BTP - I

Developing and analyzing algorithms for the Multi-armed Bandit

Sanit Gupta

Supervised by: Prof. Shivaram Kalyanakrishnan

Indian Institute of Technology Bombay
Mumbai 400 076

Contents

1 Problem Description

In the multi-armed bandit problem, an agent must choose to pull one of n available arms. Each arm has a reward distribution associated with it. These distributions are fixed but unknown.

We study the bernoulli bandit problem where each the reward distribution for each arm i is assumed to be a bernoulli distribution with parameter μ_i.

We study the regret minimization setting for the MAB problem. Regret is defined as how much better in terms of cumulative reward one could have done on the bandit instance. The expected regret can be written as:

$$E[R(T)] = \mu^* T - E[\sum_{t=1}^{T} r(t)]$$

where $R(T)$ is the cumulative regret in T time steps and $r(t)$ is the reward received in the t^{th} time step

Previously, Lai and Robbins [LR85] gave lower bounds on regret for all bandit algorithms:

$$E[R(T)] \geq [\Sigma_{i:\mu_i<\mu^*} \frac{\Delta_i}{D(\mu_i||\mu^*)} + o(1)]lnT$$

where D is KL divergence

Some popular algorithms for this setting that match the lower bound are UCB1 and Thompson Sampling.

Upper bound on expected regret for UCB1 from [ACF02]:

$$E[R(T)] \leq 8[\Sigma_{i:\mu_i<\mu^*} \frac{lnT}{\Delta_i}] + (1 + \frac{\pi^2}{3})(\Sigma_{j=1}^{K} \Delta_j)$$

Upper bound on expected regret of Thompson Sampling from [KKM12]:

$$E[R(T)] \leq (1+\epsilon)\Sigma_{i:\mu_i<\mu^*} \frac{\Delta_i(ln(T) + ln(ln(T)))}{D(\mu_i||\mu^*)} + C(\epsilon, \mu_1, .., \mu_n)$$

where ϵ and C are problem dependent constants

2 Key Idea - Persistence

Algorithm 1 Persistence Variant of Bandit_Algorithm

1: $n \leftarrow Number\ of\ Arms$
2: $T \leftarrow Time\ Horizon$
3: **for** i = 1 to n **do**
4: $\quad true\_reward\_distribution[i] \leftarrow Bernoulli(\mu_i)$
5: $reward\_history \leftarrow [\ ]$
6: $action\_history \leftarrow [\ ]$
7:
8: $r \leftarrow 0$
9: **for** i = 1 to T **do**
10: $\quad$ **if** $r == 0$ **then** //remove this condition $->$ regular bandit algorithm
11: $\quad\quad action\_choice \leftarrow bandit\_algorithm(action\_history, reward\_history)$
12: $\quad r \leftarrow sample(true\_reward\_distribution[action\_choice])$
13: $\quad reward\_history \leftarrow reward\_history.append(r)$
14: $\quad action\_history \leftarrow action\_history.append(action\_choice)$

A bandit algorithm is a map from the whole history of arms pulled and rewards observed to an arm choice or a probability distribution over arms i.e. given a history, it will return an arm or a distribution over arms. 'bandit_algorithm' mentioned in Algorithm 1 is such an algorithm.

We can add 'Persistence' to any bandit algorithm. In the persistence variant of any bandit algorithm, the only difference with the regular version is that in the persistence version whenever one gets a 1 reward, they stick with their choice for the next time instance. In Algorithm 1, line 10 is what makes it persistent. If line 10 was absent, this would simply be a regular bandit algorithm.

Why should persistence work? The intuition behind persistence is that it will lead to arms with higher means being pulled more often and hence result in lower regret. When an arm with mean μ_i is picked, in expectation, with persistence, it will be picked $\frac{1}{1-\mu_i}$ times before the next decision needs to be made about which arm to pick. So, with persistence we expect the better arms to be picked more often and hence incur lesser regret as compared to the regular variant of most bandit algorithms.

3 Empirical Results

Our experiments are done for the two-armed case because usually analysis of the two-armed bandit problem can be generalised to the n-armed bandit. We wish to examine how 'persistence' influences the performance of various bandit algorithms.

For ϵ-greedy [Wat89], we use a constant value of ϵ. In our experiments, we keep $\epsilon = 0.05$. It picks the arm with the highest empirical mean with probability $1 - \epsilon$, and a random arm with ϵ.

