Influence of Delay on Contextual Multi-Armed Bandits

Author: Dragos-Cristian Arsene<D.C.Arsene-2@student.tudelft.nl>
Supervisor: Dr. Julia Olkhovskaia<I.M.Olkhovskaia@tudelft.nl>

01 Introduction

- An agent chooses actions over discrete time steps, each action having a reward distribution
- The objective is to minimise regret
- The key challenge is to balance exploration and exploitation
- Real-life applications often introduce delays in reward feedback

02 Research Questions

How does delay affect the cumulative regret of algorithms run in contextual settings?
How can hyperparameters be chosen to mitigate this effect?

03 Methodology

Metric used for comparing the performance of algorithms:

$CR_T = \sum_{t=1}^{T}(r_t^* - r_{a_t})$

Data is generated artificially and simulates real-life scenarios. In particular, the delay has to be modelled using a discrete distribution over positive values.

The SMPyBandits module [1] runs four algorithms: UCB [2], Exp3 [3], LinUCB [4], and OTFLinUCB [5] for different hyperparameters and environments. A run consists of 50 repetitions, after which the results are averaged and plotted. The time window size hyperparameter is of most interest, as it has the biggest effect on CR

04 Results

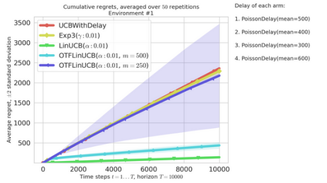

Fig. 1

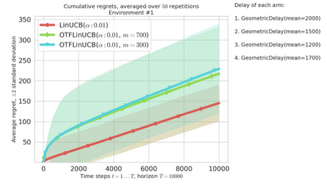

Fig. 2

The tables on the left exhibit how OTFLinUCB performs when run in a delayed setting. The other three algorithms are used as a baseline, being run in the same environment but with no delay.

- Fig. 1 uses Poisson delays, while Fig. 2 uses Geometric delays.
- In Fig. 1, two instances of the OTFLinUCB algorithm with different time window sizes (m) are compared against the baselines, showing that a smaller m significantly impacts the CR.
- In Fig. 2, a bigger difference in time window sizes produces a much smaller difference in CR.
- Choosing a suitable value for m depends entirely on the probability distributions that model delay.

05 Discussion

To study how delay influences cumulative regret, the rho variable is introduced.

Table 1: Conversion rates and their corresponding regret(Part 1)

ρ	100%	90%	80%	70%	60%
CR	35.836	36.057	36.29	37.472	43.264

Table 2: Conversion rates and their corresponding regret(Part 2)

ρ	50%	40%	30%	20%	10%
CR	45.904	51.06	55.091	66.128	105.59

Table 3: Values of m for different Conversion Rates and Discrete Distributions (Part 1)

	Conversion Rate				
Probability Distribution of Delay	100%	90%	80%	70%	60%
Poisson($\lambda = 500$)	∞	529	519	512	506
Poisson($\lambda = 1000$)	∞	1041	1027	1016	1008
Geometric($p = 0.002$)	∞	1151	804	602	458
Geometric($p = 0.001$)	∞	2302	1609	1204	916
NegBin($r = 5, p = 0.01$)	∞	793	666	584	519
Uniform(0, 1000)	1000	900	800	700	600

Table 4: Values of m for different Conversion Rates and Discrete Distributions (Part 2)

	Conversion Rate				
Probability Distribution of Delay	50%	40%	30%	20%	10%
Poisson($\lambda = 500$)	500	494	488	481	471
Poisson($\lambda = 1000$)	1000	992	983	973	960
Geometric($p = 0.002$)	347	256	179	112	53
Geometric($p = 0.001$)	693	511	357	224	106
NegBin($r = 5, p = 0.01$)	462	410	359	305	240
Uniform(0, 1000)	500	400	300	200	100

$$\rho_a = \frac{\text{number of observed rewards for arm } a}{\text{number of total rewards for arm } a}$$

$$\rho_a = F_{X_a}(m) = \Pr(X_a < m)$$

$$m = Q(\rho_a) = F_{X_a}^{-1}(\rho_a)$$

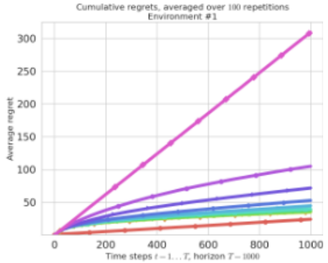

Fig. 3

Fig. 3 shows the cumulative regrets for conversion rates of 0%, 10%, ... up to 100%

- The number of ignored rewards relative to the total number of rewards is one of the two causes of increased cumulative regret due to delay and the only cause that can be mitigated by tuning the time window size. It can be seen in Tables 1 and 2 that this relation is not linear.
- Tables 3 and 4 show time window sizes for different conversion rates and probability distributions.
- A conversion rate can result in vastly different time window sizes, depending on the shape of the probability distribution modelling delay.
- Each arm's distribution can differ, requiring a time window size which ensures all arms' rewards are observed enough times.
- Setting the global window size as the maximum of all arms' window sizes ensures no arm has a conversion rate lower than 70%, which minimizes cumulative regret and optimizes memory usage.

05 Conclusion and future work

- The study discusses how the probability distribution modelling delay can be used to calculate an appropriate window size that balances cumulative regret and memory usage.
- Results suggest a 70% conversion rate does not significantly increase cumulative regret compared to 100%, while decreasing memory usage.
- It would be valuable to see how the uncertainty introduced by delay distribution estimation can be considered when calculating the time window size.

06 References

[1] L. Besson, SMPyBandits: an Open-Source Research Framework for Single and Multi-Players Multi-Arms Bandits (MAB) Algorithms in Python, Online at: GitHub . com / SMPyBandits / SMPyBandits, Available: https://github.com/SMPyBandits/SMPyBandits/.

[2] Lattimore and C. Szepesvári. Bandit algorithms, pages 102–116. Cambridge University Press, 2020.

[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The non-stochastic multiarmed bandit problem. SIAM Journal on Computing, 32(1):48–77, 2002.

[4] 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, pages 208–214. JMLR Workshop and Conference Proceedings, 2011.

[5] C. Vernade, A. Carpentier, T. Lattimore, G. Zappella, B. Ermis, and M. Brueckner. Linear bandits with stochastic delayed feedback. In International Conference on Machine Learning, pages 9712–9721. PMLR, 2020.