IMPERIAL

Learning to Bid Above a Threshold Using Multi-Armed Bandits

Quentin Leconte,
Supervisor: Ciara Pike-Burke

1. Problem

Situation: During an auction, T products are offered one after the other. A threshold τ_t is associated with each product, for round $t = 1, \dots T$. Only one bid b_t can be made per round, and the product is won if it is above the threshold Figure 1. How can the number of products won be maximised while minimising the value of the bids?

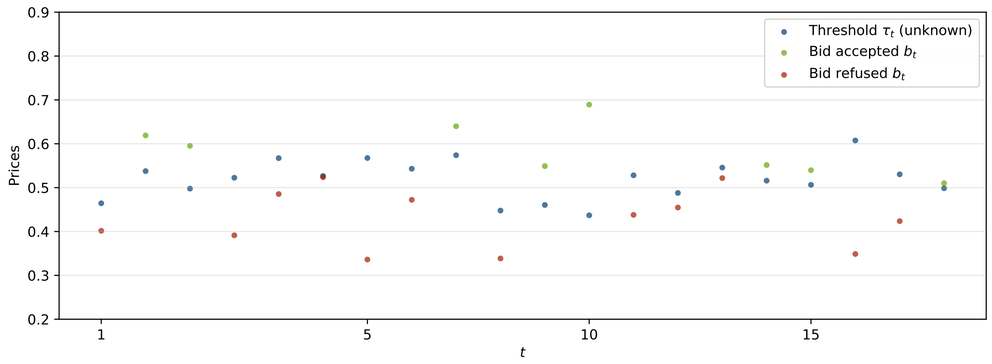

Figure 1: Acceptance and rejection of bids b_t based on thresholds τ_t.

- The bid values and thresholds range between 0 and 1.
- The thresholds are i.i.d and $\tau_t \sim \mathcal{N}(\tau, \sigma^2)$, unknown parameters.
- Only the information $\delta_t = \mathbb{I}_{\{b_t \geq \tau_t\}}$ is known after making the bid b_t.
- The interval $[0, 1]$ is partitioned into J sub-intervals of equal size. Playing arm j at round t is then equivalent to making a bid $b_t^{(j)}$ uniformly in $\left[a_0^{(j)}, a_1^{(j)}\right]$, where $a_0^{(j)} = \frac{j-1}{J}$ and $a_1^{(j)} = \frac{j}{J}$.
- The reward associated with each round (after playing arm j) is

$$X_t^{(j)} = \delta_t^{(j)}(1 - b_t^{(j)}).$$

2. Strategies for Selecting the Arm to Maximise Reward

- For each arm j, the expected reward (independent of t) $\mu_j = \mathbb{E}\left[X^{(j)} | \tau, \sigma\right]$, given the parameters of the threshold distribution, can be calculated. The expected rewards are plotted for $J = 1000$ and some values of τ and σ in Figure 2.

