Regret Minimization for Correlated Bandits

Naman Aggarwal (160010058)

Indian Institute of Technology Bombay

26 June 2020

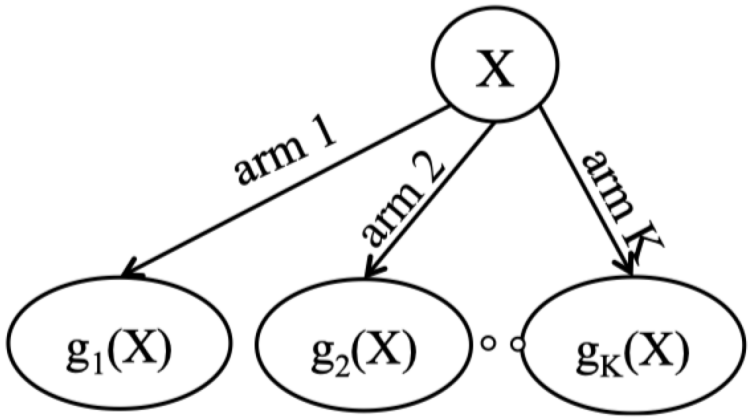
Why Study Correlated Bandits?

Unlike the classic MAB model that considers arms with independent rewards, correlated arms framework captures several applications where the rewards of arms $k = 1, ..., K$ depend on a common source of randomness. For example, the response to K possible advertisements/products can depend on a latent variable X that represents the social/economic condition of a customer.

We can understand the latent random phenomena from the apparent/observable random behaviour.

Problem Formulation

Number of arms = K
Discrete latent random variable, $X \in \{x_1, x_2, \cdots x_n\}$
Latent distribution, $P_X = \{p_1, p_2, \cdots p_n\}$
Bandit Instance = $\{g_1(.), g_2(.), \cdots g_K(.)\}$



Mean of the i'th arm, $\mu_i = E[g_i(X)]$
Optimal arm mean, $\mu_* = \max_i E[g_i(X)]$
At time t, pull arm A_t. Observed reward, $r_t = g_{A_t}(X_t)$
Active set at time t, $S_t = g_{A_t}^{-1}(r_t) = \{x : g_{A_t}(x) = r_t\}$

Problem Formulation

Continued...

- Note that the functions $\{g_i(.)\}_{i=1}^{K}$ need not be invertible. Infact, having invertible functions makes our problem trivial and is reduced to the learning from experts setting.
- **Main question:** Can we exploit or make use of the structure of this setting, the fact that we have a common latent source to have improved regret performance? How does then our proposed algorithm compare in performance to algorithms for the general setting like UCB?

General Strategy

In the general setting, pulling one arm yields no useful information about the reward of any other arm at that same timestep. In this correlated setting however, we can exploit the fact that rewards are generated from the same latent source of randomness to gain non-trivial information about $r_j(t)$, reward generated by any arm j ($\neq i$) at time t, just by observing $r_i(t)$ assuming $A_t = i$.

Idea: Generate the active set, S_t from A_t and r_t. Now, $\tilde{r}_j(t) = f_{j,t}(S_t)$ becomes your estimate of $r_j(t)$. This is a general framework on how to exploit common structure/information between arms. The exact design of this estimator is left to the algorithm designer to be achieved based on his own objectives.

There is existing work in this space ([Gupta et al., 2020]). We move to our proposed strategy.

Proposed Strategy

The most obvious approach to exploit correlation is to learn the latent distribution, P_X. Let $\tilde{P}_X(t) = \{\tilde{p}_1(t), \tilde{p}_2(t) \cdots, \tilde{p}_n(t)\}$ be our estimator of the latent distribution at time t.

- Define estimation error $\varepsilon(t)$ as, $\varepsilon(t) = E[\sum_{j=1}^{n}(\tilde{p}_j(t) - p_j)^2]$.
- If $\varepsilon(t) \to 0$ as $t \to \infty$ under some policy, say for example $\varepsilon(t) = O(\frac{1}{t})$, then $\tilde{\mu}_i(t) = E_{\tilde{P}_X(t)}[g_i(X)] \approx \mu_i$ for large enough t.

This motivates the following algorithm:

Algorithm 1: Use an epsilon-greedy policy with an appropriate ϵ_t schedule.

$$\begin{aligned} A_t &= \arg\max_i \tilde{\mu}_i(t-1), \text{ w.p. } 1-\epsilon_t \\ &= \text{some\_exploratory\_policy}(t), \text{ w.p. } \epsilon_t \end{aligned}$$

such that $\lim_{t\to\infty} \epsilon_t = 0$.