Graphs for ϵ-greedy:

Throughout these graphs, it can be seen that persistence is doing better than the regular version.

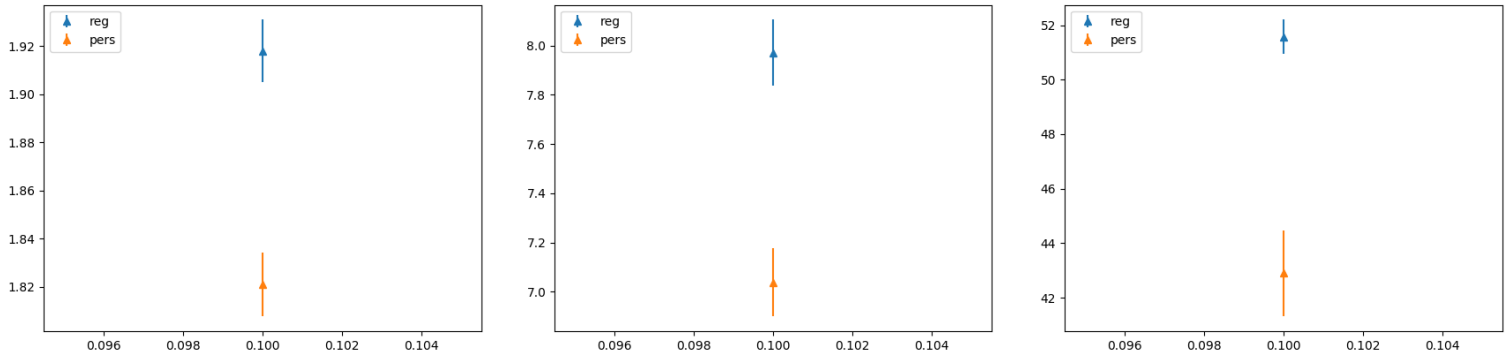

Table 1: Regret for (0.3, 0.1) Horizons = 100, 1000 and 10000; Orange = Persistence; Blue = Regular

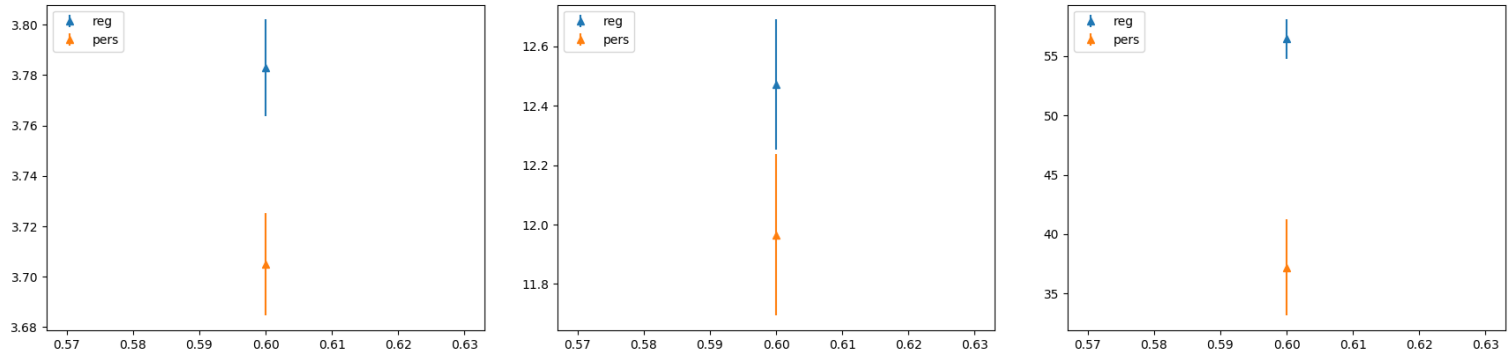

Table 2: Regret for (0.8, 0.6) Horizons = 100, 1000 and 10000; Orange = Persistence; Blue = Regular