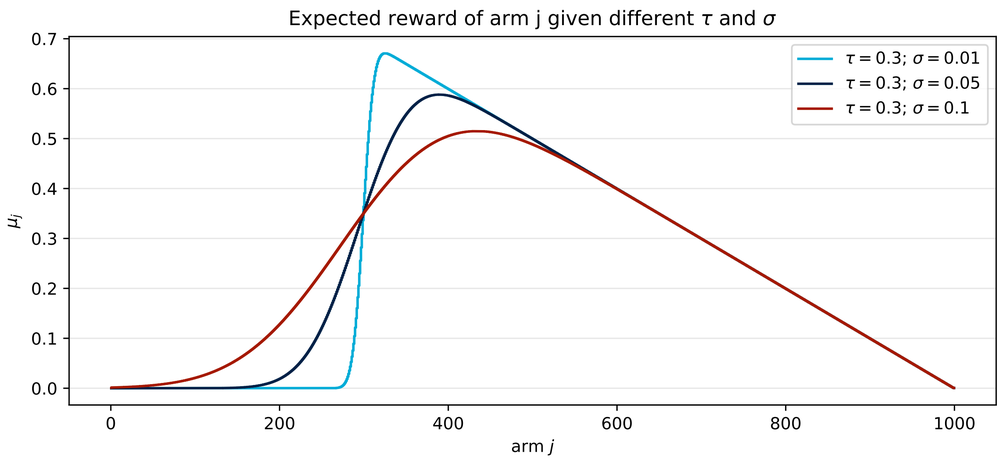

Figure 2: Barplots of expected rewards for 1000 arms partitioning $[0, 1]$ for different threshold distribution generated from $\mathcal{N}(\tau, \sigma^2)$ ($\tau = 0.3$ and $\sigma \in \{0.01, 0.05, 0.1\}$). The smaller σ is, the closer the optimal arm is to $\tau \times J$.

- After playing arm j, the empirical expected reward

$$\hat{\mu}_t^{(j)} = \frac{1}{\max\left(1, N_j(t)\right)} \sum_{s=1}^{t} X_s^{(j)} \mathbb{I}_{\{J_s = j\}}$$

is updated, where J_s is the arm played in round s and $N_j(t) = \sum_{s=1}^{t} \mathbb{I}_{\{J_s = j\}}$.

3. Algorithm

- The approach presented is strongly inspired by the sequential halving algorithm[1]. As shown in Figure 2, the function $j \to \mu_j$ is a step function with a single maximum. The algorithm is divided into episodes of three rounds and aims to find an increasingly precise bracket for the optimal arm j^* in each episode. At each episode, a left arm, middle arm, and right arm are repeatedly assigned to get as close as possible to the optimal arm. The assignment of arms is described in Algorithm 1 and illustrated in Figure 3.

Algorithm 1 Adapted Sequential Halving Algorithm

Input: n_{ep} ▷ number of times an arm is played per episode.
Initialize: Episode 0: $L, M, R \leftarrow 1, \lfloor K/2 \rfloor, K$;
for Episode = 1, ... **do**
  Play each arm n_{ep} times
  t_{ep} is the final round number of episode
  if $\hat{\mu}_{t_{ep}}^{L} < \hat{\mu}_{t_{ep}}^{M}$ and $\hat{\mu}_{t_{ep}}^{R} < \hat{\mu}_{t_{ep}}^{M}$: **then**
    $L, M, R \leftarrow \lfloor (L + M)/2 \rfloor, M, (R + M)/2$
  end if
  if $\hat{\mu}_{t_{ep}}^{L} < \hat{\mu}_{t_{ep}}^{M} < \hat{\mu}_{t_{ep}}^{R}$: **then**
    $L, M, R \leftarrow M, R, 2R - M$
  end if
  if $\hat{\mu}_{t_{ep}}^{L} > \hat{\mu}_{t_{ep}}^{M} > \hat{\mu}_{t_{ep}}^{R}$: **then**
    $L, M, R \leftarrow 2L - M, L, M$
  end if ▷ Certain cases must be handled differently when the boundary arms are reached.
end for