Remark: An exploratory policy is essential for learning the latent distribution (more on this later).

Proposed Strategy

Continued...

Note that the above algorithm works only if the resulting estimation error $\varepsilon(t)$ goes to 0 in the limit which is yet to be verified.

Algorithm 2: This is more similar in spirit to CUCB in the sense that you generate pseudo-rewards for all unpulled arms at a timestep and maintain corresponding pseudo-empirical means. Say $A_t = i$ and the observed reward is r_t, using which generate the active set, S_t. Generate pseudo-reward for arm k at time t as,

$$\tilde{r}_{k,i}(r_t) := \frac{\sum_{j:x_j \in S_t} \tilde{p}_j(t) g_k(x_j)}{\sum_{j:x_j \in S_t} \tilde{p}_j(t)} = E_{\tilde{P}_X(t)}[g_k(X)|X \in S_t]$$

Define the pseudo-empirical mean as,

$$\tilde{\mu}_k(t) = \frac{\sum_{\tau=1}^{t} \tilde{r}_{k,A_\tau}(r_\tau)}{t}$$

Proposed Strategy

Continued...

Algorithm 2 is epsilon greedy with respect to the above defined notion of pseudo-empirical mean.
Remark: Note that $E[\tilde{r}_{k,i}(r_t)] = E_{\tilde{P}_X(t)}[g_k(X)]$ using the towering property of conditional expectation. Therefore,

$$\begin{aligned} E[\tilde{\mu}_k(t)] &= \frac{\sum_{\tau=1}^{t} E_{\tilde{P}_X(\tau)}[g_k(X)]}{t} \\ &= \frac{\sum_{\tau=1}^{t} E_{P_X}[g_k(X)]}{t} + \frac{\sum_{\tau=1}^{t} E_{\tilde{P}_X(\tau)}[g_k(X)] - E_{P_X}[g_k(X)]}{t} \\ &= \mu_k + b_k(t) \end{aligned}$$

$b_k(t)$ is the bias term. Therefore, $\tilde{\mu}_k(t)$ is a biased estimator of μ_k. We show below that the bias term dies down to 0 in expectation as $t \to \infty$ if and only if $\varepsilon(t)$ also goes to 0 in the limit.

Proposed Strategy

Continued...

$$
\begin{aligned}
b_k(t) &= \frac{\sum_{\tau=1}^{t} E_{\tilde{P}_X(\tau)}[g_k(X)] - E_{P_X}[g_k(X)]}{t} \\
&= \frac{\sum_{\tau=1}^{t} \left(\sum_{j \in [n]} (\tilde{p}_j(\tau) - p_j) g_k(x_j) \right)}{t}
\end{aligned}
$$

Taking expectation followed by norm (using norm-squared of expectation is less than expectation of norm-squared),

$$
\begin{aligned}
\|E[b_k(t)]\|^2 &\leq \sum_{\tau=1}^{t} \frac{E\left[\left\| \sum_{j \in [n]} (\tilde{p}_j(\tau) - p_j) g_k(x_j) \right\|^2 \right]}{t^2} \\
&\leq \sum_{\tau=1}^{t} \frac{E\left[\sum_{j \in [n]} \|(\tilde{p}_j(\tau) - p_j) g_k(x_j)\|^2 \right]}{t^2}
\end{aligned}
$$

Proposed Strategy

Continued...

As mentioned earlier, $g_i(.) \in [0, 1]\ \forall i$,

$$\begin{aligned} \|E[b_k(t)]\|^2 &\leq \sum_{\tau=1}^{t} \frac{E\left[\sum_{j\in[n]} \|(\tilde{p}_j(\tau) - p_j)\|^2\right]}{t^2} \\ &= \frac{\sum_{\tau=1}^{t} \varepsilon(\tau)}{t^2} \end{aligned}$$

Since $\{\varepsilon(t)\}_t$ is a convergent sequence, numerator in the expression above is bounded as $t \to \infty$ whereas the denominator is not. Therefore,

$$\lim_{t\to\infty} \|E[b_k(t)]\|^2 = 0,\ \forall\ k$$

Hence, proved.

Proposed Strategy

Continued...

Algorithm 3: This proposed algorithm is akin to the ETC (explore then commit) algorithm for general bandit settings. An exploratory policy is run for some time (hyper-parameter) to learn the latent distribution following which the arm with greatest estimated expected reward is pulled until the end of time horizon.
Performance illustrated in later slides.

We first discuss existing work on latent distribution learning followed by how we adopt it in our algorithms.