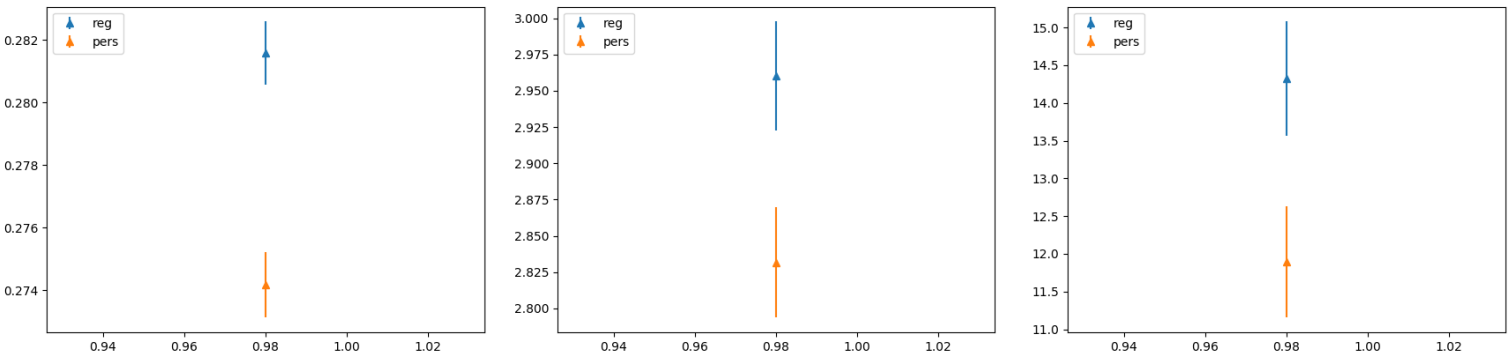

Table 3: Regret for (0.99, 0.98) Horizons = 100, 1000 and 10000; Orange = Persistence; Blue = Regular

Graphs for Thompson Sampling:

Here, we observe that for instances where both means aren't very high, the persistence version outperforms the regular version. On the other hand, for those instances we observe a 'slump' in Tables 6 and 8. The higher the two arms' means are, the longer is the slump. But, for large enough time horizon, the persistence version again starts to outperform the regular version.

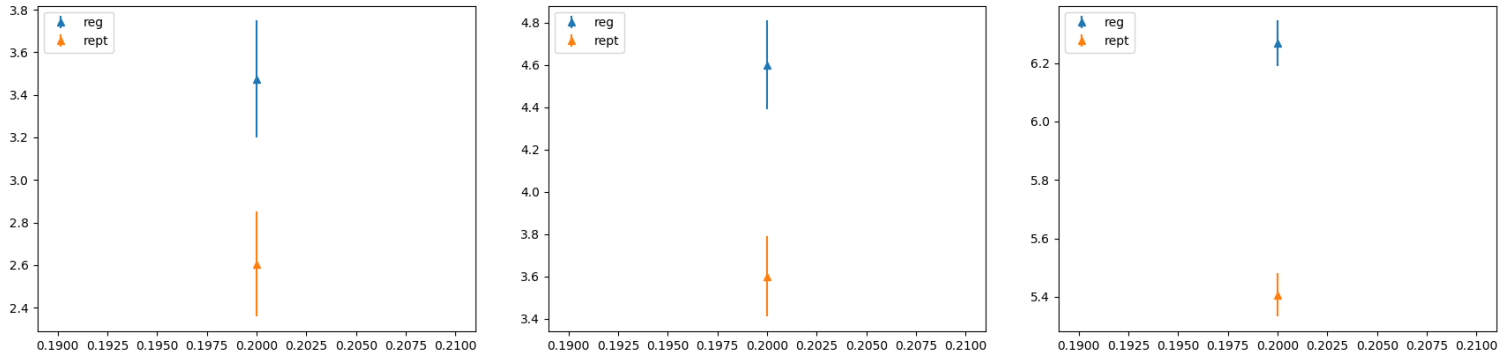

Table 4: Regret for (0.7, 0.2) Horizons = 100, 1000 and 10000; Orange = Persistence; Blue = Regular