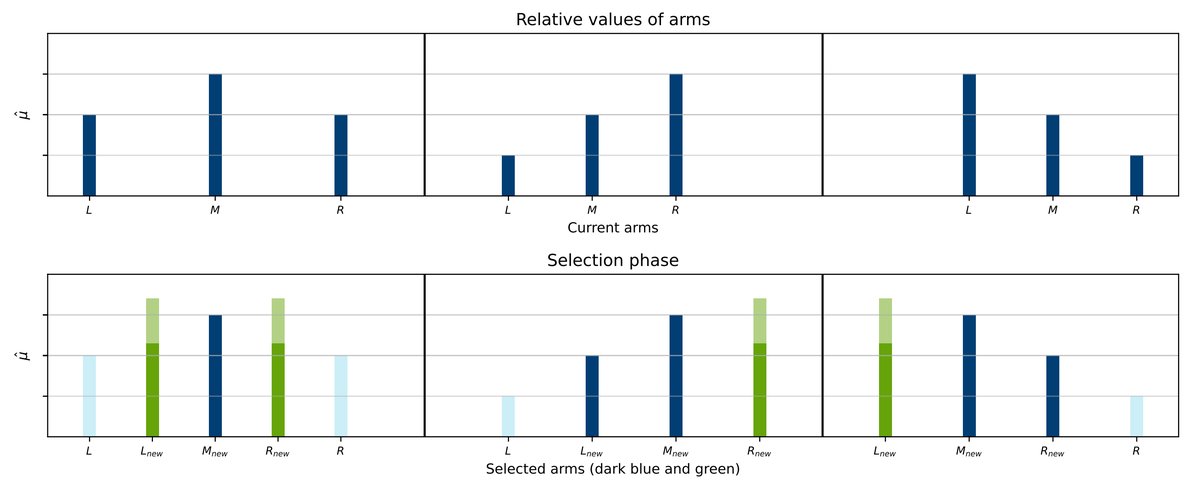

Figure 3: Selection of new left, middle, and right arms based on the empirical expected reward relative values of the current left, middle, and right arms.

- The regret of an algorithm over t rounds is

$$\mathfrak{R}_t(\pi) = \sum_{s=1}^{t} (\mu^* - \mu_{J_s}),$$

where $\mu^* = \max_{j=1,\dots J} \mu_j$ is the expected reward of the optimal arm. The algorithm seeks to minimize its regret. (Figure 4).

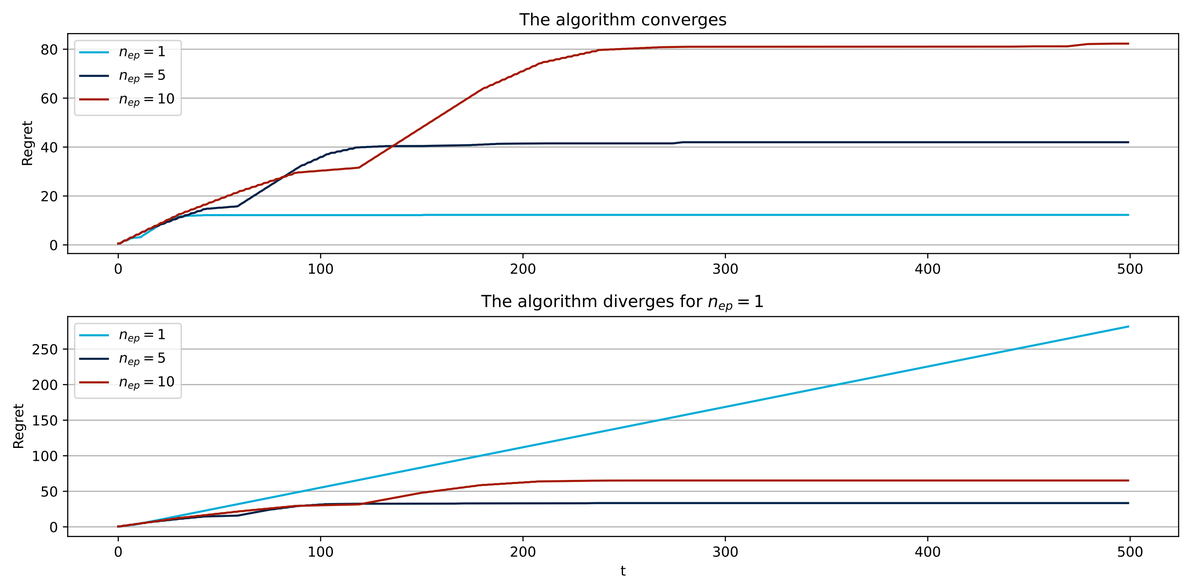

Figure 4: Regret plot of adapted sequential halving theorem for $J = 50, \tau = 0.4, \sigma = 0.01$. Several simulations were done until convergence or divergence. The algorithm is not stable, especially when n_{ep} is low. Two examples of regret plots show the instability of the algorithm based on the input data and that modifications need to be made to ensure convergence and good performance.

References

[1] Cheshire, James and Menard, Pierre and Carpentier, Alexandra (2021) Problem Dependent View on Structured Thresholding Bandit Problems, In Proceedings of the 38th International Conference on Machine Learning, p1846–1854

Imperial College London
quentin.leconte23@imperial.ac.uk