Background: Active Distribution Learning from Indirect Samples [Gupta et al., 2018]

Given a latent random variable X and arms $\{g_1, g_2, \cdots, g_K\}$, the estimated latent distribution $\tilde{P}_X(t)$ is said to be asymptotically consistent if $\lim_{t\to\infty} \varepsilon(t) = 0$ (definition of $\varepsilon(t)$ stated earlier).

For each $k = 1, \cdots, K$, let $\{z_{k,1}, z_{k,2}, \cdots z_{k,m_k}\}$ denote the set of possible outcomes (or range) of function g_k. m_k is the total number of distinct outputs of g_k.

Sample Generation Matrix: The information about g_k required to estimate P_X can be captured in matrix A_k with m_k rows and n columns,

$$\begin{aligned} A_k(i,j) &= 1, \text{ if } g_k(x_j) = z_{k,i} \\ &= 0, \text{ otherwise} \end{aligned}$$

for each $i = 1, \cdots, m_k$ and $j = 1, \cdots, n$.

Background: Active Distribution Learning from Indirect Samples

Continued...

A_k is referred to as the *Sample Generation Matrix* for arm k. Let the matrix A be given by $A = [A_1^T, A_2^T, \cdots A_K^T]^T$; the size of A is $m \times n$ where $m = m_1 + m_2 + \cdots + m_K$.

Result: It is possible to achieve asymptotically consistent estimation if and only if rank$(A) = n$ (see [Gupta et al., 2018] for proof).

Examples: Consider the following two instances of a 3-arm bandit

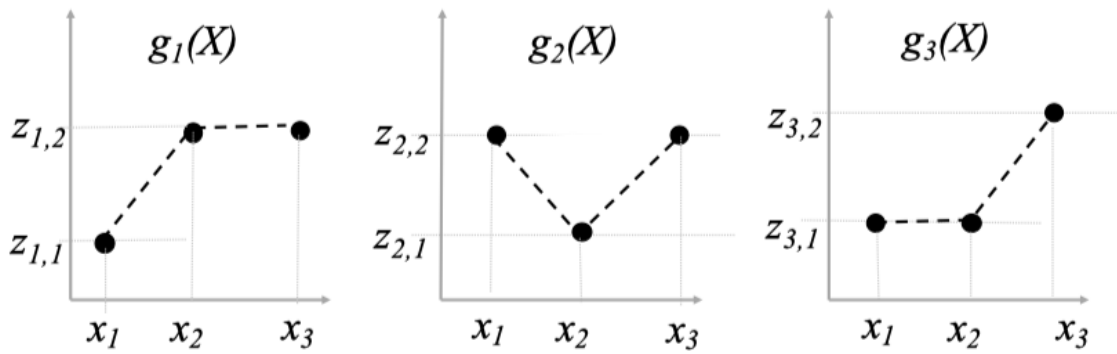


Figure 1: Instance 1 - asymptotically consistent estimation possible

Background: Active Distribution Learning from Indirect Samples

Continued...

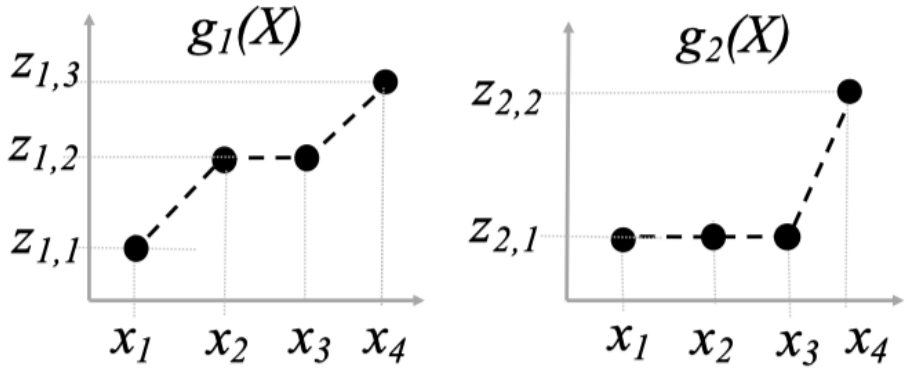


Figure 2: Instance 2 - not possible to get consistent estimation

Constructing the *sample generation matrices* for the two bandit instances,

$$A^{\text{Example-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A^{\text{Example-2}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Background: Active Distribution Learning from Indirect Samples

Continued...