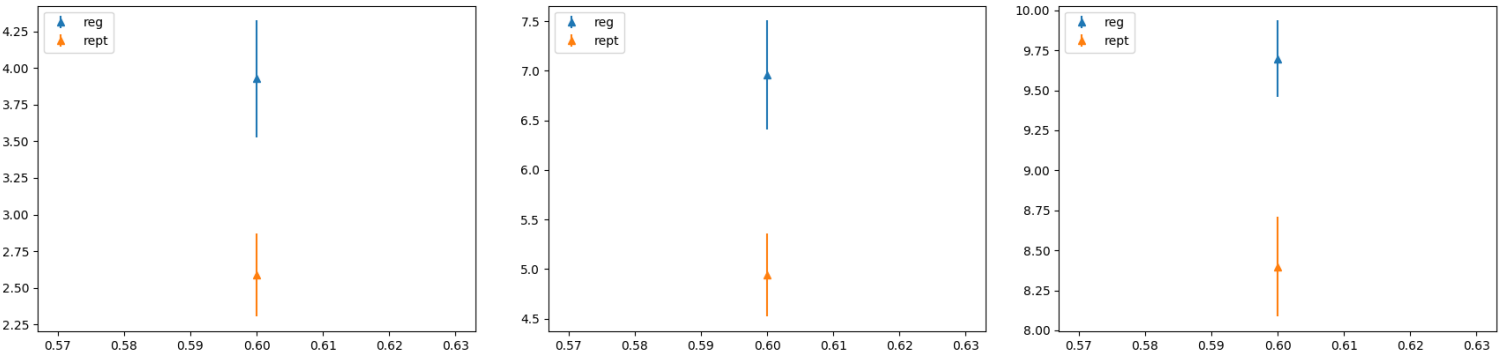

Table 5: Regret for (0.8, 0.6) Horizons = 100, 1000 and 10000; Orange = Persistence; Blue = Regular

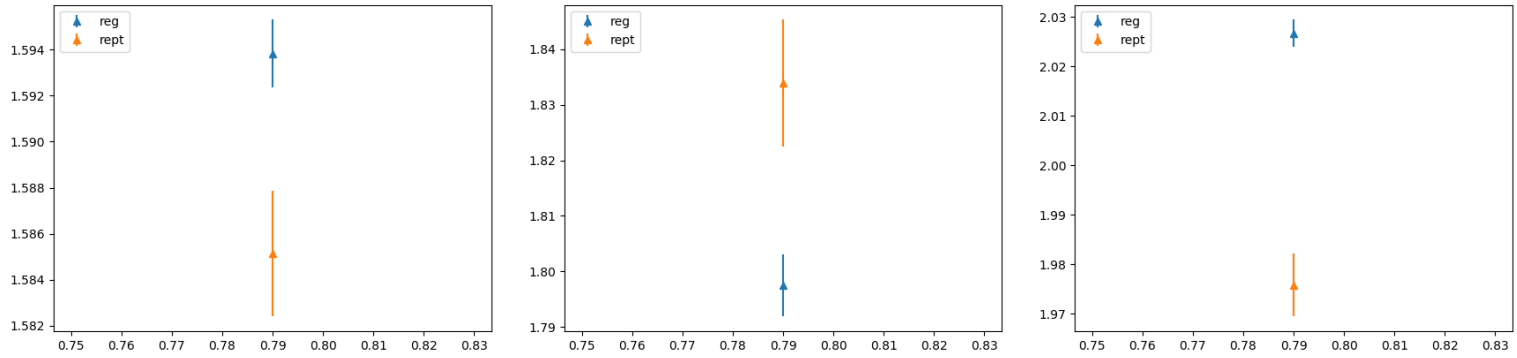

Table 6: Regret for (0.99, 0.79) Horizons = 50, 100 and 500; Orange = Persistence; Blue = Regular

