σ**: **0.4**



4b. Results

All contextual environments:

- Stochastic algorithms nearly always gather far more regret over time than contextual ones in a contextual environment

Static contextual environments:

- The performance of the contextual algorithms is generally very similar in static contextual environments.
- Both linUCB and CW-OFUL seem to be very consistent in their performance
- SW-UCB sometimes performs far worse.

Non-static contextual environments:

- linUCB and SW-UCB show high regret values in certain, rare, configurations, but never in the same environment. The cause is unclear.
- CW-OFUL is consistently (one of) the algorithm(s) with the lowest regret.

5. Conclusion

In static contextual environments, linUCB performs the best, generally slightly better than CW-OFUL and far better than the others. In non-static contextual environments CW-OFUL tends to perform better.

6. References

[1] T. Lattimore and C. Szepesv´ari, Bandit algorithms. Cambridge University Press, 2020.
[2] W. Chu, L. Li, L. Reyzin, and R. Schapire, “Contextual bandits with linear payoff functions,” in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2011, pp. 208–214
[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, “The nonstochastic multiarmed bandit problem,” SIAM journal on computing, vol. 32, no. 1, pp. 48–77, 2002
[4] J. He, D. Zhou, T. Zhang, and Q. Gu, “Nearly optimal algorithms for linear contextual bandits with adversarial corruptions,” Advances in neural information processing systems, vol. 35, pp. 34 614–34 625, 2022
[5] W. C. Cheung, D. Simchi-Levi, and R. Zhu, “Learning to optimize under non-stationarity,” in The 22nd International Conference on Artificial Intelligence and Statistics, PMLR, 2019, pp. 1079–1087