How does this work? Recall that $z_{k,i}$ represents the i'th distinct output of arm k. Let $q_{k,i}$ represent the probability of observing $z_{k,i}$ each time arm k is pulled. We can relate these probabilities of occurence of outputs to the distribution of the latent random variable X by the following system of linear equations:

$$q_{k,i} = \sum_{j=1}^{n} A_k(i,j) p_j$$

for each $k = 1, \cdots, K$ and $i = 1, \cdots, m_k$. This set of equations can be written as:

$$AP_X = Q$$

where $Q = \{q_{1,1}, \cdots, q_{1,m_1}, \cdots\cdots, q_{K,1}, \cdots, q_{K,m_K}\}^T$.

Active Distribution Learning Algorithm

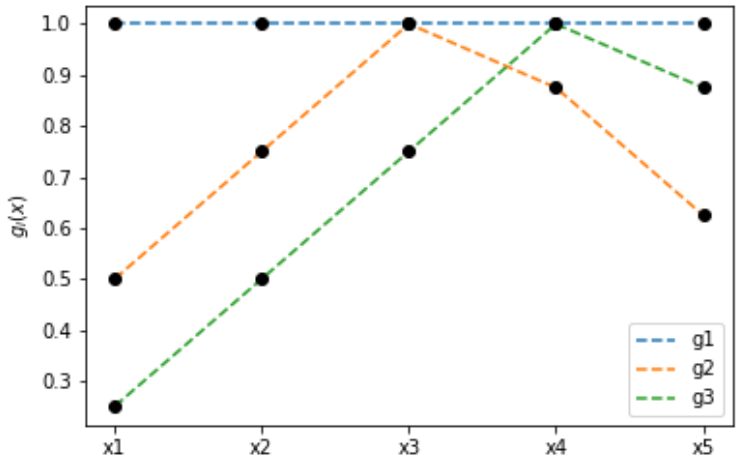
We calculate and update the empirical probability of all outputs at each timestep $\tilde{Q}(t)$ and obtain $\tilde{P}_X(t)$ as:

$$\tilde{P}_X(t) = A^+ \tilde{Q}(t)$$

where A^+ is the pseudo-inverse or the Moore-Penrose inverse of the matrix A. Note that $\tilde{P}_X(t)$ is an unbiased estimator of P_X.

Trade-off between Distribution Learning and Regret Minimization

Consider the following bandit instance.



Notice that arm 1 is the optimal arm with maximum expected reward but reveals nothing useful about the latent distribution. Therefore, distribution learning and regret minimization are two starkly different problems and any algorithm which uses an estimate of the latent distribution to pull arms to facilitate regret minimization needs to have an appropriate exploratory schedule.

Performance

We use the following bandit instance for all our experiments. Note that the optimal arm (arm 1) is non-invertible. Lets analyze the performance of our algorithms against this bandit instance.

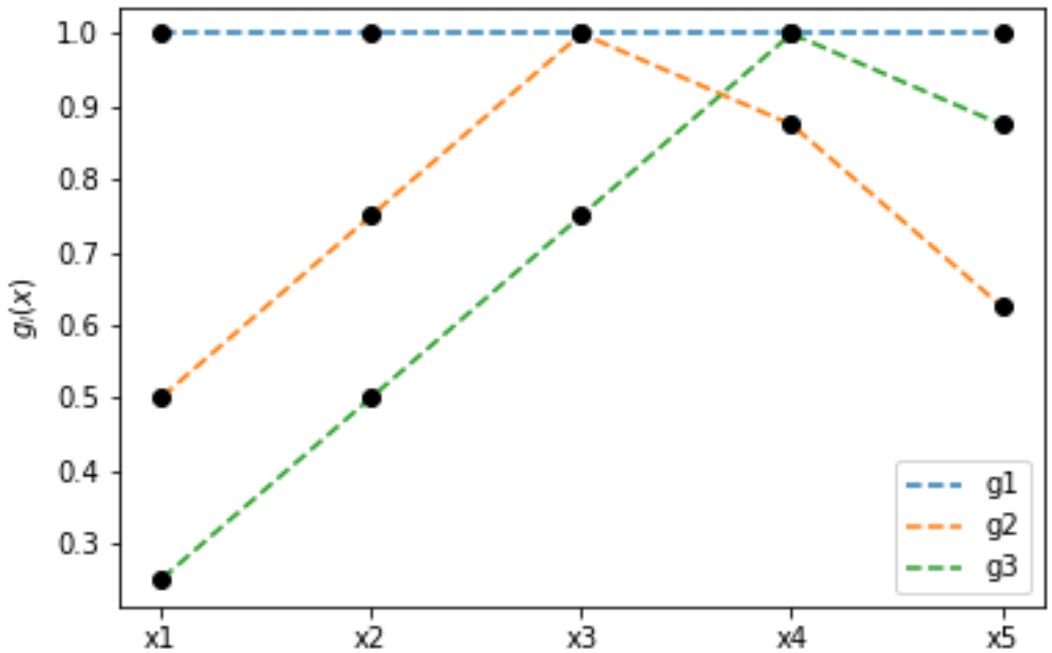


Figure 3: Bandit Instance

Performance

Continued...