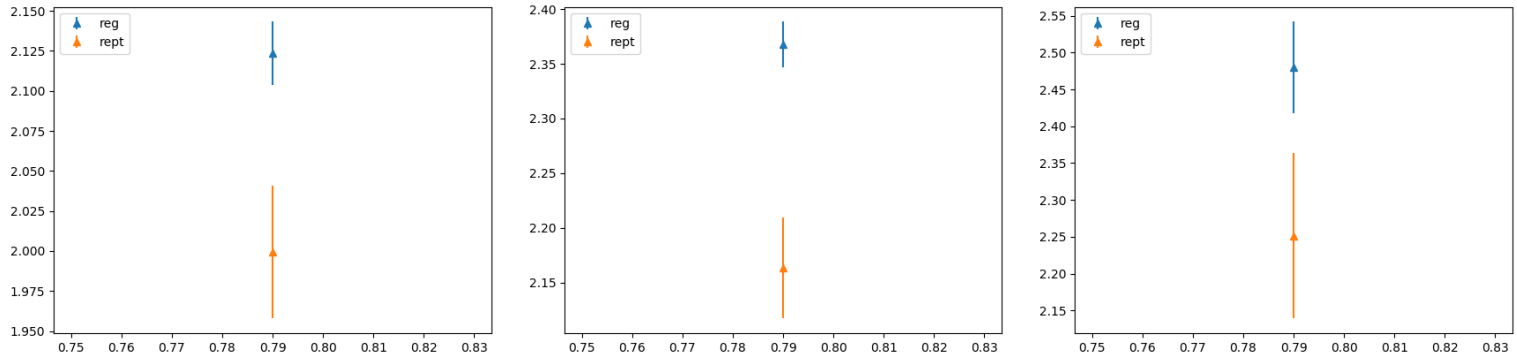

Table 7: Regret for (0.99, 0.79) Horizons = 1000, 5000 and 10000; Orange = Persistence; Blue = Regular

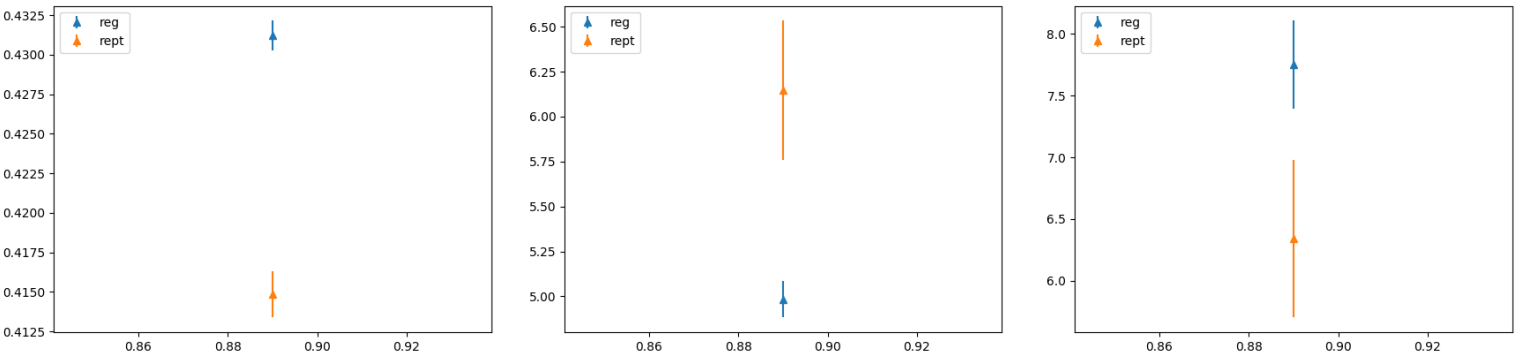

Table 8: Regret for (0.99, 0.89) Horizons = 10, 10^6 and 10^8; Orange = Persistence; Blue = Regular

4 Theoretical Guarantees and Discussion on Empirical Results

For ϵ-greedy, we prove that we are in something called a 'bad' state at most for a constant number of time steps. We define a 'good' state as a state where the arm with the highest expected value given the history indeed is the one with the highest mean. If this is not the case, we are in a 'bad' state. Also, we prove that, whenever we are in a good state, in expectation, the regret incurred is lower for the persistence variant than the regular one. Together, these two statements are enough to say that, in expectation, the persistence variant does better than the regular variant beyond a certain horizon.

We'll do all our analysis for the two-armed case because usually analysis of the two-armed bandit problem can be generalised to the n-armed bandit. Without loss of generality, let us assume that the two arms have means μ_1 and μ_2 with $\mu_1 > \mu_2$. Let $\Delta = \mu_1 - \mu_2$

Fact 1 Hoeffding's Inequality: Let $X_1, ..., X_t$ be i.i.d random variable bounded by the interval [0, 1] and such that $\mu = E[X_i]$ and $M(k) = (X_1 + ... + X_k)/k$

$$P(M(k) - \mu \geq c) \leq e^{-2kc^2}$$

where $c \geq 0$

For the regular version:

At any given time t, in expectation, at least $\epsilon t/2$ pulls of each arm have been made. Using Fact 1 for the rewards of arm 2, with $c = \Delta/2$ and $k = \epsilon t/2$,

$$P(M_2(t) - \mu_2 \geq \Delta/2) \leq e^{-\epsilon t \Delta^2/4}$$
$$=> P(M_1(t) < M_2(t)) \leq e^{-\epsilon t \Delta^2/4}$$

This is the probability of being in a 'bad' state in ϵ-greedy at time t. Let the expected number of times this event happens till time T be $k_2(T)$.

$$k_2(T) \leq \Sigma_{t=1}^{T} e^{-\epsilon t \Delta^2/4}$$

$$k_2(T) \leq \frac{e^{-\epsilon\Delta^2/4} - e^{-\epsilon(T+1)\Delta^2/4}}{1 - e^{\epsilon\Delta^2/4}} \leq \frac{e^{-\epsilon\Delta^2/4}}{1 - e^{\epsilon\Delta^2/4}}$$

Therefore, for at least $T - k_2(T)$ decision times, we are in a good state. In a good state, the expected regret in one time step is:

$$\epsilon\frac{\Delta}{2} \qquad (1)$$

Now, for the persistence version:

The analysis is going to be similar but we are going to only look at the times when compound pulls i.e. our time scale, instead of looking at each pull, will now look at only the time when a new decision needs to be made. Note that, after arm i is chosen, in expectation it will get pulled $\frac{1}{1-\mu_i}$ times before a new choice needs to be made. Here, time t represents the time when the t^{th} choice is made.

Now, at any given time t, in expectation, at least $\epsilon t/2$ compound pulls of each arm have been made. Using Fact 1 for the rewards of arm 2, with $c = \Delta/2$ and $k = \epsilon\frac{t}{2(1-\mu_2)}$,

$$P(M_2(t) - \mu_2 \geq \Delta/2) \leq e^{-\epsilon\frac{t}{1-\mu_2}\Delta^2/4}$$
$$=> P(M_1(t) < M_2(t)) \leq e^{-\epsilon\frac{t}{1-\mu_2}\Delta^2/4}$$

This is the probability of being in a 'bad' state in persistent ϵ-greedy at time t. Let the expected number of times this event happens till time T be $k_2(T)$.

$$k_2(T) \leq \Sigma_{t=1}^{T} e^{-\epsilon\frac{t}{1-\mu_2}\Delta^2/4}$$

$$k_2(T) \leq \frac{e^{-\epsilon\Delta^2/4} - e^{-\epsilon\frac{T+1}{1-\mu_2}\Delta^2/4}}{1 - e^{\epsilon\Delta^2/4}} \leq \frac{e^{-\epsilon\Delta^2/4}}{1 - e^{\epsilon\Delta^2/4}}$$

Therefore, for at least $T - k_2(T)$ decisions, we are in a good state. In a good state, the expected regret per time step (the actual time, not the 'compound' time) is:

$$\frac{\frac{\epsilon\Delta}{2(1-\mu_2)}}{\frac{\epsilon}{2(1-\mu_2)} + \frac{1-\epsilon/2}{1-\mu_1}} \qquad (2)$$

Comparing (2) and (1):

$$(1) - (2) = \epsilon\frac{\Delta}{2} - \frac{\frac{\epsilon\Delta}{2(1-\mu_2)}}{\frac{\epsilon}{2(1-\mu_2)} + \frac{1-\epsilon/2}{1-\mu_1}}$$

$$= \frac{\epsilon\Delta}{2}[1 - \frac{\frac{1}{1-\mu_2}}{\frac{\epsilon}{2(1-\mu_2)} + \frac{1-\epsilon/2}{1-\mu_1}}]$$

$$= \frac{\epsilon\Delta}{2}[\frac{\frac{1-\epsilon/2}{1-\mu_1} - \frac{1-\epsilon/2}{1-\mu_2}}{\frac{\epsilon}{2(1-\mu_2)} + \frac{1-\epsilon/2}{1-\mu_1}}] \geq 0$$

Clearly, (2) is less than (1). Therefore, except at 'bad' states which occur only finitely many times, the average regret incurred is lesser in persistent ϵ-greedy.

Therefore, beyond a certain horizon, persistent ϵ-greedy is going to be better than regular ϵ-greedy.

From our experiments: We see that for ϵ-greedy, for all problem instances and all horizons we run experiments at, the persistence variant outperforms the regular one. We can see this in Table 1 to Table 3.