Experiment 1: ETC-CB

We run an exploratory policy (round-robin for our experiments) for time $t_{\text{explore}} = 1/\epsilon$.

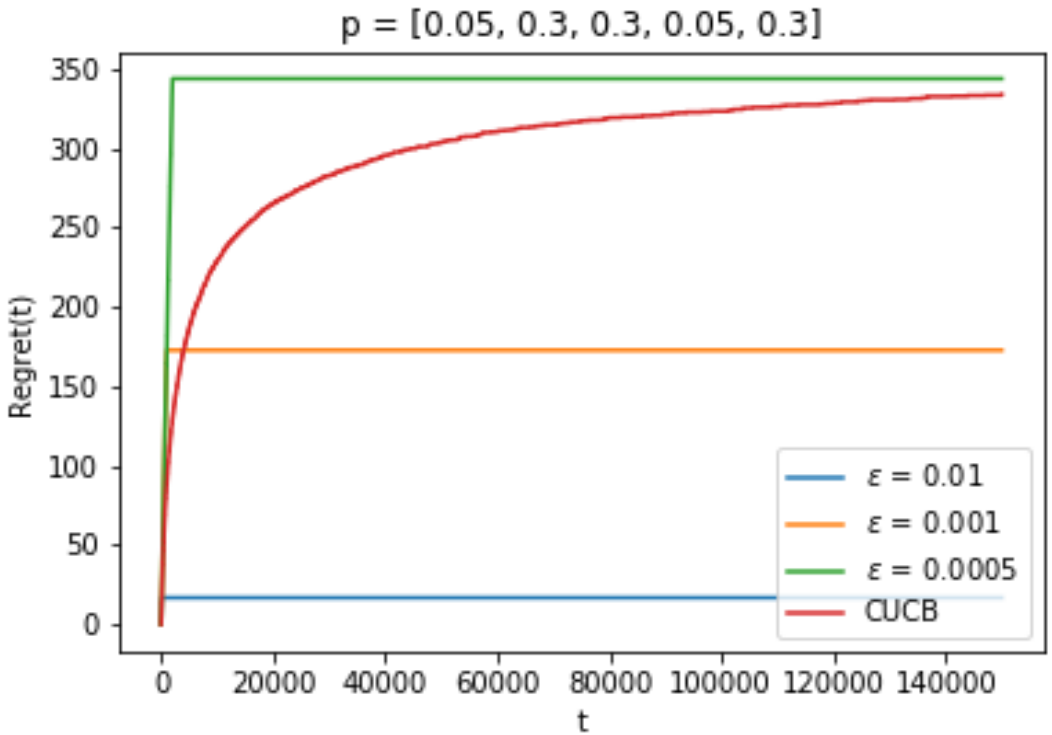


Figure 4: Comparing CUCB to our ETC-CB

Performance

Continued...

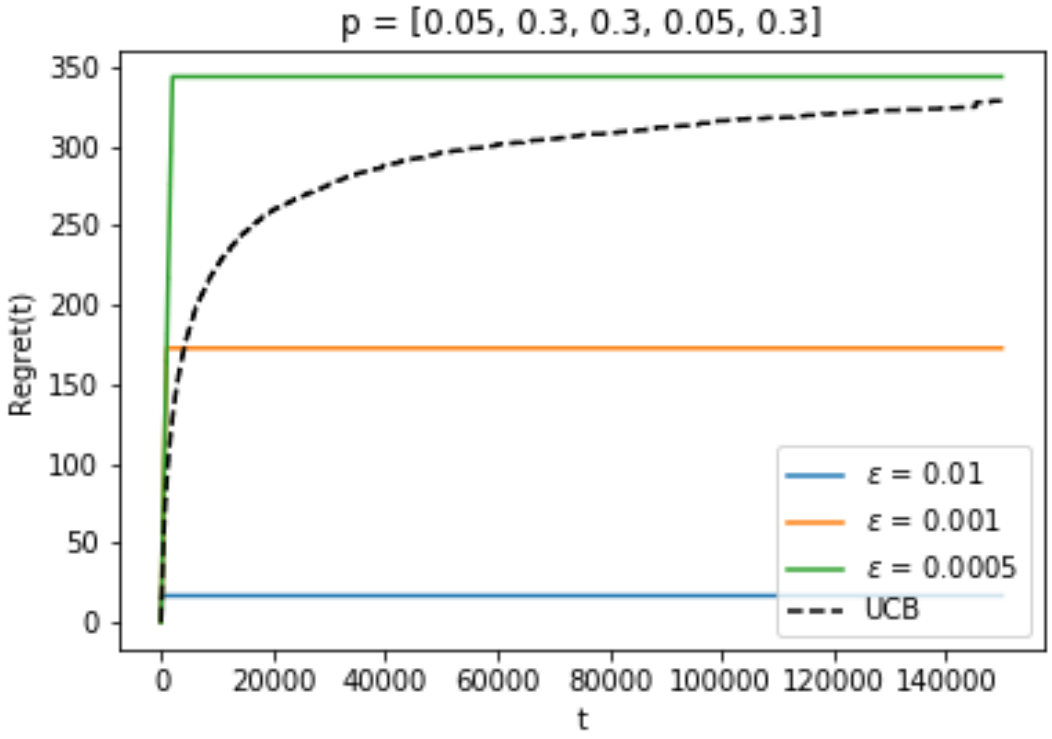


Figure 5: Comparing UCB to our ETC-CB

With the right exploratory schedule (parametrized by ϵ), we see a significant improvement of regret performance over both CUCB and UCB.

Performance

Continued...

Experiment 2: ϵ_t-greedy

We use Algorithm-2 as defined before and the associated definition of generating pseudo-rewards for our simulations in Experiment-2.

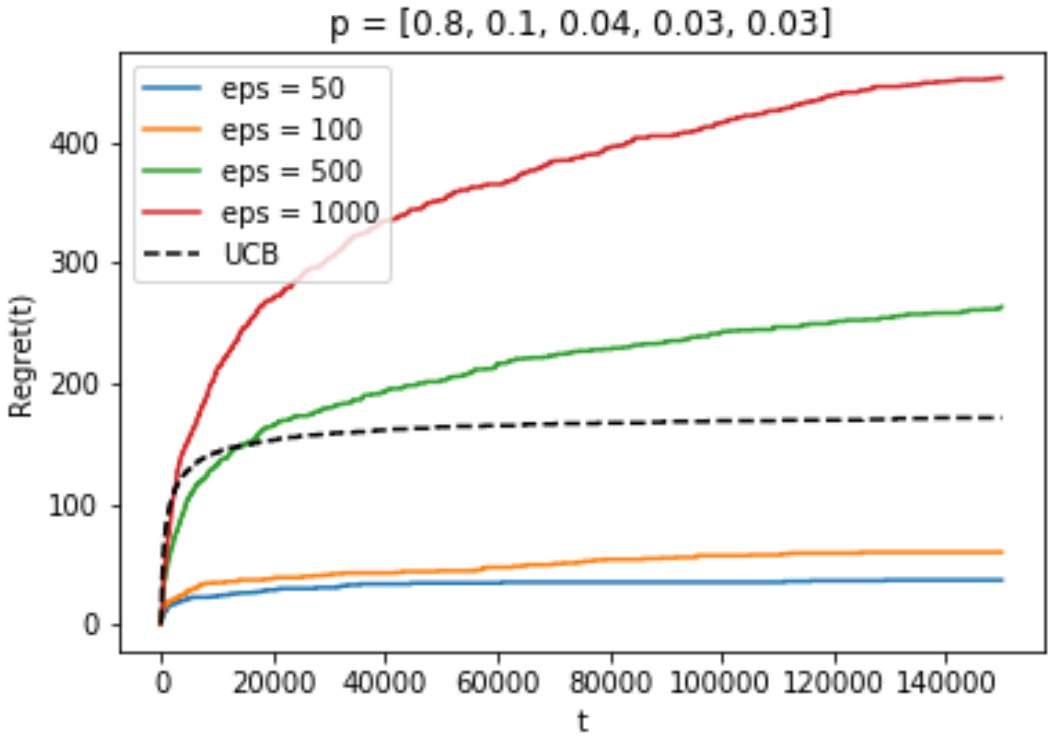


Figure 6: p $= p_1$, $\epsilon_t = \frac{0.25}{1+t/\text{eps}}$

Performance

Continued...