On the other hand, for Thompson Sampling the picture is a little more complicated. For most problem instances, the persistence variant outperforms the regular version consistently (Tables 4 and 5). There is a small set of problem instances, though, where both μ_1 and μ_2 are high. For such instances, the persistence variant has a 'slump' compared to the regular version (Tables 6 - 8). Between some t_1 and t_2, the regular version performs better. Our observations suggest that t_1 keeps decreasing and t_2 keeps increasing when μ_1 and μ_2 become even higher. For example, the slump for (0.99, 0.89) starts earler and ends later than the one of (0.99, 0.79). We hypothesize that, for any bandit instance (μ_1, μ_2), there exists, a time that is a function of μ_1 and μ_2, beyond which the persistence outperforms the regular version even for the harder instances.

Why does this 'slump' occur? The algorithm has 1/2 probability of picking the μ_2 arm in the 1^{st} time step. If μ_2 is high and the persistence version has been deployed, this arm will end up being pulled a lot in the beginning ($\frac{1}{1-\mu_2}$ times in expectation) updating its beta parameters to indicate a high mean for arm 2with high certainty. This lowers the probability of μ_1 arm being pulled and ensures that the higher mean of the μ_1 arm is discovered at a much later time. This means that with 1/2 probability the persistence algorithm can reach a local minima and remain stuck there for a while. But, things aren't that bad, because this only happens when the second arm has a high mean too, and so the regret incurred, is only slightly more than that of the regular version.

5 Future Work

There are many potential directions for future research. First, theoretical guarantees need to be proved for Thompson Sampling. We wish to prove in the future that, in expectation, Persistent Thompson Sampling has lesser regret than Regular Thompson Sampling for problem instance dependent time horizons.

A family of persistence algorithms can be looked at, where the maximum persistence, i.e. the maximum time one can go without making a new decision, is a

parameter. This parameter can be constant or a variable.

6 Miscellaneous

Other things we worked on:
- **Tighter bounds on Thompson Sampling:** We looked at the two armed problem as a markov chain and tried to get a handle on the regret by looking at analytically computed probabilities of ending up in different states. It was extremely complex because of the complicated expression for probabilities of arm choices from a given state but it did give us the exact expected regret for the simple (1, 0) bandit.
- **The Batch Bandit problem**, where the next b pulls need to be decided simultaneous. We had promising experimental results, but we couldn't get the bounds as tight as we wanted them.
-Experiments on **persistence family of algorithms** with fixed persistence and with persistence increasing by addition or multiplication by a constant when the previous limit is reached.
- **Discrete Support Thompson Sampling** where instead of using Beta distributions as priors for the arms we used distributions over candidate arm values with varying discretization between 0 and 1, e.g. (0, 0.01, 0.02, 0.03...0.99, 1). This discretization outperformed regular Thompson sampling for all our experiments.
- **Binary Bandits** - Similar to discrete support, here the support distribution is over the two arms (μ_1, μ_2). This setting was easier to deal with analytically compared to regular Thompson Sampling because the arm choice probabilities given a state is a function which is easy to compute. We were able to show that, in expectation, probability of choosing optimal arm would increase from any state.
- **Open Loop Algorithm** - Here, when a choice is made for an arm to be pulled, another choice is made about how long to pull the arm for. This algorithm has promising empirical results. We haven't looked into this theoretically yet.

7 References

[LR85] T.L Lai and Herbert Robbins. "Asymptotically Efficient Adaptive Allocation Rules". In: *Adv. Appl. Math.* 6.1 (Mar. 1985), pp. 4–22. ISSN: 0196-8858. DOI: 10.1016/0196-8858(85)90002-8. URL: http://dx.doi.org/10.1016/0196-8858(85)90002-8.

[Wat89] C.J.C.H. Watkins. "Learning from Delayed Rewards". In: *Ph.D. Thesis, University of Cambridge, Cambridge* (1989).

[ACF02] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. "Finite-time Analysis of the Multiarmed Bandit Problem". In: *Machine Learning* 47.2 (May 2002), pp. 235–256. ISSN: 1573-0565. DOI: 10.1023/A:1013689704352. URL: https://doi.org/10.1023/A:1013689704352.

[KKM12] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. "Thompson Sampling: An Asymptotically Optimal Finite Time Analysis". In: (2012). eprint: arXiv:1205.4217.