Squared error, $e(t) = \sum_{\tau=1}^{t} \sum_{j \in [n]} (\tilde{p}_j(\tau) - p_j)^2$

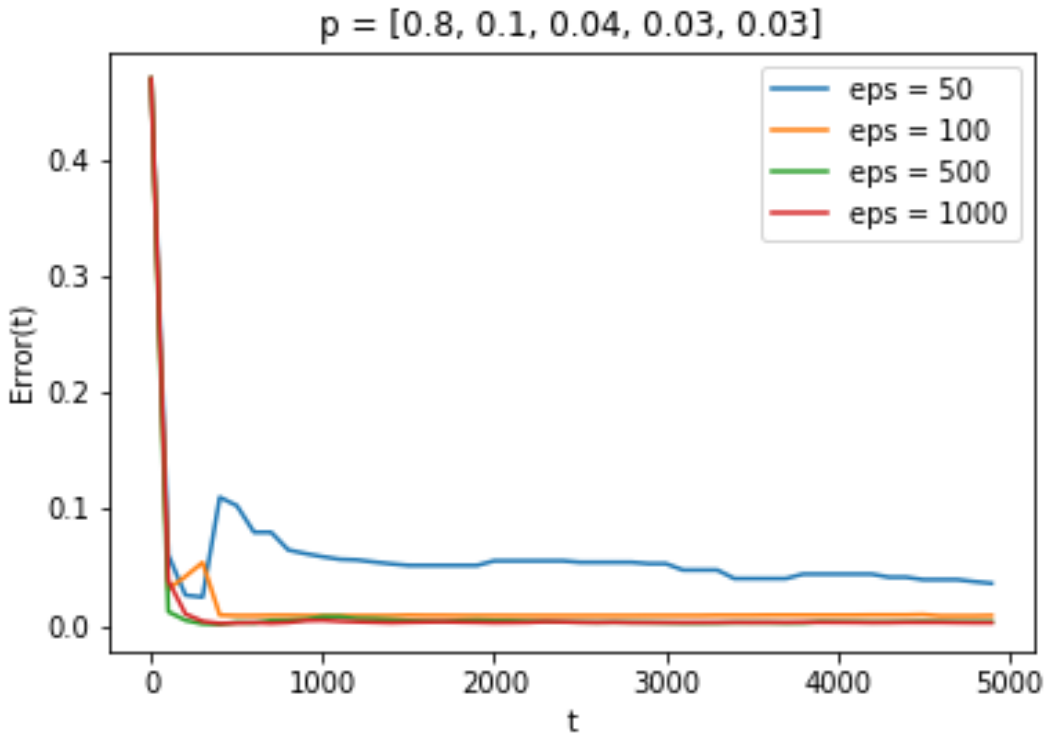


Figure 7: $\mathrm{p} = p_1$, $\epsilon_t = \frac{0.25}{1+t/\mathrm{eps}}$

The above two plots are for the latent distribution, $p = p_1 = [0.8, 0.1, 0.04, 0.03, 0.03]$.

Performance

Continued...

Now, let us perform simulations for $p = p_2 = [0.03, 0.07, 0.41, 0.245, 0.245]$.

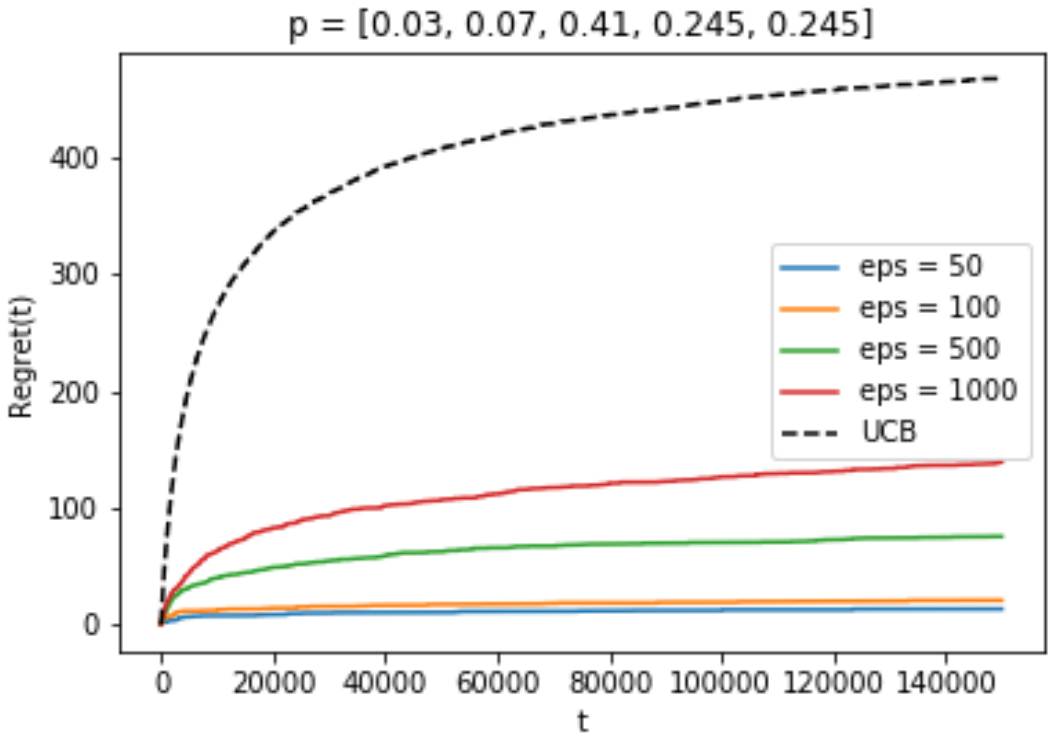


Figure 8: p $= p_2$, $\epsilon_t = \frac{0.25}{1+t/\text{eps}}$

Performance

Continued...

Squared error, $e(t) = \sum_{\tau=1}^{t} \sum_{j \in [n]} (\tilde{p}_j(\tau) - p_j)^2$

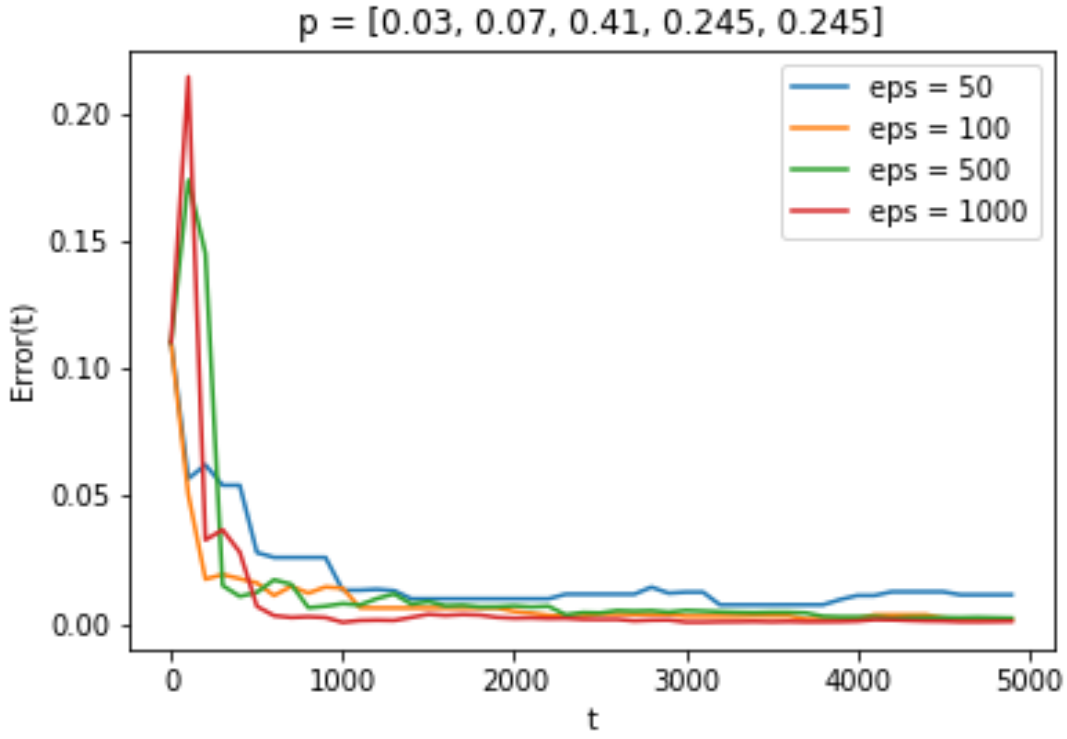


Figure 9: p $= p_2$, $\epsilon_t = \frac{0.25}{1+t/\text{eps}}$

Conclusions and Future Work

Conclusions

- Our algorithms work as a proof-of-concept for achieving regret minimization through latent distribution learning.
- This requires a carefully chosen exploratory schedule.
- These results encourage us to probe further and gain theoretical backing for our proposed algorithms.

Future Work

- Finite time and asymptotic regret analysis
- Theoretically capturing the trade-off between our exploratory schedule and regret
- Characterizing "optimal" exploratory schedule for our regret minimization algorithm

References

Samarth Gupta, Gauri Joshi, and Osman Yağan. Correlated multi-armed bandits with a latent random source. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3572–3576. IEEE, 2020.

Samarth Gupta, Gauri Joshi, and Osman Yağan. Active distribution learning from indirect samples. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1012–1019. IEEE, 2018.